3-2005

Controller Area Network Based Distributed Control for Autonomous Vehicles

Matthew J. Darr
The Ohio State University

Timothy S. Stombaugh
University of Kentucky, tim.stombaugh@uky.edu

Scott A. Shearer
University of Kentucky

Controller Area Network Based Distributed Control for Autonomous Vehicles

Digital Object Identifier (DOI)
https://doi.org/10.13031/2013.18312

Notes/Citation Information
Published in *Transactions of the ASAE*, v. 48, issue 2, p. 479-490.

CONTROLLER AREA NETWORK BASED DISTRIBUTED CONTROL FOR AUTONOMOUS VEHICLES

M. J. Darr, T. S. Stombaugh, S. A. Shearer

ABSTRACT. *The goal of this project was to evaluate the potential of a controller area network (CAN bus) to be used as the communication network for a distributed control system on an autonomous agricultural vehicle. The prototype system utilized microcontroller-driven nodes to act as control points along a CAN bus. Messages were transferred to the steering, transmission, and hitch control nodes via a task computer. The task computer utilized global positioning system data to generate appropriate control commands. Laboratory and field testing demonstrated that each of the control nodes could function simultaneously over the CAN bus. Results showed that the task computer adequately applied a feedback control model to the system and achieved guidance accuracy levels well within the desired range. Testing also demonstrated the system's ability to complete normal field operations, such as headland turning and implement control.*

Keywords. *Autonomous vehicle, Controller area network, Distributed control systems, Microcontrollers, Precision agriculture.*

Over the past several years, technology has continued to play an increasing role in agriculture. The industry has recently seen the advent and development of many types of automated vehicles ranging from planters to sprayers to harvesters. These vehicles have all sustained different levels of automation. Some were capable of fully autonomous field operations, while others were developed for specific control operations such as autosteering (Reid et al., 2000). Commercialization of these technologies has come at a substantial cost to the farmer. Autosteer systems range in cost from $10,000 to $50,000 depending on their accuracy and functionality. The largest component of the system price is the level of global positioning system (GPS) accuracy desired.

There has also been a recent increase in the number of electronic components on agricultural equipment. During normal field situations, operators must interact with spray rate controllers, variable rate controllers, and implement system controllers, as well as controls for normal vehicle operation. Attempts have been made to create a standard communication link within all agricultural equipment but have thus far failed within the U.S. The most common ideology for a worldwide equipment communication protocol is the International Organization for Standardization 11783 standard (ISO, 1998), which was designed for agricultural and construction equipment. This standard utilizes the controller area network (CAN) 2.0B protocol to transmit serial data between networked control system components (Stone et al., 1999b). The CAN system was designed to link multiple electronic control units (ECU) over a single data bus and inherently lends itself to becoming the backbone of a distributed control system for autonomous vehicle operations because of its high data transmission rate, expandability, and reliability (CAN-CIA, 2002).

Robert Bosch GmbH designed the controller area network in 1986 upon request by Daimler Benz to develop a system that would allow for communication between three electronic control units. It was noted that standard UART communication could not complete this task because it only allowed for point-to-point communication. Although the CAN bus was originally designed for automotive applications, it has been applied to many areas of automation and control (CAN-CIA, 2002). The unique aspects of a CAN network are that each message is preceded by an identifier that is unique to the transmitting controller, and that multiple controllers can communicate over a single two-wire bus. If two messages are transmitted simultaneously, an automatic arbitration process ensures that the highest priority message is transmitted first. The lower priority message then has the opportunity to retransmit upon completion of the first message (Bosch, 1991).

The International Standards Organization (ISO, 1998) 11783 standard was created to standardize electronic communication on agricultural tractors and equipment. The ISO 11783 standard comprises 13 documents outlining specifications for the physical communication layer, network layout, and message prioritization, among others. CAN 2.0B was chosen for the communication protocol because of its growing use in mobile equipment. The ISO 11783 standard extends the definition of the CAN 2.0B protocol and specifies

Article was submitted for review in September 2004; approved for publication by the Power & Machinery Division of ASAE in February 2005. Presented at the 2004 ASAE Annual Meeting as Paper No. 041155.

Mention of trade name, proprietary product, or specific equipment does not constitute a guarantee or warranty by The Ohio State University or the University of Kentucky and does not imply the approval of the named product or the exclusion of other products that may be suitable.

The authors are **Matthew J. Darr, ASAE Student Member,** Research Associate, Department of Food, Agricultural, and Biological Engineering, The Ohio State University, Columbus, Ohio; and **Timothy S. Stombaugh, ASAE Member Engineer,** Assistant Professor, and **Scott A. Shearer, ASAE Member Engineer,** Professor, Department of Biosystems and Agricultural Engineering, University of Kentucky, Lexington, Kentucky. **Corresponding author:** Matthew J. Darr, 590 Woody Hayes Drive Room 212, Columbus, OH 43210; phone: 614-292-1406; fax: 614-292-9448; e-mail: darr.27@osu.edu.

  ISSN 0001−2351

many parameters concerning the serial communication and hardware connections. Much of ISO 11783 follows the exact specifications of the Society of Automotive Engineers standard J1939 (SAE, 1998) in an attempt to make the two standards compatible (Stone et al., 1999b).

Several manufacturers have implemented partial networks or CAN-based systems on their equipment in the past several years. The Genesis series tractor from New Holland (New Holland, Pa.) incorporated a CAN-based network in 1994 (Young, 1993). Ag-Chem Equipment Company (Duluth, Ga.) patented the use of network-based control systems for multi-product applications, which have been used in their Falcon control systems (Monson and Dahlen, 1995). Deere and Company (Moline, Ill.) has implemented ECU-based control networks since the introduction of the 7000 series tractor in 1992 (Stone et al., 1999b).

Computer simulations have shown that normal agricultural machinery configured with an ISO 11783 based CAN bus will produce average message latency of less than 6 ms with proper prioritization. When bus loads increase to 80% capacity, messages with low priorities could see latency times approaching 70 ms (Hofstee and Goense, 1999). Hofstee and Goense (1999) also showed that average message latency associated with passing through a network bridge during normal bus load was between 2.5 and 2.6 ms, with a maximum latency of 3.6 ms for messages with a high priority. This indicates that the CAN system is suitable for the low-latency data communications required in autonomous vehicle control.

The CAN network has the capability to serve as the backbone of a distributed control system on an automated machine. Distributed control systems have been used successfully in several sub-network functions of automatic equipment. Tian et al. (1999) used a distributed control system to identify individual weeds within row crops and operate appropriate control systems to apply pesticide to the weeds. Similarly, Stone et al. (1999a) used a CAN-based distributed control system to control the application rate of liquid fertilizer depending on the fertility of wheat, as determined from a near-infrared reflectance sensor.

Researchers and agriculturalists have pursued automatic vehicle control for many years. In fact, patents dating back to 1924 detail methods of automatic guidance in furrow strips (Willrodt, 1924). With the advent of precision agriculture and GPS capabilities, vehicle-based guidance again became a focus of research efforts. Researchers at the University of Illinois showed that 16 cm steady-state error straight-line accuracy could be achieved while traveling at speeds up to 6.8 m sec^{-1} (Stombaugh et al., 1999). Benson et al. (1998) showed that at slower speeds the addition of a geomagnetic direction sensor (GDS) could reduce the straight-line steady-state error to 1 cm.

While research has been completed on autonomous vehicle guidance, few studies have focused on end-of-row turning methods. Noguchi et al. (2001) showed that a spline function can be used to estimate the turning function of a vehicle at the end of a row. Other work has discussed the importance of full field operations, but has not addressed the problem of vehicle guidance during turning (Han and Zhang, 2001).

