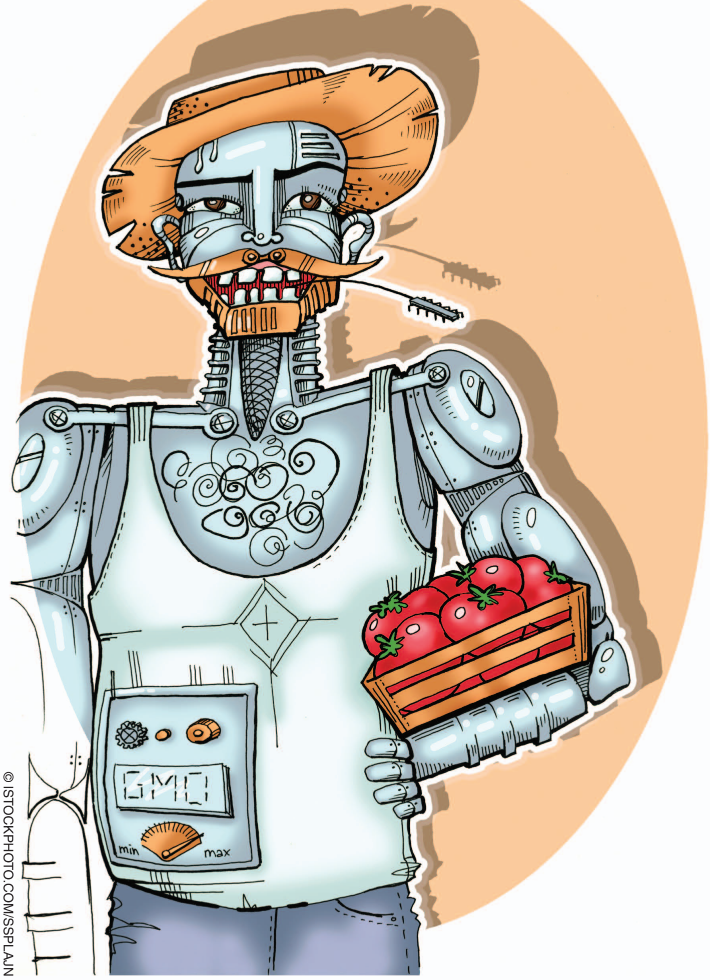

© ISTOCKPHOTO.COM/SSPLAJN

Robot Farmers

Autonomous Orchard Vehicles Help Tree Fruit Production

By Marcel Bergerman, Silvio M. Maeta, Ji Zhang, Gustavo M. Freitas, Bradley Hamner, Sanjiv Singh, and George Kantor

This article presents perception and navigation systems for a family of autonomous orchard vehicles. The systems are customized to enable safe and reliable driving in modern planting environments. The perception system is based on a global positioning system (GPS)-free sensor suite composed of a two-dimensional (2-D) laser scanner, wheel and steering encoders, and algorithms that process the sensor data and output the vehicle's location in the orchard and guidance commands for row following and turning. Localization is based on range data to premapped landmarks, currently one at the beginning and one at the end of each tree row. The navigation system takes as inputs the vehicle's current location and guidance commands, plans trajectories for row following and turning, and drives the motors to achieve fully autonomous block coverage. The navigation system also includes an obstacle detection subsystem that prevents the vehicle from colliding with people, trees, and bins. To date, the vehicles sporting the perception and navigation infrastructure have traversed over 350 km in research and commercial orchards and nurseries in several U.S. states. Time trials showed that the autonomous orchard vehicles enable efficiency gains of up to 58% for fruit production tasks conducted on the top part of trees when compared with the same task performed on ladders. Anecdotal evidence collected from growers and workers indicates that replacing ladders with autonomous vehicles will make orchard work safer and more comfortable.

Digital Object Identifier 10.1109/MRA.2014.2369292
Date of publication: 13 March 2015

Agricultural Robotics

Agriculture is the backbone of society, providing the food, feed, fiber, and fuel on which all humans depend to live. The industry's greatest challenge is to meet the demands of a growing population without increasing—or better yet, while reducing—its environmental footprint. Experts indicate that we must double our agricultural production if we are to meet the needs of humankind in 2050 [4]. Obviously, this cannot be done by simply doubling the inputs (water, land, labor, seeds, chemicals, etc.) because many of them are overstretched or their environmental impact is already too large.

1070-9932/15©2015IEEE

What is needed is a sustained increase in production efficiency, on the order of 25% [9]. Part of it will come from better farming practices, new seeds, improved disease prediction and control, and pest management. Robotics and automation will also be part of this equation. In the grain industry, GPS- and vision-guided farm vehicles have been proven to increase efficiency and reduce operator fatigue. The tree fruit industry is starting to experience a similar revolution, part of which is being developed at our lab in Pittsburgh, Pennsylvania, and orchards in Pennsylvania and Washington.

In this article, we describe a family of autonomous vehicles developed for the tree fruit industry under the United States Department of Agriculture (USDA)-funded Comprehensive Automation for Specialty Crops (CASC) project [3], [10], [6]. The vehicles can be operated unmanned, for example, when covering an entire orchard block to mow, spray, scout for disease and insects, or perform crop yield estimation. They can also transport workers pruning limbs, thinning fruit, performing tree maintenance, installing insect pheromone dispensers, or harvesting fruit. The contribution of this article is the integration of modules into a working system aimed at increasing efficiency and reducing labor costs in fruit production.

The vehicles' basic behavior consists of following the driving row (or simply row) between the trees, detecting the end of a row, exiting the row, turning toward the next row, and entering it. By repeating this behavior, the vehicle is capable of covering an entire block, including visiting each row multiple times. This seemingly simple behavior is accompanied by several challenges that had to be addressed to enable the autonomy necessary for orchard operations: 1) robust row following in the presence of tall grass, tree limbs in the laser's field of view, and missing trees; 2) GPS-free localization with respect to the block with a maximum error of 0.5 m laterally and 1% of the row length longitudinally; 3) detection of major obstacles such as people and bins; and 4) reliable speed control from 0.05 to 2 m/s. These were all addressed with the perception and navigation architecture described in this article. More formally, the vehicles' autonomous behavior spans three control modes, defined together with extension educators and industry stakeholders (Table 1).