Objectives

The goal of this project was to evaluate the potential of a CAN bus to be used as the communication network for a distributed control system on an autonomous agricultural vehicle. This project also evaluated an inexpensive compass as a feedback tool for in-field turning operations. Whenever applicable, the ISO 11783 standard was followed during this project. These project goals were accomplished through implementation of the following sub-objectives:

- Design a modular distributed control system.
- Interface the electronic control units on the CAN bus with sensors and controls specific to the autonomous operation of a prime mover vehicle.
- Develop a digital steering control system for the autonomous test vehicle.
- Demonstrate the ability of the vehicle to complete a typical field operation.

Design of a Modular Distributed Control System

The requirements for the distributed control system on the autonomous vehicle were as follows:

- A dedicated node was to be located at every critical control location on the vehicle.
- All nodes were to be able to communicate to all other nodes in a multi-master communication system.
- Each node was to have the capability to implement control routines and interface with feedback sensors.
- An RS232 interface was to be used for task controller interaction with the CAN bus.

The CAN 2.0B protocol was chosen as the communication network for the distributed control system. Not only did the CAN system fully satisfy the objective of a multi-master network, but it also enabled simple implementation of dedicated control systems at each node location. CAN systems have also been shown to be reliable when used in harsh operating environments, such as those found on automotive or agricultural vehicles. Many microcontrollers were available with internal CAN engines, which simplified implementation and kept the overall system cost low. A bus baud rate of 250 kbits sec^{-1} was selected to maintain compliance with the ISO 11783 standard.

The microcontrollers required for this project had to execute control routines and interface to external sensors to fulfill the distributed control requirements. Specifically, the microcontroller had to have the following capabilities:

- Minimum of 4 Kbytes of program memory.
- Minimum of 1 Kbyte of RAM.
- Minimum of three channels of 10-bit analog to digital conversion.
- Internal CAN engine.
- Hardware pulse width modulation.
- Universal asynchronous receive and transmit (UART) port.

The microcontrollers utilized in the distributed control system were PIC 18F258 microcontrollers produced by Microchip (Chandler, Ark.). This chip provided capabilities that met or exceeded the requirements of the project, including an internal CAN module, five channels of 10-bit analog to digital conversion, 25 mA sink/source current loads, 8 × 8 single-cycle hardware multiplier, 16-bit

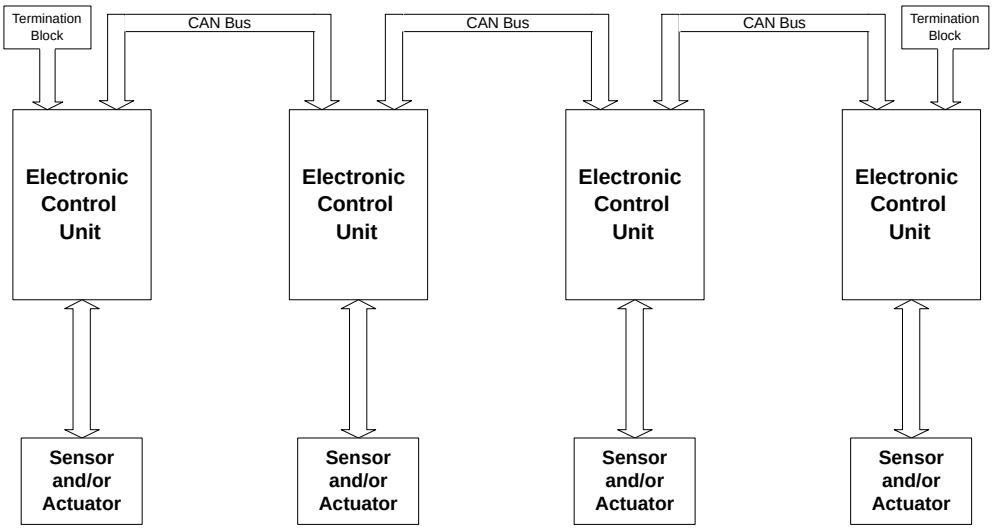

Figure 1. Controller area network layout.

counter/timer, hardware pulse width modulation, 32,000 bytes of FLASH memory, 1600 bytes of SRAM, and 256 bytes of EEPROM. An MCP2551 transceiver, also produced by Microchip, was incorporated into each node to provide switching between the digital TTL logic of the microcontroller and the differential output required on the CAN bus. The transceiver also acted as a buffer on the ECU to prevent transient voltage spikes on the CAN bus from reaching the microcontroller. The slew rate was also controlled via an external resistor network on the transceiver. The slew rate setting for a particular bus is based on the length and nominal voltages on the bus. Because a short bus length and a low nominal bus voltage were utilized in this project, no slew rate control was implemented. Limited bus fault protection was incorporated into the transceiver. The transceiver disabled the CAN output lines if an extended low-voltage state was sensed on the transmit pin. This prevented the bus from being corrupted with bad data if one of the ECUs malfunctioned. The transceiver reinitiated on the first rising edge of the transmit pin.

Two options were available to link each ECU together. One option consisted of linking each ECU via a single connector attached to the CAN bus. This would reduce the overall connector cost, but it would require that the precise location of each node be specified to designate the spacing between each connector. A second option was used for this project. Two connectors were incorporated into each ECU to link to the CAN bus. The CAN bus lines were then connected internally in the node to allow for a continuous bus (fig. 1). The final ECU design (fig. 2) included two CAN bus connections on one end of each enclosure. The opposite end of the enclosure housed one or two connections for interfacing with external sensors.

Machine code for the microcontroller was compiled using the PIC Basic Pro Compiler and written to the microcontroller using the EPIC programmer; both packages are products of MicroEngineering Labs, Inc. (Colorado Springs, Colo.).

A CAN-to-RS232 bridge was designed to interface the CAN messages with a task computer. The bridge was designed by configuring a specialized high-speed serial communication ECU. The bridge node had a single objective: to receive all messages from the CAN bus at a baud rate of 250 kbits sec^{-1} and retransmit the message via an RS232 link to the handheld data collection unit at a baud rate of 256 kbits sec^{-1}. By retransmitting the message at a rate faster than the incoming messages, the microcontroller ensured that the CAN receive buffers would not overflow with incoming messages. The CAN messages were retransmitted via RS232 beginning with a leading identifier ($), followed by the source address of the message and the eight message data bytes in a comma-delimited fashion. The message concluded with an ending identifier (#).

Initial testing validated the ruggedness and repeatability of the CAN-based control system. Four ECUs were linked together over the CAN bus and programmed to transmit data at a rate of 0.5 Hz. A handheld GPS receiver was used in simulation mode to transmit NMEA strings while indoors. Three nodes were designed to read independent analog voltages from potentiometers attached to each node. The potentiometers were connected as a single-ended input on each node. A simple code set was implemented on each node to collect analog input data and transmit the measured values on the CAN bus (fig. 3).

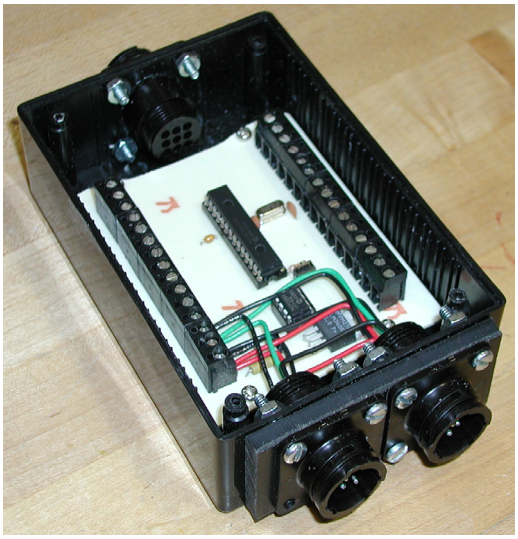

Figure 2. Electronic control unit.

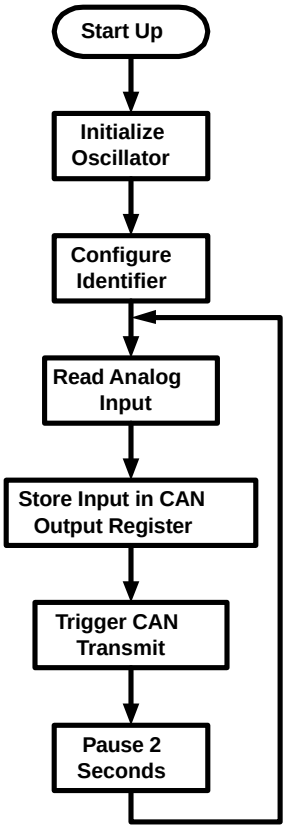

Figure 3. Microcontroller program flowchart used during initial testing.

The fourth node was a GPS node. This node was unique in that data were acquired through the on-chip RS232 serial port rather than the analog to digital converter. In addition, no Pause statement was needed since the 0.5 Hz timing was triggered by the output from the GPS receiver.

The message transmissions were logged by the task computer via the CAN-RS232 bridge. The incoming data messages were collected, time-stamped, and stored to a comma-delimited file. The four transmitting ECUs and one CAN-RS232 bridge were tested for 24 h and were subjected to random adjustments of the input voltages. The data messages were post-processed to determine the number of transmission errors that occurred within a 24 h period. Results showed no transmission errors were found to have occurred over the 24 h period. The period of the stored data was only 2 s, so it is possible that transmission errors did occur, but that they were recognized by the ECUs and the errant messages were retransmitted.

A secondary test was performed to determine the robustness of the CAN bus to external noise sources. A radio controller, similar to those used in remote aircraft flight, was placed adjacent to the CAN bus network cabling. An oscilloscope was used to monitor the noise level on the bus. The oscilloscope showed noise present on both CAN bus wires, but the messages still transmitted without errors (fig. 4). At the maximum, the noise signal reached a peak-to-peak magnitude of 5 V. This test demonstrated the robustness of CAN's low-voltage differential signaling.

Sensor Interfacing

With the initial testing of the modular network completed, the next project objective was to interface each of the ECUs with sensors specific to autonomous vehicle guidance. This involved developing a list of sensor requirements to provide an appropriate level of functionality to the vehicle. The test

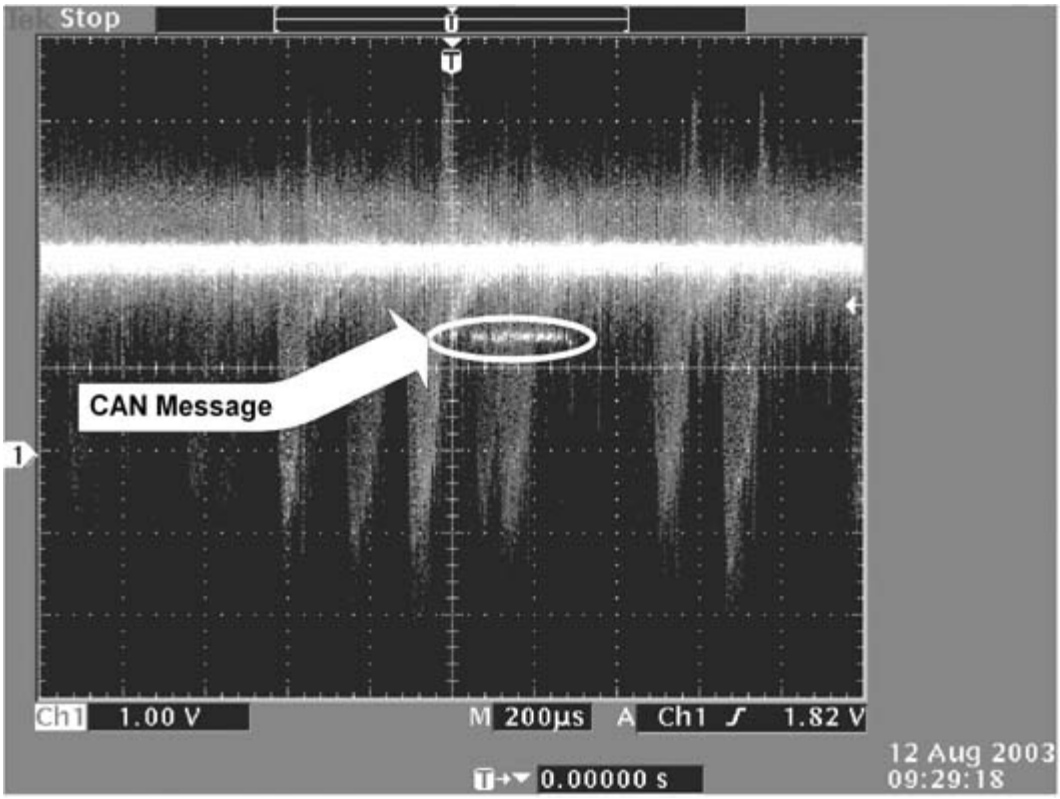

Figure 4. Oscilloscope screen capture showing voltage on CAN_L communication line during high noise conditions.

Figure 5. Test vehicle for autonomous guidance.

vehicle for this project was an 18.6 kW hydrostatic drive tractor (fig. 5). All driver amenities were removed to reduce the gross vehicle weight and to lower the vehicle center of gravity.

The following list outlines the minimum required control and sensing points to convert the test vehicle from manual operation to a limited-functionality autonomous machine:

- Electronic control of the steering axle with position feedback.
- Electronic control of the hydrostatic transmission swash plate with position feedback.
- Electronic control of the three-point hitch control lever with position feedback.
- Acquisition of vehicle position from a GPS receiver.
- Acquisition of vehicle heading from a digital compass.

The linkages on the vehicle that controlled the hydrostatic transmission swash plate position, three-point hitch control lever, and front-end steering were replaced with electric linear actuators. Each actuator contained a 10 kΩ linear potentiometer to provide position feedback. Each ECU responsible for executing a control command via an electric actuator was interfaced with the position potentiometer through the internal analog to digital converter on the microcontroller. Initial calibrations were performed to determine the maximum operating ranges for each of the actuators.

A sub-meter accuracy GPS receiver with WAAS correction provided vehicle positioning information. An ECU was designed to monitor the NMEA output of the GPS receiver via an RS232 connection and capture the GGA data string at a rate specified by the task computer. The latitude, longitude, time, GPS quality indicator, and number of satellites in use were each transmitted onto the CAN bus upon reception.

The steering actuator was calibrated to relate the count value acquired from the analog to digital conversion to the specific steering angle of the vehicle. The vehicle was subjected to a series of tests in which the steering position was held constant at a known feedback setting. While the vehicle traversed the resulting circular path, GPS data points were acquired by the task computer. The vehicle completed three circular paths for each steering position tested. The test was performed on a firm sod surface. The data points were post-processed in a GIS package to determine the mean diameter of each circular path (fig. 6).