Figure 1 shows two of the autonomous orchard vehicles. Figure 1(a), Tuscarora, is the base platform for pace and mule modes. Figure 1(b), Allegheny, is equipped with a scissors lift for scaffold-mode operations.

Base Autonomous Platform

The orchard vehicles presented in this article are based on a Toro eWorkman MDE electric utility vehicle adapted with the sensing, computing, and actuation needed for autonomous operation. The modifications made include

1) steering and wheel encoders with angular resolution of 0.38°/tick and linear resolution of 2.33×10^{-5} m/tick
2) a Sick LMS111 planar laser scanner with a 270° field of view, 0.5° resolution, and 30-m range, mounted horizontally on the center of the front bumper about 1 m from the ground
3) a SmallPC SC240ML fanless, industrial embedded computer with an Intel Core 2 Duo 1.6-GHz central processing unit and 2 GB random-access memory (RAM)
4) steering and brake servomotors from Dynetic Systems actuated by Roboteq dc motor controllers; the steering servo actuates the steering column via a back-drivable chain and a 4:1 gear reduction box, and the brake servomotor via a 49:1 gear reduction box
5) a Sensible Machines, Inc. (SMI) custom interface board that receives encoder and user inputs and translates autonomy commands to the servomotors and the original Toro eWorkman speed controller board

Table 1. The control modes enabled by the autonomous orchard vehicles, from least to most complex.

	Mule Mode	Scaffold Mode	Pace Mode
How it works	The vehicle follows farm workers as they walk along the row, tending and harvesting and placing fruit in bins on the vehicle	Farm workers stand on the vehicle while it self-steers in the row	The vehicle autonomously drives an entire block at a time without requiring any further interaction
Production tasks enabled	Tree tending, harvesting	Pruning, fruit and blossom thinning, tree maintenance, and harvesting	Mowing, spraying, and scouting for disease, insects, and crop yield estimation
Autonomous functionalities	Row following (continuous or stop-and-go), end-of-row detection, obstacle detection	Row following (continuous or stop-and-go), end-of-row detection, obstacle detection	Row following, end-of-row detection, turn and enter new row, obstacle detection
Vehicle speed	0.5–1 m/s	0.05–0.1 m/s	1–2 m/s
Human–machine interface	Pocket-size control box with buttons	Control box and foot pedals installed on vehicle	Handheld tablet or smartphone
Permanent infrastructure installed in the orchard	None	None	Reflective landmarks installed on row ends

(a)

(b)

Figure 1. The autonomous orchard vehicles: (a) Tuscarora and (b) Allegheny. (Photo courtesy of Carnegie Mellon University.)

6) an industrial Ethernet switch ICP-DAS model NS-205-IP67 to connect the onboard computer with the laser, SMI custom interface board, and an external notebook used for development purposes
7) a ruggedized wireless access point EnGenius ENH to provide wireless connection to the pace mode user interface (smartphones and tablets)
8) a ZigBee (serial wireless) radio board to provide wireless connection to the mule- and scaffold-mode user interfaces
9) a manual/autonomous switch on the dashboard
10) a joystick for direct vehicle control
11) emergency stop (e-stop) buttons in strategic locations accessible to workers on the ground or onboard the vehicle.

Figure 2 shows the hardware architecture and the connection between the physical components.

The manual/autonomous switch and e-stop buttons determine the state of the vehicle: manual, autonomous, or e-stopped. In the manual state, the vehicle can be driven in the same way as a regular electric car. If any of the e-stop buttons is pressed, the vehicle goes into e-stopped state and can only recover after all of the e-stop buttons have been released. When in autonomous state, the vehicle can be in one of five substates.

- `SystemReady`: the vehicle is ready to execute an autonomous task.
- `SystemTeleop`: the vehicle is being teleoperated through the joystick.
- `SystemAutonomous`: the vehicle is executing an autonomous task. When the task is finished, the system returns to the `SystemReady` state.
- `SystemAutonomousError`: if an error is detected—e.g., laser scanner data are missing for more than 1 s—the system generates a software e-stop and enters this state.
- `SystemWaitingClear`: in some error situations, the system recovers from the error condition, e.g., the laser scanner starts sending data after an interruption, but the operator is required to manually clear the vehicle so it returns to autonomous operation.

The `SystemAutonomous` substate is further divided into the following:

1) `CascInactive`: the system state before it starts executing an autonomous task or after it finishes it.
2) `CascActive`: the system state while performing a task in one of the three autonomy modes (mule, scaffold, or pace).
3) `CascPaused`: the system state when the operator requests a pause or when the system recovers from an error condition (`SystemWaitingClear`). If the latter case, when the operator clears the vehicle, it resumes operation.

The entire system software runs on the onboard computer on Ubuntu Linux, with the message passing provided by Willow Garage's ROS.

Perception System

The perception system comprises the sensing and vehicle localization subsystems. The sensing suite is composed of the wheel and steering encoders and laser scanner. Sensing is used in every autonomy control mode, while localization is only necessary in pace mode. The perception system outputs the tree rows' supporting lines and the vehicle's position with respect to the reference origin at the corner of the block, both of which are used by the navigation system to drive the vehicle. Their computation is described in the following.

The laser scanner points in the direction of motion and perceives objects on a plane parallel to the ground. It is mounted approximately 1 m above the ground, allowing it to perceive tree trunks and the lower part of tree canopies while avoiding tall grass and weeds. A filter checks the consistency of the laser returns to deal with scanning noise and moving objects. The laser points perceived in the previous measurement frame are projected into the current one using odometry and matched over multiple frames using a grid map—i.e., two points from different frames are matched if they fall on the same grid cell. If a matched laser point appears consistently in a certain number of consecutive frames, it is considered to be a stationary point; otherwise, it is considered noise and discarded (Figure 3).

The next step is to fit the laser points into two parallel straight lines representing the trees. The line fitting employs a random sample consensus-based approach. Consider Figure 4, where the blue lines represent the tree rows on the left and right sides of the vehicle. In each iteration, the approach randomly selects three points—two on the same side of the vehicle and one on the other side—and fits two parallel lines through them. It then uses the lines to evaluate the linearity of the laser points, selects a subset of them as inliers, and uses these inliers to recompute the parallel lines based on a least-square fitting. The line-fitting process is repeated until the mean of the squares of the point-to-line distance is smaller than a threshold or the number of iterations reaches the maximum defined a priori.