A kinematic model was used to define the vehicle's steering angle based on the radius of the circular path (Grovum and Zoerb, 1970). The model showed that the radius of each circle was directly related to the steering angle of the vehicle:

$$\phi = \arctan\left(\frac{WB}{R1}\right) \quad (1)$$

where

ϕ = steering angle (degrees)
$R1$ = effective turning radius of the vehicle (m)
WB = vehicle wheelbase (m).

An equation relating the actuator feedback counts, acquired by a 10-bit analog to digital converter internal to the

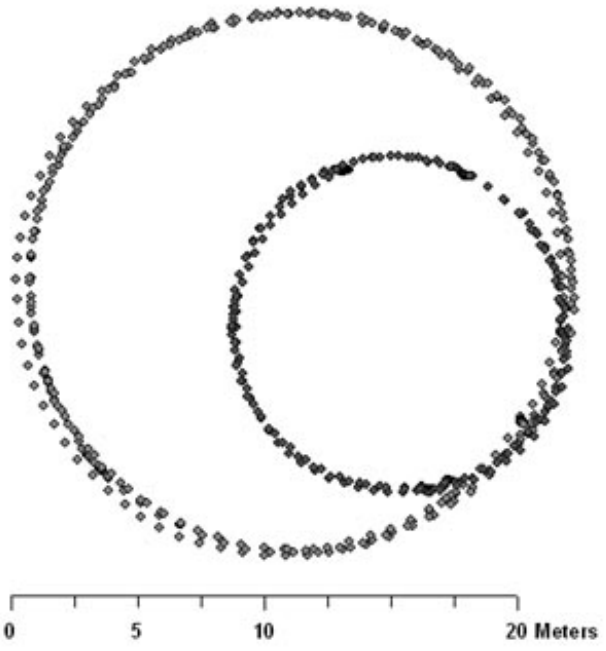

Figure 6. Steering actuator calibration.

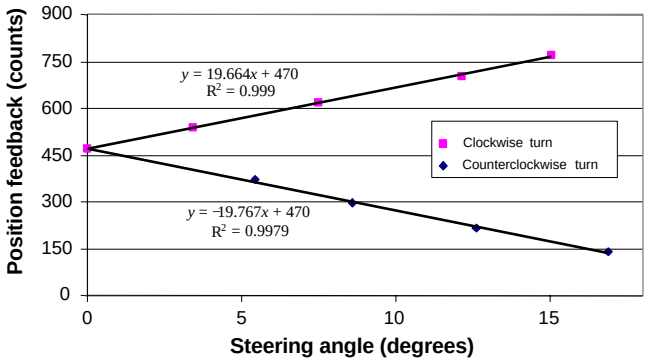

Figure 7. Steering axle actuator calibration.

node microcontroller, to the actual steering angle of the vehicle (eq. 2) was determined from equation 1 and the results of the circular path tests:

$$P = 19.70\phi + CenterPosition \quad (2)$$

where P is the actuator position (counts).

For these tests, the value of *CenterPosition* was 470. Regular recalibration was required due to drift in the steering axle position potentiometer. A linear regression was performed to determine the accuracy of the steering angle equation (fig. 7). The R^2 values for the linear regressions were 0.998 and 0.999 for counterclockwise and clockwise turns, respectively.

Due to the large current requirements of each linear actuator in the system, the microcontroller itself could not supply the necessary power to excite the actuator motor. An H-bridge motor controller was installed to isolate the ECU from the high current and voltage loads necessary to drive the actuators. The transmission and hitch actuators were driven by an LMD18201 H-bridge produced by National Semiconductor (Santa Clara, Cal.), and the steering actuator was driven by an OSMC3 (www.robot-power.com) motor controller. Software compensation and electronic braking were used to minimize actuator overshoots.

Vehicle heading information was acquired using a simple board-level digital compass. This allowed for sampling of the heading state at 4 Hz, as opposed to the 2 Hz sampling capabilities when using GPS information. A Vector 2X digital compass produced by Precision Navigation (Santa Rosa, Cal.) was chosen and interfaced with the steering ECU. The steering ECU was chosen as the interface node because of its proximity to the front of the vehicle. The sensor was a low-cost, 2-axis compass and magnetic sensor that provided accuracy of 2˚ for vehicle heading at a resolution of 1˚. The sensor utilized a patented magneto-inductive magnetometer (U.S. Patent No. 4,851,775). The sensor was calibrated to remove hard-iron distortions in the form of a constant offset that resulted from the host vehicle. Errors from non-constant magnetic fields from objects such as AC motors could not be calibrated out of the system and were physically avoided. This was not a long-term complication, but rather a known limitation that was readily accounted for in the initial design. Additional noise suppression was incorporated by mounting the compass 12 inches in front of the forwardmost part of the vehicle on a non-magnetic mount. The sensor was interfaced to the steering node via an SPI communication link.

The final network layout (fig. 8) contained seven ECUs. The steering, three-point hitch, and transmission nodes were the actuation points for the distributed control system and were required to accept a control setpoint and implement a feedback control routine to correctly position the actuator. The ECUs were programmed to continuously run the same control routine; thus, they monitored the CAN receive buffers for a new setpoint message and conducted a feedback control routine to maintain their actuator at the desired location (fig. 9).

Development of a Guidance Algorithm for Autonomous Vehicle Control

Each of the sensors and nodes depicted in figure 8 was installed on the test vehicle (fig. 5). CAN bus messages were defined to transfer feedback data into the task computer and transmit control commands to the ECUs. All messages required for autonomous vehicle operation were sent each time a new GPS message was received. The GPS update rate throughout the guidance system testing was set at 2 Hz, rather than the 0.5 Hz used for initial validation testing. Average bus load during normal transmission with a bus speed of 250 kbits sec^{-1}, a GPS signal rate of 2 Hz, and an average message length of 150 bits was only 0.72%, which allowed for future expansion of the system.

The vehicle ground speed during autonomous testing was set at 3 km h^{-1} (1.86 mph), which resulted in a vehicle travel distance of 0.467 m (1.37 ft) per control message when operating at a 2 Hz update rate. The identifier priority levels used for this project correspond to the message priority levels defined in ISO 11783 Section 3.

With the CAN-RS232 bridge installed, a task computer was required to link the CAN-based control system with a process control system. A Pentium 4, 1200 MHz laptop computer served as the task computer. The task computer ran an executable file that was written and compiled using Microsoft Visual Basic 6.0. The program received information from the CAN bus through a CAN-RS232 bridge node. Incoming messages were decoded based on the source address of the sender and used for vehicle guidance (fig. 10).

The implementation of automated straight-line guidance control by the task computer can be broken into five specific steps:

- Initialize the system by defining a desired path of travel.
- Receive feedback information from the GPS receiver through the CAN bus.
- Determine the appropriate steering axle position for autonomous guidance.
- Transmit the appropriate control commands to the steering, transmission, and three-point hitch ECU.
- Log position and control data for post-processing and evaluation.

To accomplish the first step, the task computer was given initial geographic points A and B. The task computer created a line between the two defined points and made a guidance decision based on this desired line of travel. These two points, as well as all subsequent locations received from the GPS receiver, entered the task computer in WGS 84 geographic coordinates and were projected to Universal Transverse Mercator (Zone 16N) Cartesian coordinates. The geographic coordinate data contained ten significant digits; thus, a resolution of 1 mm was available after the UTM conversion.