The localization subsystem solves the pose estimation problem in two dimensions, returning the heading and planar position of the vehicle. It is implemented in an extended Kalman filter (EKF) framework with one prediction step and two update steps (Figure 5). The prediction step uses the odometry from the wheel and steering encoders as its input; the update steps use the laser points processed in two different ways: 1) row detection and 2) landmark detection.

The landmark update step currently depends on reflective tape installed on the posts at the ends of each tree row. The tape is

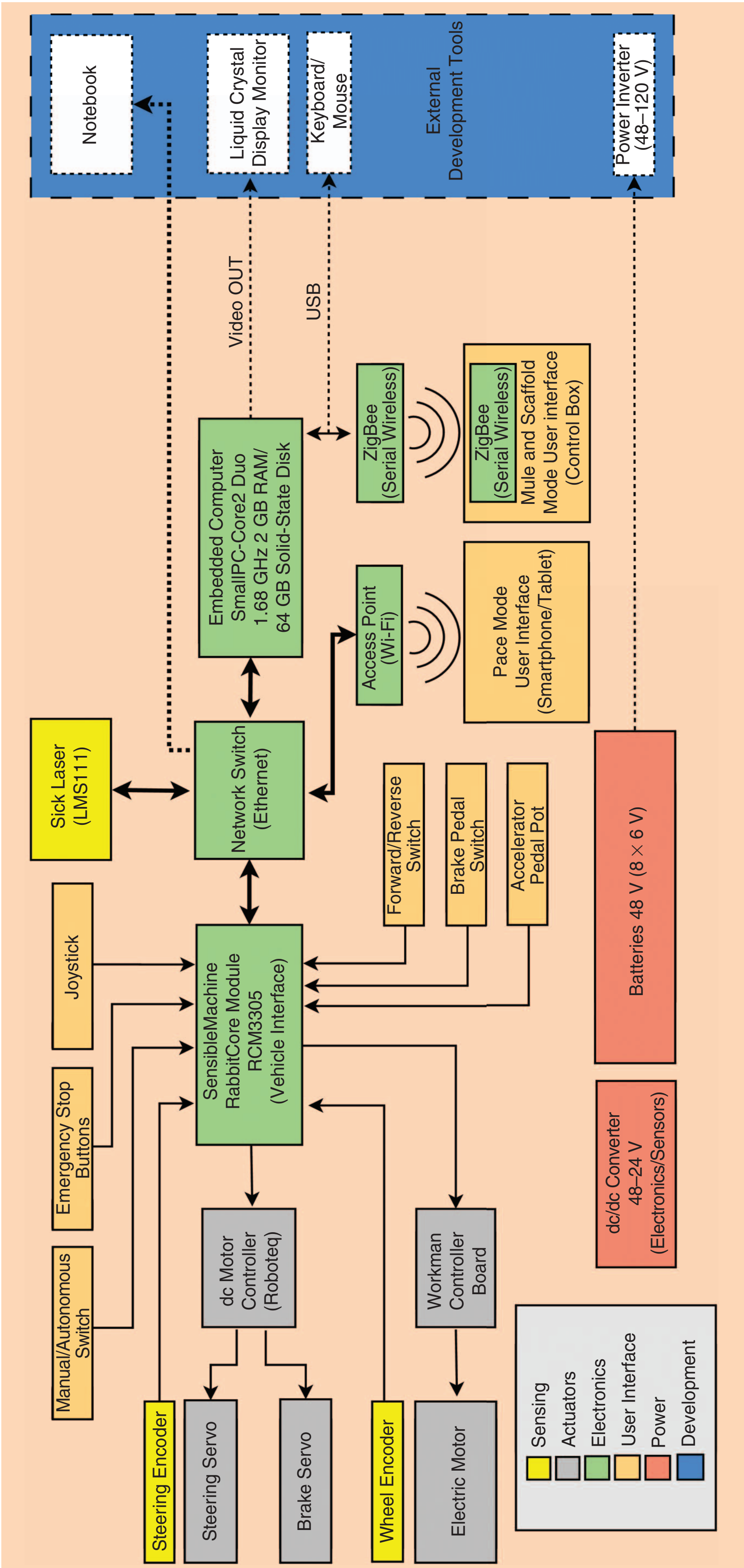

Figure 2. The autonomous orchard vehicle hardware architecture. The Sensible Machines board receives data from sensors and user-provided inputs and sends them to the onboard computer. The computer calculates the guidance commands and sends them—via the interface board—to the driving, brake, and steering motors.

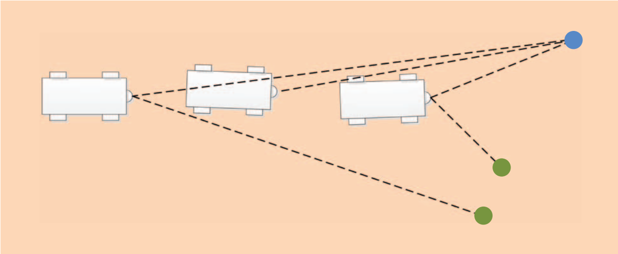

Figure 3. Checking the consistency of a laser scan. Laser points are projected into the current measurement frame and matched over multiple frames. If a point appears in a certain number of frames consistently (e.g., the blue point), it is considered a stationary point; otherwise it is considered noise (e.g., the green points) and discarded.

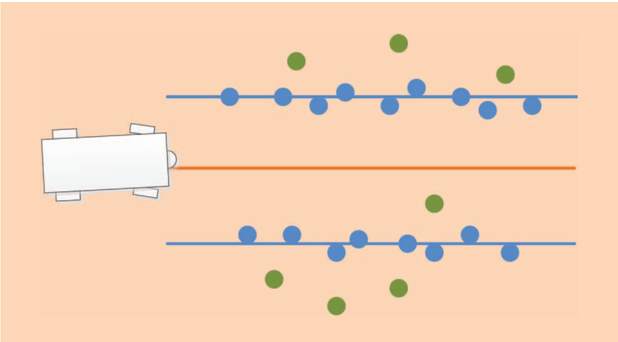

Figure 4. Laser-based row detection. Laser points are fitted into two parallel straight lines (the blue lines) representing the tree rows. The line fitting selects a subset of the laser points as inliers (the blue points) and treats the others as outliers (the green points). The orange line is the center line of the tree rows.