Each time a new GPS message was received, the task computer calculated the current vehicle error from the

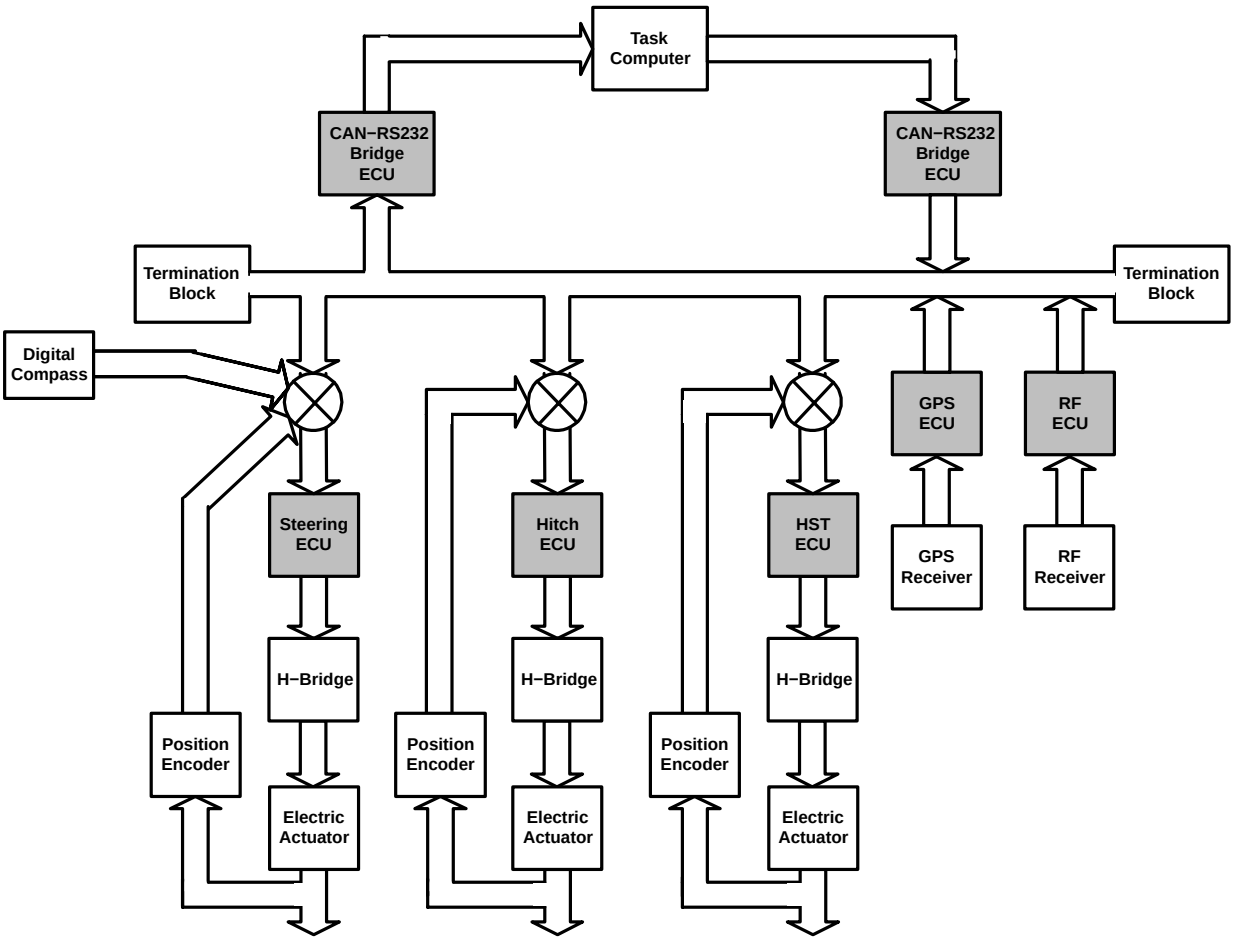

Figure 8. Final network layout for autonomous vehicle navigation control.

desired line of travel and applied a digital control routine to determine the new guidance control command. This command was transmitted through the CAN-RS232 bridge and was received by the steering node. During the latency time between incoming GPS messages, the task computer recorded position and control parameters to a hard disk for future evaluation.

The statistical parameters used to describe the performance of the control system were: mean error value, standard deviation of the error, and the mean 95% confidence interval. These statistical parameters were calculated during the steady-state portion of the test. Steady state was defined to begin when the vehicle performance settled to within 2% of the input offset amplitude. The target accuracy of the system

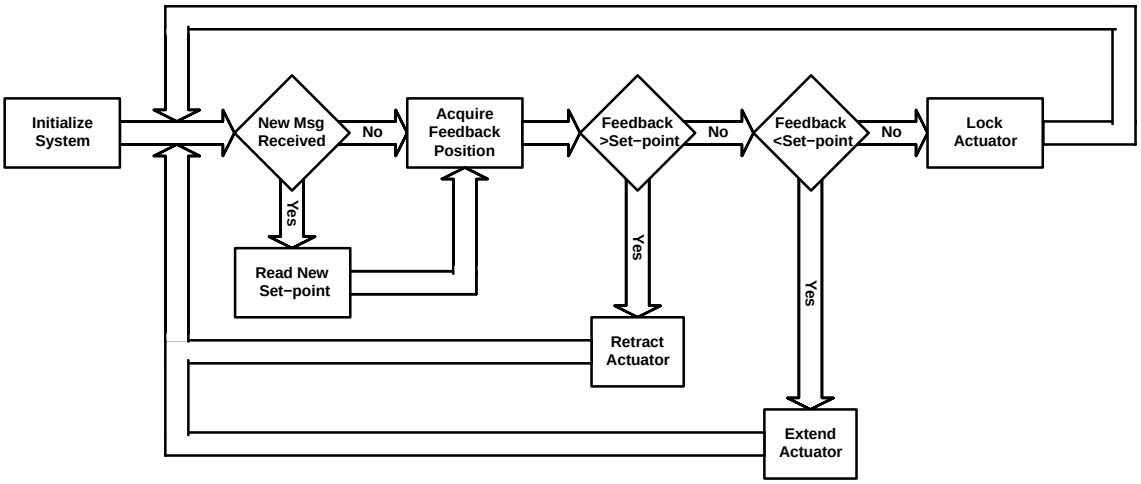

Figure 9. ECU control routine.

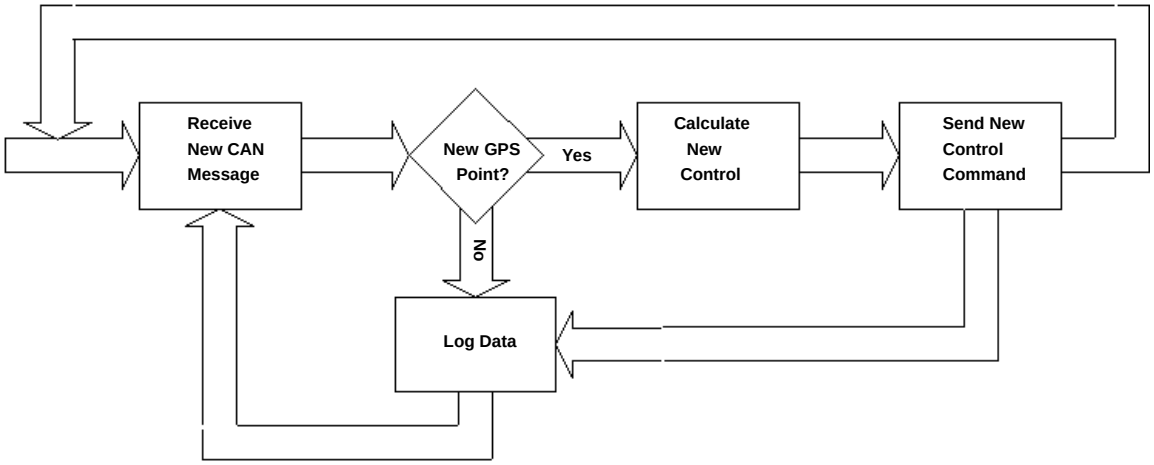

Figure 10. Task computer control program flowchart.

was a mean error value of nearly zero along with a 95% confidence interval of no more than 5 cm. Julian (1971) showed that lateral deviations of 5 cm or less at the front wheel would produce negligible deviations at the rear wheel. The accuracy measurements were relative to the accuracy of the GPS receiver used for field tests; thus, they were not absolute accuracies but rather a reflection of vehicle performance relative to the position indication.