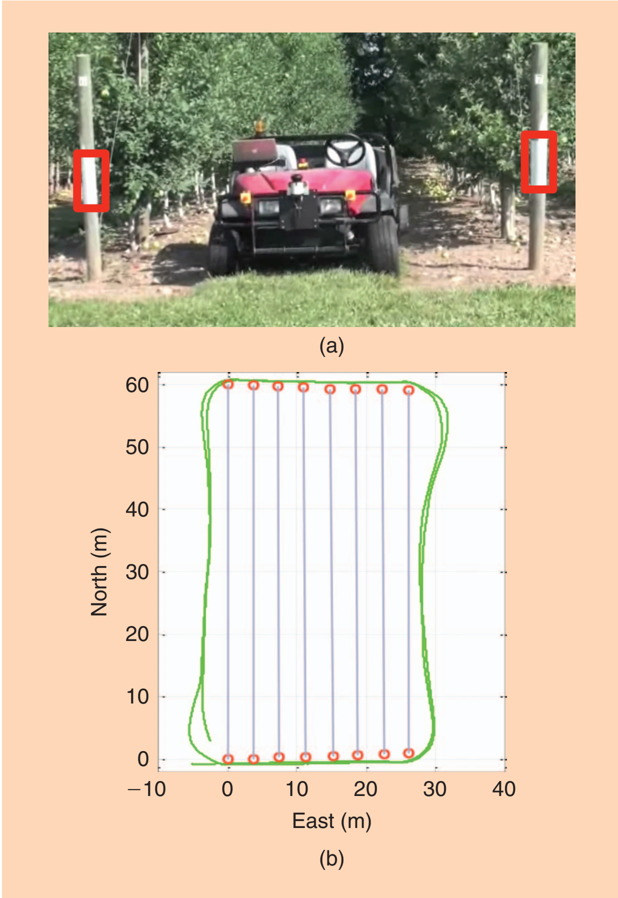

Figure 6. (a) The reflective tape installed on posts at both ends of each row act as landmarks for the localization subsystem. (b) The vehicle is driven around the block to create the local orchard map used for localization [12]. The red circles represent the mapped landmarks, the gray lines connecting the circles represent the tree rows, and the green line around the block represents the trajectory followed by the vehicle during mapping. (Photo courtesy of Carnegie Mellon University.)

visible as the vehicle approaches the end of the row; it is clearly distinguishable from all other objects in the scene because of its high laser returns, which can be easily filtered with a simple threshold (Figure 6).

The first type of EKF update is the row update. It uses as input the tree rows detected as explained earlier. Recall that the tree rows are fitted into two parallel straight lines. In this update step, we use the fact that the parallel lines connect the landmarks on the map because the landmarks are attached to the ends of the tree rows. Using this geometric relationship, the tree rows can be located on the map; then, using the relative position of the vehicle with respect to the tree rows, the pose of the vehicle can be corrected. Note that this update step corrects only the vehicle heading and position perpendicular to the tree rows. The position in the direction of motion cannot be obtained from this geometric relationship alone.

The second EKF update is called a *landmark update* and uses the actual positions of the mapped landmarks. As shown in Figure 7, when two or more landmarks are perceived, planar geometry can be used to determine the vehicle's pose. To avoid mismatches, we use a checking mechanism where the relative positions of the perceived landmarks are compared with those on the map built a priori. If the relative positions do not fit those on the map, the perceived landmarks are considered to be noise and are discarded.

A typical output of the localization subsystem after combining the prediction and the update steps is shown in

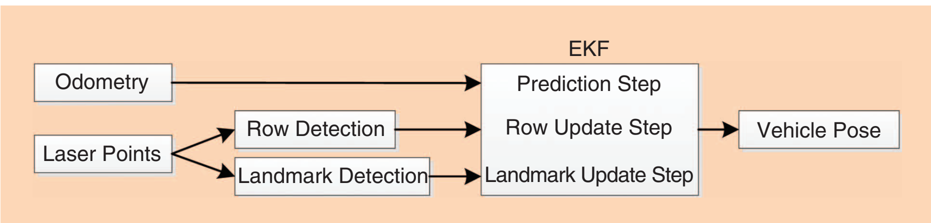

Figure 5. A block diagram of the localization subsystem. Its inputs are the vehicle odometry and laser points, and its output is the 2-D vehicle pose (position and heading).

Figure 8(a). The blue line represents the localization output, and the red line represents the ground truth measured by a high-accuracy inertial navigation system. A histogram of the corresponding downtrack and crosstrack errors is shown in Figure 8(b). Note that 98% of the time, the errors are smaller than the desired 0.5 m, which means the localization subsystem is accurate enough for autonomous navigation in orchards.

Navigation System

The navigation system is responsible for guiding the vehicle during row following and turning. It receives as inputs the vehicle location with respect to the mapped landmarks and the two line equations for the rows of trees and outputs the commands that are sent to all motors by the SMI board. Although it is not navigation from a systemic perspective, for the sake of completeness, we include in this section a description of the human–machine interfaces for the mule, scaffold, and pace autonomy control modes.

When the vehicle is in autonomous mode, it is constantly executing one of two preprogrammed behaviors: row center driver and turn-around driver. In scaffold and mule mode, only the former is executed; in pace mode, it alternates between the two until it covers the entire block. Each behavior checks different criteria to determine when it has completed its execution, upon which the next queued behavior starts.

The row center driver behavior is responsible for ensuring that the vehicle follows the center line between the tree rows (the orange line in Figure 4) plus or minus a constant lateral offset if the vehicle is to drive closer to one side than the other. Currently we use a pure pursuit controller that sets the speed to a constant and the steering angle ϕ to $\phi = k_a a + k_d(d - o_d)$, where a and d are the angle and distance of the row center line with respect to the vehicle, k_a and k_d are proportional gains, and o_d is the lateral offset. To reduce high-frequency noise, a and d are passed through a low-pass filter before being used by the controller.

When the odometry indicates that the vehicle is approaching the end of the row, the navigation system starts to search for a gap in the tree rows on both sides of the vehicle. If it detects a gap with a length larger than a predefined value, it interprets that as a valid end-of-row condition and terminates the row center driver behavior execution.