Position-Based Digital PID Controller

A simple digital PID model was developed to minimize the computational load of the task computer, provide accurate control commands, and incorporate gain scheduling. Digital implementation of a PID controller can be derived in many forms. Multiple forms were considered and analyzed based on their relevance to this application. The chosen model (eq. 3) was based on a position PID controller, which transforms the analog PID model into a digital form by using rectangular integration approximation. The general form of the controller accounted for the current position error as well as the previous two position errors. An auxiliary term was added to the control algorithm to account for the center position of the steering actuator. The sampling time was held constant at 2 Hz and was lumped into the values for the gains. To offset noise present in the GPS position data, each new error value was calculated by averaging the previous three error points. Thus, E_K was the average of the current and two previous error points (e_k, e_{k-1}, and e_{k-2}), and E_{K-1} was the average of e_{k-1}, e_{k-2}, and e_{k-3}.

$$C_{PID} = CP + P_G[E_K] + I_G[I_K] + D_G\left[\frac{E_K - E_{K-1}}{T_S}\right] \quad (3)$$

where

C_{PID} = current control signal
CP = center location of steering actuator
E_K = current average error (cm)
E_{K-1} = previous average error (cm)
I_K = current integral error (= I_{K-1} + E_K [T_s], cm-sec)
P_g = proportional gain
I_g = integral gain
D_g = derivative gain
T_s = sampling time (sec).

As equation 3 shows, when all error values are driven to zero, the output is simply the calibrated value for the center location of the steering actuator; thus, the vehicle will travel straight.

Gain scheduling was incorporated to reduce the effects of large gain values when the offset error was small. Initial testing data were used to determine the scheduling parameters. The derivative gain value was set to zero whenever the value of ($E_K - E_{K-1}$) became less than 0.5 cm. Likewise, the integral gain was not activated in the system unless the value of E_K was less than 15 cm.

This algorithm was implemented on the test vehicle and used to guide it through a step response based on an initial vehicle offset of 5 m and a forward ground speed of 3 km h^{-1}. The P_g and D_g gain variables were tuned to stabilize the vehicle response. It was found that a proportional gain of 0.75 and a derivative gain of 7 produced a damping ratio of 0.65 (fig. 11). According to Doebelin (1998), a damping ratio (ζ) of 0.65 is often used for second-order systems that require a quick, yet stable, response. Steady state was reached within a 15 m travel distance (fig. 11).

There was still a steady-state error of 0.11 m apparent in the step response curve. Several values of integral gains were tested to reduce the amount of steady-state offset error (fig. 12). As expected, the standard deviation of steady-state errors increased with an increase in integral gain. An integral gain of 0.8 was chosen for this application. This provided a mean offset error of 0.34 mm and a standard deviation of 28.7 mm. These performance parameters were within the desired operating characteristics for an autonomous field vehicle.

Control System Validation Testing at Increased Ground Speed

All previous tests were conducted at a ground speed of 3 km h^{-1}. Further testing using the digital PID controller design was conducted at a forward ground speed of 6.12 km

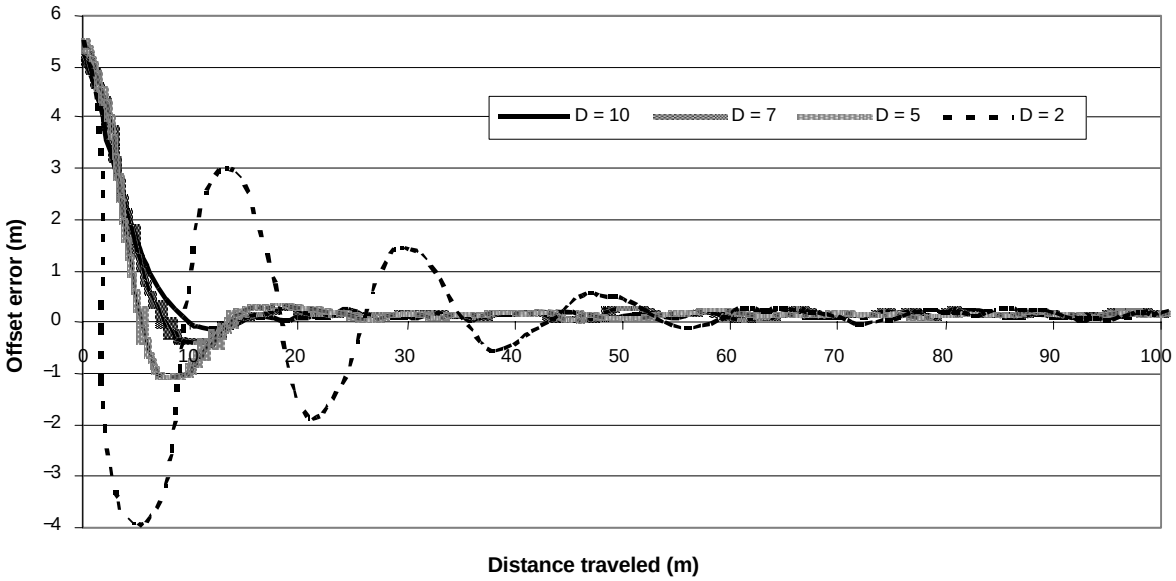

Figure 11. Step response curve for P_g = 0.75 and various D_g gains.

h^{-1}. Integral gains from 0.6 to 0.9 were used to evaluate any differences in the system response caused by a higher ground speed. The proportional gain was held at 0.75 and the derivative gain at 7 for all tests.

Analyses showed that at a higher ground speed of 6.12 km h^{-1}, an integral gain of 0.8 was still optimal (fig. 13). This produced a steady-state offset error of 0.05 mm and a standard deviation of 32.3 mm. These error values were well within the design limits of the project and showed that the control system could be applied to a faster autonomous vehicle. Further high-speed testing could not be completed due to a maximum forward speed of 6.12 km h^{-1} of the test vehicle.

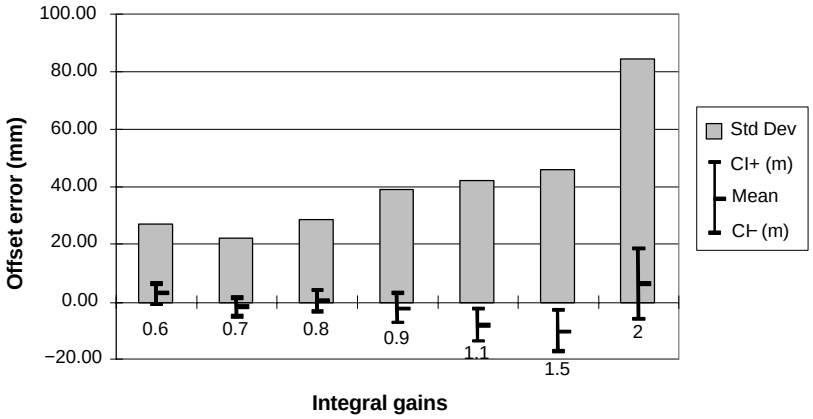

Figure 12. Steady-state performance of PID controller for various integral gains at a vehicle forward speed of 3 km h^{-1}.