The turn-around behavior is responsible for guiding the vehicle from one row to the next. The vehicle's physical constraints (minimum turn radius, width, and length) and the orchard's dimensions dictate what kinds of turns can be executed. Rarely will an orchard have wide enough rows and ample enough headland to allow the vehicle to simply exit one row, make a U-turn, and enter the next one. Modern orchard rows are usually 3.5–5-m wide, with a headland no longer than 10 m. We have experimented successfully with two types of turns that obey the various physical constraints involved: between adjacent rows and skipping rows.

In the first case, we use an asymmetric thumb turn, so named because of its distinctive shape. As the vehicle leaves the row, it first turns away from the row it will enter and then

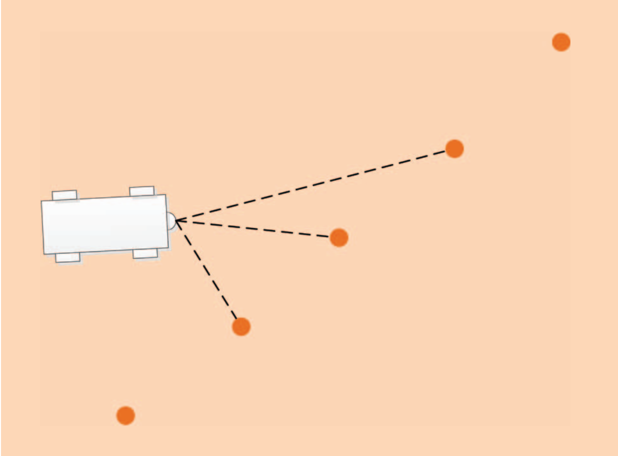

Figure 7. An illustration of the EKF landmark update step. The orange points represent the landmarks on the map. Each dashed line indicates a landmark perceived by the laser scanner. The vehicle's pose can be deduced from its relative position to the landmarks using planar geometry.

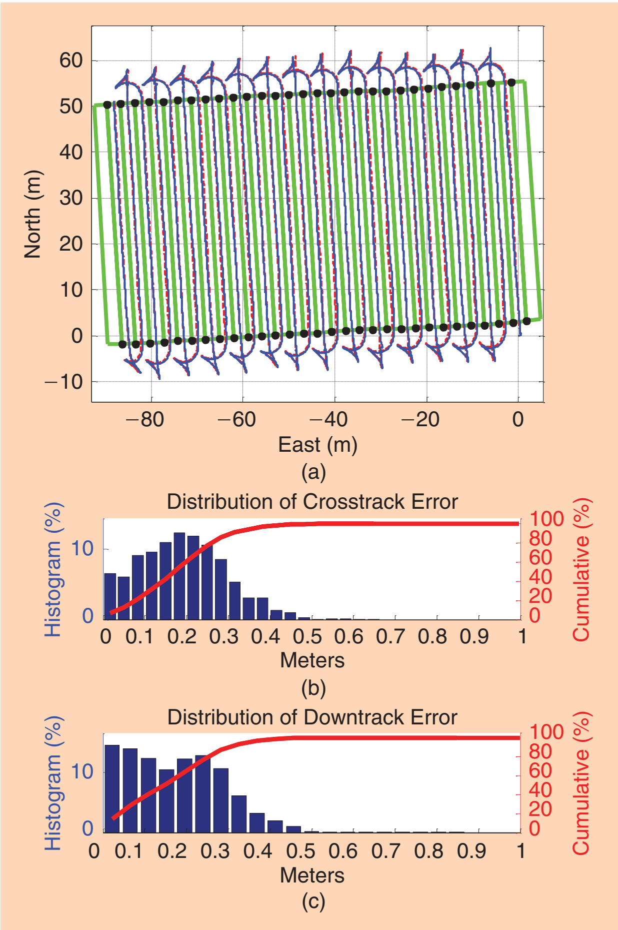

Figure 8. (a) The localization output for a typical apple production block. The black dots represent the mapped landmarks. The blue line represents the localization output, and the red line represents the ground truth provided by a high accuracy inertial navigation system. Their proximity indicates how small the localization error is. The corresponding (b) crosstrack and (c) downtrack error histograms; the red lines represent cumulative errors. The localization system estimates the vehicle position to within 0.5 m 98% of the time and is accurate enough for autonomous navigation in orchards.

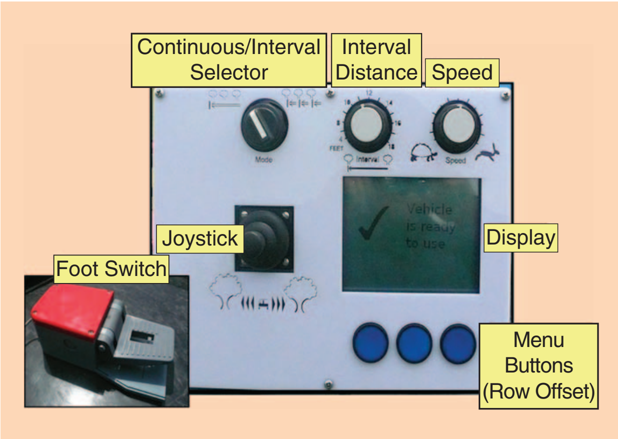

Figure 9. The scaffold mode user interface. The foot pedals on the floor of the platform allow hands-free vehicle operation. The joystick is used to nudge the vehicle left or right during row following, and the three blue buttons allow for offset setting—the center button for zero offset and the left and right buttons for maximum left and right offset. (Photo courtesy of Carnegie Mellon University.)

again turns toward it. The vehicle then aligns itself with the target row such that the landmarks enter the field of view of the laser scanner, thus increasing the probability of correctly localizing itself. In the second case, a manually hardcoded plan (not described in this article) decides a priori in which order the rows will be traversed. In this case, as the vehicle exits the row, it turns toward the next row and then again aligns itself with it before entering. Rows are skipped according to the vehicle's size and the orchard characteristics.

The four autonomous orchard vehicles developed for the CASC project have driven more than 350 km in research and commercial orchards in Pennsylvania and Washington.

In both cases, the vehicle uses the landmarks to correct its position while executing the turn. This prevents errors associated with dead reckoning, which are more pronounced during turns than during straight driving, and errors associated with wheel slippage, a common occurrence in uneven or muddy terrain. As the vehicle completes the turn, it starts looking for the tree rows again. When a valid row center is detected and the vehicle is inside the row, the turn-around driver behavior is terminated and the row center driver behavior restarts. When all rows have been traversed, the vehicle returns to the starting position and waits for a new task.

As the vehicle drives, the perception system updates the vehicle's position and the tree rows' supporting lines. These are used by the navigation system to calculate path tracking and heading errors and to generate the corresponding correction commands to the SMI board. The navigation system also checks for error conditions and, if necessary, interrupts vehicle operation.

The user's interaction with the vehicle is done via custom-designed interfaces. Several iterations were produced until the workers felt comfortable using the interfaces within a few minutes of being shown how to operate the various controls.

For mule mode, the interface is simply a pocket-sized plastic box with one button and a ZigBee radio board. When the user presses the button, the vehicle starts performing row following; when he/she presses the button again, the vehicle stops. The only behavior invoked is row center driver, which means the vehicle knows when it reaches the end of the row and stops automatically.

For scaffold mode, the interface must allow the workers riding on the vehicle to control its speed, its offset from the center, and whether the vehicle will run continuously (e.g., when thinning fruit) or automatically stop after driving a predetermined interval (e.g., when tying trees to guide wires). The design of this interface was informed by a series of user studies over an 18-month period. Our design team followed a typical user interface design process involving 1) the study of the existing work flow and environment, 2) the creation of an initial design and low-fidelity simulation, and 3) testing by users iterating toward increasingly high-fidelity simulations and a working prototype. Volunteers included owners, managers, and workers from commercial apple orchards in Pennsylvania.

The initial interface designs were based on laptops and tablets. This kind of design would be easy to implement since the hardware could be purchased off the shelf. The hardware, however, proved ill suited for the interface. The users did not feel comfortable using the computers, the touchscreens were unfamiliar to workers, and it was too hard for users to quickly find what they needed on the computer screen. Also, the computer and tablet interfaces led to a futuristic feel that users felt at odds with, as if the vehicle were too expensive and fragile for an ordinary worker to use. The users felt uncertain and did not want to break the machine. Finally, the hardware fared poorly in the outdoor environment. The regular computer screens were not bright enough (although e-ink-type screens could alleviate that), and the off-the-shelf tablets fared poorly in hot and dusty conditions.

The final design uses dials, pedals, and levers similar to those found on existing orchard vehicles and tractors (Figure 9). Such components are robust to the orchard environment, the workers are familiar with them, and they allow for easy use even by workers wearing thick gloves. This interface is in operation in the field to this day.

The pace mode interface runs on a tablet or other handheld device, connected to the onboard computer via the wireless access point. The current prototype has two screens (Figure 10). In the Settings screen, the user selects the block where the vehicle will execute the operation, whether the vehicle is to skip rows or not, how many times each row is to

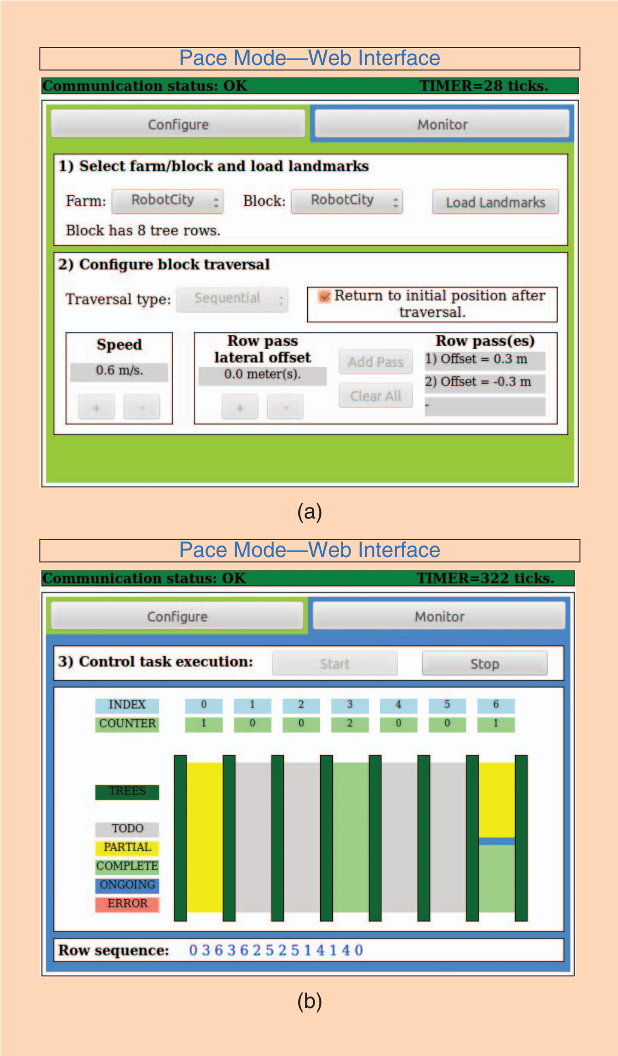

Figure 10. The pace mode user interface: (a) the configuration screen and (b) the monitoring screen.

Figure 11. The autonomous orchard vehicle functions as a "bin dog" for workers harvesting apples. At the push of a button, the vehicle follows the workers as they pick the fruit and place it into the bins. When the bins are full, the vehicle takes them to the end of the row to be loaded into a truck. (Photo courtesy of Carnegie Mellon University.)

Figure 12. The workers at the Allan Bros. Orchard in Prosser, Washington, thin green fruit from atop the orchard vehicle. (Photo courtesy of Carnegie Mellon University.)

be traversed, the offset for each passage, and whether the vehicle should return to its initial position after covering the entire block. In the Monitoring screen, the user can start and stop the operation and monitor progress in real time via a graphical representation of the vehicle and the status of each row: to be traversed, partially traversed, or completed.

Deployment Examples

The four autonomous orchard vehicles developed for the CASC project have driven, in total, more than 350 km in research and commercial orchards in Pennsylvania and Washington (the largest apple producing state in the country). In the first year of deployment, the vehicles were operated solely by Carnegie Mellon University (CMU) engineers. As the hardware and software stabilized, we started conducting experiments where the vehicles were used by actual farm managers and workers in production environments hundreds and even thousands of miles away from the CMU development site. The representative experiments for each control mode are described in the "Mule Mode," "Scaffold Mode," and "Pace Mode" sections.