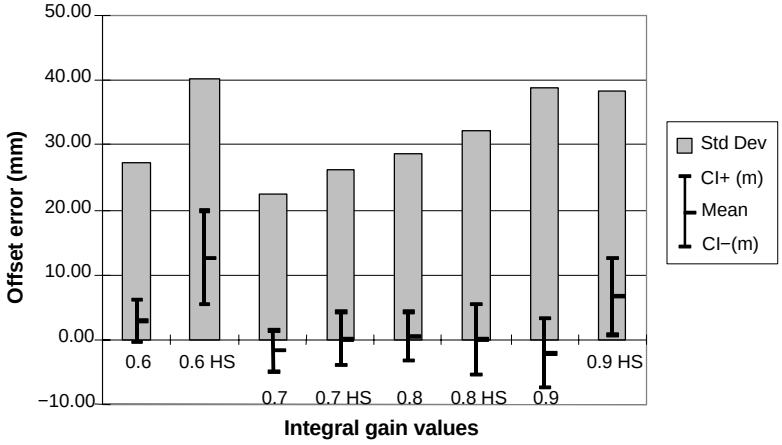

Figure 13. PID controller step response data for various integral gains at a forward ground speed of 6.12 km h^{-1}.

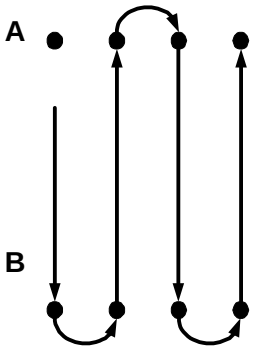

Figure 14. Field operation schematic.

Demonstrating the Ability to Traverse a Normal Field Operation

The final objective of the project was to demonstrate the autonomous capabilities of the vehicle by completing consecutive headland turning and straight-line guidance routines. The parallel swath pattern used to test the vehicle's autonomous capabilities (fig. 14) represented a common path in production agricultural operations and demonstrated the feasibility of the vehicle for this application. Only the initial two guidance points and the swath width between the operating passes were supplied to the vehicle.

The following specific tasks were completed to demonstrate successful autonomous field operation:

- Develop an algorithm to determine when the headland area had been reached.
- Develop an algorithm to calculate the next guidance line based on the current guidance line and defined vehicle swath width.
- Utilize a digital compass to implement an open-loop headland turning routine.

A simple algorithm was developed to determine when the headland area had been reached. The first step in the algorithm was to determine whether the vehicle was heading towards point A or point B. Then, the distance from the current location to the origination point and the distance between the two initial setpoints were compared. If the vehicle had passed into the headland area, then the distance between its current location and the point that originated the line of travel was greater than the distance between the two points used to define the line of travel.

The next step in completing a field operation required that the task computer be able to calculate the next desired guidance path. Two points, A and B, defined the initial path. When the vehicle extended beyond either of these end points, as discussed above, the control system redefined points A and B based on the set swath width of the field operation (eqs. 4 to 7). This newly defined line was parallel to the initial line and contained new A-B points that were perpendicular to the initial points (fig. 15).

$$B_{2x} = \left[\frac{B_{1y} + \frac{1}{m_1} B_{1x} - A_{1y} + m_1 A_{1x} + \frac{SW}{\cos\theta_1}}{m_1 + \frac{1}{m_1}} \right] \quad (4)$$

$$B_{2y} = \left[m_1 B_{2x} + A_{1y} - m_1 A_{1x} - \frac{SW}{\cos\theta_1} \right] \quad (5)$$

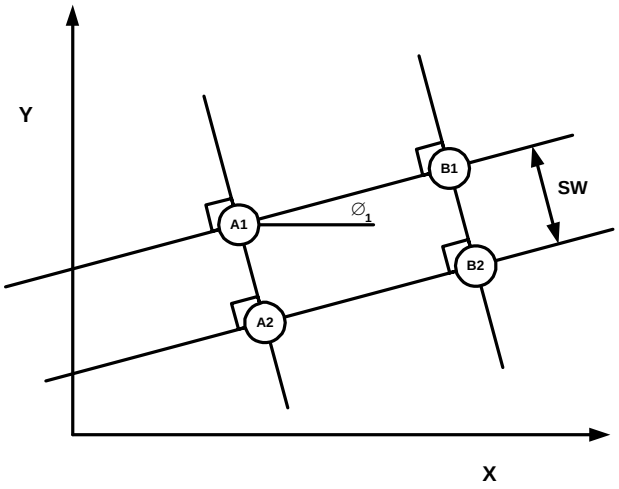

Figure 15. Guidance line translation.

$$A_{2x} = \left[\frac{A_{1y} + \frac{1}{m_1} A_{1x} - A_{1y} + m_1 A_{1x} + \frac{SW}{\cos\theta_1}}{m_1 + \frac{1}{m_1}} \right] \quad (6)$$

$$A_{2y} = \left[m_1 A_{2x} + A_{1y} - m_1 A_{1x} - \frac{SW}{\cos\theta_1} \right] \quad (7)$$

where

A_{1x} = initial point A easting
A_{1y} = initial point A northing
B_{1x} = initial point B easting
B_{1y} = initial point B northing
m_1 = initial slope of line AB
SW = swath width
θ_1 = $\tan^{-1}(m_1)$.

Open-loop steering control was used during specialized routines such as headland turning operations. When the end of the row was reached during a normal field operation, the vehicle was required to lift the rear implement, turn around, and begin tracking the next guidance line. To accomplish these functions, the task computer sent a single message specifying that the end of the row had been reached and specifying the direction and distance to the next guidance line. The steering ECU received this message and took appropriate action to steer the vehicle toward the next line. The vehicle maintained a constant steering angle while maneuvering the turn. The steering ECU monitored the digital compass and determined when the vehicle had changed direction by $180^\circ \pm 15^\circ$, thus marking the end of the headland turning routine. The steering ECU then notified all nodes that the headland turning operation had ended by transmitting a single CAN message.

While the vehicle maneuvered the turn, the task computer was able to calculate the new guidance line and prepared to resume autonomous guidance. The transmission node responded to the start and finish of the end of the row operation message by slowing the vehicle speed during the turning maneuver. In addition, the hitch node recognized the start and finish of the headland turning message and raised or lowered the implement appropriately.

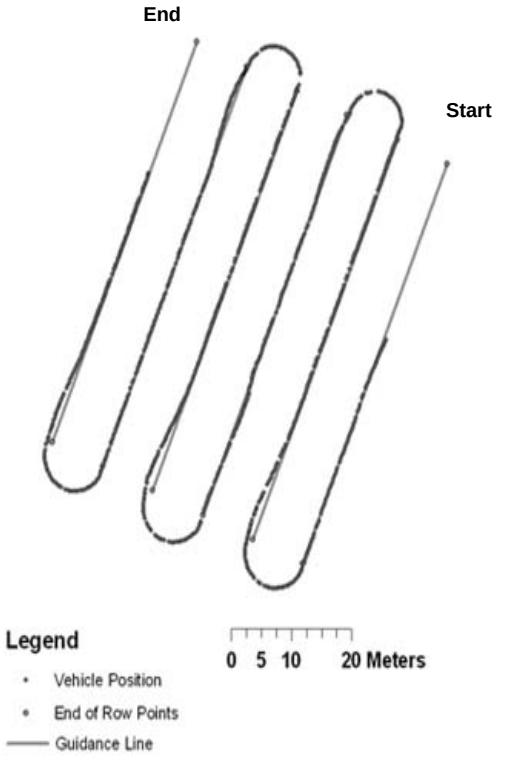

Figure 16. Field path demonstration.