Mule Mode

Mule mode was demonstrated when the orchard vehicle was used as a "bin dog," carrying apple bins for workers harvesting fruit. The vehicle performed as expected, following the row and stopping whenever the interface button was pressed. In the words of the crew manager, "I can see the practicality [of the bin dog] right away, and the financial savings of not having an operator on the tractor all the time moving the bins is going to be very helpful." Figure 11 shows an experiment in which workers harvest fruit into the bins carried

Two vehicles were equipped with a scissors lift to enable workers to ride on it.

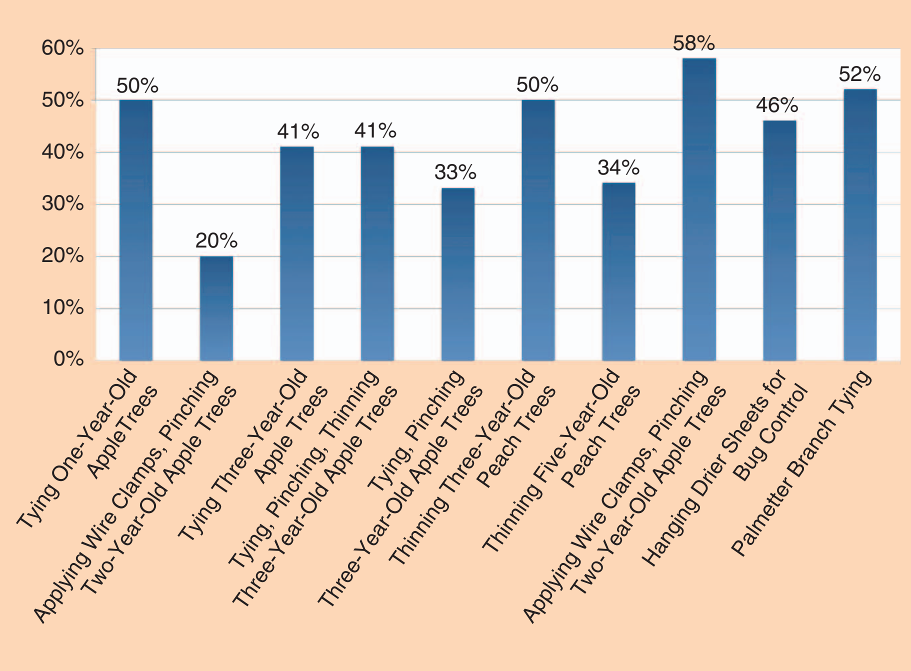

Figure 13. The efficiency gains obtained by workers using an autonomous orchard vehicle versus workers on ladders conducting tree fruit production tasks on the top part of the trees.

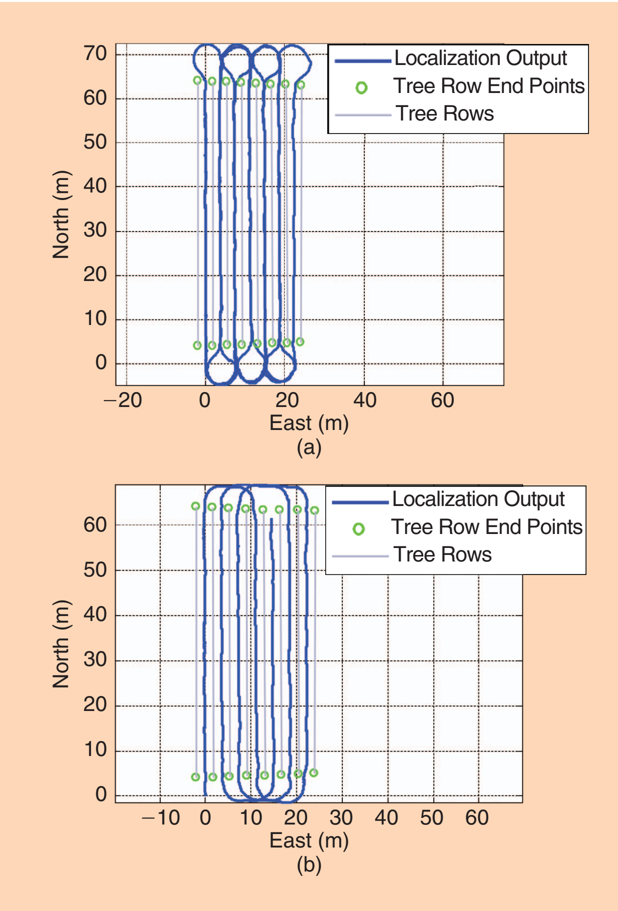

Figure 14. Examples of full-block coverage in pace mode at the Fruit Research and Extension Center in Biglerville, Pennsylvania. (a) The adjacent row turning pattern using thumb turn and (b) the skipping row turning pattern.

by the vehicle; when the bins are full, the vehicle takes them to the end of the row, where they are loaded into a truck.

Scaffold Mode

Scaffold mode is the one that growers saw as the most promising from the points of view of efficiency increase and labor cost reduction. Two vehicles were equipped with a scissors lift to enable workers to ride on it and perform production tasks on the top part of trees (Figure 12). The goal was to investigate the efficiency gains over the use of ladders, which are not only inefficient tools but are responsible for 30% of the labor claims in the Washington apple industry [8].

Time trials conducted by Pennsylvania State University in the summer of 2011 indicate that the autonomous orchard vehicles afford an efficiency increase of up to 58% for a selected number of tree fruit production tasks (Figure 13). Anecdotal evidence also suggests qualitative gains in safety and comfort.

Other experiments conducted by extension educators and growers are as follows [2]:

- Placing pheromone dispensers on top of apple trees from atop the orchard vehicle versus on foot using a tall pole. The workers on the vehicle placed twice as many dispensers as those on foot.
- Green apple thinning at Hollabaugh Bros. Orchards. Here, two men with the vehicle competed with two men with two ladders. On the vehicle crew, one man worked on the vehicle and then dismounted to help the man on the ground finish his row. The men with ladders worked on opposite sides of one row from ladders and the ground simultaneously. The ladder crew did 11.5 trees/man-hour, while the vehicle crew did 16.7 trees/man-hour, for a 45% gain in efficiency.
- Tree training at Yakima Valley Orchards, Grandview, Washington. Four women, two on the vehicle and two on ladders, worked side by side for 20 h. The women on ladders achieved 41 s/tree, while those on the vehicle got 19 s/tree, for a 116% gain in efficiency.

Pace Mode

Pace mode experiments focused on mowing apple blocks in Biglerville, Pennsylvania, over the course of several weeks. Once the landmark infrastructure was in place and a map of the block was created (Figure 6), every time we visited that block we only had to position the vehicle in a known starting position and select the coverage pattern; autonomous coverage followed from there. Figure 14 shows one example each of the adjacent row turning and the skipping row turning patterns.

What's Next

The work reported here is but one example of how autonomous orchard vehicles can be used to increase efficiency and reduce labor costs in tree fruit production. While significant progress by our team and others has been made so far, many technological challenges remain to be addressed, including the following:

- Replacement of the pure pursuit controller with a model-based controller that takes into account vehicle dynamics, especially the rear wheel sideslip, which is inevitable in the grassy, muddy terrain of tree fruit orchards. Wheel sideslip causes odometry errors that are particularly detrimental during turns and may cause the vehicle to skip a row. We have developed and tested such a controller and are in the process of integrating it into the navigation system [1].
- Detection of obstacles, in particular people and bins, and rejection of spurious obstacles such as tall grass. Preliminary work using a push-broom laser scanner with the three-dimensional point clouds processed offline is promising [5] but will require some effort to run in real time on the onboard computer.
- Development of a transit control mode that would allow the vehicle to travel autonomously from a garage to the blocks, between blocks, and back to the garage. Such a mode would enable farmers to allocate a variety of different tasks to the vehicle over the course of the day, possibly mixing the other three modes (mule, scaffold, and pace).
- Investigation of user interfaces based on formal human factors studies. Currently, the only formal guidelines used in the design of the various interfaces were interviews with orchard workers.
- Establishment of standards and regulations for the safe introduction of autonomous vehicles in orchards and other food production environments, the lack of which has hampered commercial success of many an agricultural automation system [9], [11]. We intend to conduct this work within the Robotics and Automation Society Technical Committee for Agricultural Robotics and Automation and have published a position paper in the area [7].

Acknowledgments

We would like to thank extension educators Karen Lewis from Washington State University and Tara Baugher from Pennsylvania State University for their contributions to this article. This work was partially supported by the USDA under grant 2008-51180-04876.

References

[1] G. Bayar. (2013). Long distance autonomous trajectory tracking for an orchard vehicle. *Ind. Robot: Int. J.* [Online]. *40(1)*, pp. 27–40. Available: http://www.emeraldinsight.com/doi/abs/10.1108/01439911311294237

[2] M. Bergerman, D. Lonn, and C. Seavert. 2012. Autonomous orchard vehicles. presented at Specialty Crops Engineering Solutions Workshop, Pittsburgh, PA. [Online]. Available: http://goo.gl/UKwzRa

[3] (2014, June 7). CASC (Comprehensive Automation for Specialty Crops) Project. [Online]. Available: http://www.cascrop.com

[4] J. A. Foley, N. Ramankutty, K. A. Brauman, E. S. Cassidy, J. S. Gerber, M. Johnston, N. D. Mueller, C. O'Connell, D. K. Ray, P. C. West, C. Balzer, E. M. Bennett, S. R. Carpenter, J. Hill, C. Monfreda, S. Polasky, J. Rockström, J. Sheehan, S. Siebert, D. Tilman, and D. P. M. Zaks. (2014, June 7). Solutions for a cultivated planet. *Nature* [Online]. *478*, pp. 337–342. Available: http://goo.gl/YfBFQY

[5] G. Freitas, B. Hamner, M. Bergerman, and S. Singh, "A practical obstacle detection system for autonomous orchard vehicles," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Algarve, Portugal, Oct. 2012, pp. 3391–3398.

[6] B. Hamner, M. Bergerman, and S. Singh, "Results with autonomous vehicles operating in specialty crops," in *Proc. Int. Conf. Robotics Automation*, St. Paul, MN, May 2012, pp. 1829–1835.

[7] D. Kohanbash, M. Bergerman, K. M. Lewis, and S. J. Moorehead, "A safety architecture for autonomous agricultural vehicles," presented at the ASABE Annu. Int. Meeting, Dallas, TX, July 2012, Paper 121337110.

[8] K. Lewis, P. Brown, and C. Seavert. 2012. Harvest assist systems. presented at Specialty Crops Engineering Solutions Workshop, Pittsburgh, PA. [Online]. Available: http://goo.gl/UKwzRa

[9] J. Reid. (2011). The impact of mechanization on agriculture. *Natl. Academy Eng. Bridge, Issue Agriculture Informat. Technol.* [Online]. *41(3)*, pp. 22–29. Available: http://goo.gl/eENvui

[10] S. Singh, M. Bergerman, J. Cannons, B. Grocholsky, B. Hamner, G. Holguin, L. Hull, V. Jones, G. Kantor, H. Koselka, G. Li, J. Owen, J. Park, W. Shi, and J. Teza. "Comprehensive automation for specialty crops: Year 1 results and lessons learned," *J. Intell. Service Robot., Special Issue Agricultural Robot.*, vol. 3, no. 4, pp. 245–262, Oct. 2010.

[11] K. C. Ting, T. Abdelzaher, A. Alleyne, and L. Rodriguez. 2011. "Information technology and agriculture: Global challenges and opportunities," *Natl. Academy Eng. Bridge, Issue Agriculture Informat. Technol.*, *41(3)*, pp. 6–13. Available: http://goo.gl/eENvui

[12] J. Zhang, S. Maeta, M. Bergerman, and S. Singh, "Mapping orchards for autonomous navigation," in *Proc. American Society of Agricultural and Biological Engineers Annu. Int. Meeting*, Montreal, Canada, July 2014.

Marcel Bergerman, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States. E-mail: marcel@cmu.edu.

Silvio M. Maeta, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States. E-mail: smaeta@andrew.cmu.edu.

Ji Zhang, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States. E-mail: zhangji@cmu.edu.

Gustavo M. Freitas, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil. E-mail: gfreitas@coep.ufrj.br.

Bradley Hamner, Near Earth Autonomy, Pittsburgh, Pennsylvania, United States. E-mail: bradley.hamner@gmail.com.

Sanjiv Singh, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States. E-mail: ssingh@cmu.edu.

George Kantor, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States. E-mail: kantor@cmu.edu.