The control system was tested in a field operation simulation. Two initial setpoints and a swath width were loaded into the task computer. The vehicle autonomously tracked the straight line between the points. It then determined when it had passed the farthest assigned point and began the standardized turning routine based on a given swath width. The vehicle successfully traversed the field and demonstrated appropriate turning functionality (fig. 16). It was found that while a deadband of ±15° was used to determine the end of the turning function worked well, it could have been reduced. During every recorded turning event, the vehicle released from the turning routine on the lower side of the deadband, or before a full 180° turn had been completed. All testing was conducted on a firm sod surface; thus, design modifications may be necessary when operating in low-traction environments.

Table 1. Cost of individual ECU components and CAN interface.

Component	Units Required	Unit Price ($)	Total Cost ($)
PIC 18f258	1	6.17	6.17
AMP connectors	6	5.75	34.50
28-pin socket	1	2.26	2.26
Printed circuit board	1	2.00	2.00
Screw terminal	2	1.75	3.50
Enclosure	1	1.75	1.75
MCP 2551	1	1.48	1.48
2 A fuse	1	0.84	0.84
20 MHz clock	1	0.75	0.75
8-pin socket	1	0.60	0.60
LM7805 - 1 A reg	1	0.50	0.50
Switch	1	0.43	0.43
LM78l05 - 0.1 A reg	1	0.40	0.40
Diode	1	0.27	0.27
Connector pins	54	0.26	14.04
22 pF capacitor	2	0.14	0.28
0.1 µF capacitor	1	0.08	0.08
Total			69.84

System Cost

The overall system cost was low relative to alternative designs for autonomous vehicle control (tables 1 and 2). Utilizing inexpensive microcontrollers at each node greatly reduced the data processing expense.

The overall unit cost per node was $69.84 (table 1). This included all components required to correctly operate the microcontroller and transceiver chips as well as all materials required to enclose the ECU. The connector cost included two CAN connections per box as well as a third connection for an auxiliary sensor. The costs of the female connectors that attached to the ECU male connectors were also included in the ECU analysis rather than developing a separate cost summary for interconnection hardware. Seventy percent of the overall cost of the ECU was directly related to the connectors. Practically, this is a high percentage to allocate to the physical connection between the node and the bus, but it was critical to use quality connections to ensure bus reliability.

The total amount allocated for all sensors and seven nodes was $11,520.88 (table 2). This cost could fluctuate substantially depending on the type of GPS receiver used. In addition, a less expensive single-board computer could replace the laptop task computer to further reduce cost.

Conclusions

A controller area network provided an efficient platform to develop an autonomous vehicle control system. Individual control nodes reduced the computational load of the task computer by implementing feedback control logic at the node. The CAN-RS232 bridge was an inexpensive means to transfer CAN-based data into a task computer for data logging and mathematical computation of autonomous control commands. A digital PID controller was sufficient in controlling the steering system of a 1.42 m wheelbase 18.6 kW tractor with a high degree of accuracy while traveling less than 6.12 km h^{-1}. Gain scheduling and GPS position filtering reduced the occurrence of impulsive commands during situations where the guidance errors were very low. When operated within the parameters of this project, the control system exhibited errors that were nearly

Table 2. Cost of system-wide sensors.

Component	Units Required	Unit Price ($)	Total Cost ($)
Sub-meter GPS receiver	1	3,000	3,000
Task computer	1	1,200	1,200
Steering actuator	1	750	750
Transmission actuator	1	150	150
Hitch actuator	1	150	150
High-current motor controller	1	85	85
Radio frequency controller	1	80	80
Digital compass	1	50	50
Piezo-electric rate sensor	1	25	25
IC H-bridge	2	13	26
Total			5,516

negligible. An open-loop headland turning routine was successfully implemented using a digital two-dimensional compass as the means of directional feedback.

REFERENCES

Benson, E. R., T. S. Stombaugh, N. Noguchi, J. Will, and J. F. Reid. 1998. An evaluation of a geomagnetic direction sensor for vehicle guidance in precision agriculture applications. ASAE Paper No. 983203. St. Joseph, Mich.: ASAE.

Bosch, R. 1991. CAN Specification, Version 2.0. Stuttgart, Germany: Robert Bosch GmbH.

CAN-CIA. 2002. Application of controller area networks. Erlangen, Germany: CAN in Automation. Available at: www.can-cia.de. Accessed 17 February 2003.

Doebelin, E. O. 1998. *System Dynamics: Modeling, Analysis, Simulation, Design.* New York, N.Y.: Marcel Dekker.

Grovum, M. A., and G. C. Zoerb. 1970. An automatic guidance system for farm tractors. *Trans. ASAE* 13(5): 565-573, 576.

Han, S., and Q. Zhang. 2001. Map-based control functions for autonomous tractors. ASAE Paper No. 011191. St. Joseph, Mich.: ASAE.

Hofstee, J. W., and D. Goense. 1999. Simulation of a controller area network-based tractor-implement data bus according to ISO 11783. *J. Agric. Eng. Res.* 73(4): 383-394.

ISO. 1998. ISO Standard 11783: Tractors and machinery for agriculture and forestry - Serial control and communications data network - Part 3: Data link layer. Geneva, Switzerland: ISO.

Julian, A. P. 1971. Design and performance of a steering control system for agricultural tractors. *J. Agric. Eng. Res.* 16(3): 324-336.

Monson, R. J., and E. M. Dahlen. 1995. Mobile control system responsive to land area maps. U.S. Patent No. 5453924.

Noguchi, N., J. Reid, Q. Zhang, and J. Will. 2001. Turning function for robot tractor based on spline function. ASAE Paper No. 011196. St. Joseph, Mich.: ASAE.

Reid, J. F., Q. Zhang, N. Noguchi, and M. Dickson. 2000. Agricultural automatic guidance research in North America. *Computers and Electronics in Agric.* 25(1-2): 155-167.

SAE. 1998. SAE Standard J1939: Part 7, Vehicle application layer. Warrendale, Pa.: SAE.

Stombaugh, T. S., E. R. Benson, and J. W. Hummel. 1999. Guidance control of agricultural vehicles at high field speeds. *Trans. ASAE* 42(2): 537-544.

Stone, M., D. Giles, and K. Dieball. 1999a. Distributed network systems for control of spray droplet size and application rate for precision chemical application. ASAE Paper No. 993112. St. Joseph, Mich.: ASAE.

Stone, M., K. McKee, C. Formwalt, and R. Benneweis. 1999b. ISO 11783: An electronic communications protocol for agricultural equipment. ASAE Distinguished Lecture 23. St. Joseph, Mich.: ASAE.

Tian, L., J. F. Reid, and J. W. Hummel. 1999. Development of a precision sprayer for site-specific weed management. *Trans. ASAE* 42(4): 893-900.

Willrodt, F. L. 1924. Steering attachment for tractors. U.S. Patent No. 1,506,706.

Young, S. C. 1993. Electronic control system for Genesis 70 series tractor. SAE Paper No. 941789. Warrendale, Pa.: SAE.