Iowa State University

From the Selected Works of Matthew J. Darr

August, 2004

CAN-based Distributed Control for Autonomous Vehicles

Matthew J. Darr, *Ohio State University*

Timothy S. Stombaugh, *University of Kentucky*

Scott A. Shearer, *University of Kentucky*

John P. Fulton, *Auburn University*

Available at: https://works.bepress.com/matthew_darr/15/

*The Society for engineering
in agricultural, food, and
biological systems*

*The Canadian Society for
Engineering in Agricultural,
Food, and Biological Systems*

An ASAE/CSAE Meeting Presentation

Paper Number: 041155

CAN-based Distributed Control for Autonomous Vehicles

Matthew J. Darr, Research Associate

Food, Agricultural, and Biological Engineering Department, The Ohio State University,
Columbus, OH 43210

Timothy S. Stombaugh, Assistant Professor

Department of Biosystems and Agricultural Engineering, University of Kentucky, Lexington,
Kentucky 40546

Scott A. Shearer, Professor

Department of Biosystems and Agricultural Engineering, University of Kentucky, Lexington,
Kentucky 40546

John P. Fulton, Assistant Professor

Biosystems Engineering Department, Auburn University, Auburn, Alabama 36849

**Written for presentation at the
2004 ASAE/CSAE Annual International Meeting
Sponsored by ASAE/CSAE
Fairmont Chateau Laurier, The Westin, Government Centre
Ottawa, Ontario, Canada
1 - 4 August 2004**

The authors are solely responsible for the content of this technical presentation. The technical presentation does not necessarily reflect the official position of ASAE or CSAE, and its printing and distribution does not constitute an endorsement of views which may be expressed. Technical presentations are not subject to the formal peer review process, therefore, they are not to be presented as refereed publications. Citation of this work should state that it is from an ASAE/CSAE meeting paper. EXAMPLE: Author's Last Name, Initials. 2004. Title of Presentation. ASAE/CSAE Meeting Paper No. 041155. St. Joseph, Mich.: ASAE. For information about securing permission to reprint or reproduce a technical presentation, please contact ASAE at hq@asae.org or 269-429-0300 (2950 Niles Road, St. Joseph, MI 49085-9659 USA).

Abstract. Through the work of researchers and the development of commercially available products, automated guidance has become a viable option for agricultural producers. One limiting factor of many commercially available technologies is that they only automate one function of an agricultural vehicle and that their control systems are proprietary to a single machine model.

The objective of this project was to evaluate a controller area network (CAN bus) as the basis of an automated agricultural vehicle. The prototype system utilized several microcontroller-driven nodes to act as control points along a system wide CAN bus. Messages were transferred to the steering, transmission, and hitch control nodes from a task computer. The task computer utilized global positioning system data to determine the appropriate control commands.

Infield testing demonstrated that each of the control nodes could be controlled 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. Controller Area Network, Distributed Control Systems, Autonomous Vehicle, Precision Agriculture, Embedded Microcontroller Systems

Introduction

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 autonomous vehicles ranging from planters to sprayers to harvesters. These vehicles have all sustained different levels of autonomous operation. Some were capable of fully autonomous field operations while others were developed for specific control operations such as autosteering (Reid et al., 2000). While these advances have been impressive, commercially available products have come at a substantial cost to the farmer. These products range in cost from \$20,000 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. Additionally, many systems developed for fully autonomous control rely on a single central processing unit to coordinate all vehicle operations and to carry out all control routines. This design places high demands on the processing unit, which in turn triggers high unit cost.

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 planter 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 US. The most common ideology for a worldwide equipment communication protocol is the International Organization for Standardization (ISO) 11783 standard, which was designed for agricultural and construction equipment. This standard utilizes the Controller Area Network (CAN bus) 2.0B protocol to transmit serial data between networked control systems (Stone et al., 1999b). The CAN bus 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).

Robert Bosch GmbH designed the Controller Area Network in 1986 upon request by Mercedes to develop a system that would allow for communication between three electronic control units. It was noted that a standard UART communication could not complete the 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 has been used in applications including warehouse shipping automation, packaging machines, medical devices including X-ray collimators and patient tables, and building controls including alarm and sprinkler systems (CAN-CIA). The unique aspect of a CAN network is that each message is preceded with an identifier that is unique to the transmitting controller and that multiple controllers can communicate over a single two-wire bus. Two wires are required for the node to assert the two different voltage levels defined by the CAN protocol. If two messages are sent simultaneously, an automatic arbitration process ensures that the highest priority message is sent first. The lower priority message then has the opportunity to retransmit upon completion of the first message. Incoming messages are filtered by the ECUs based on the unique message identifier of the sender (Bosch, 1991).

There are three separate CAN standards: CAN Version 1.0, Version 2.0A (Standard CAN), and Version 2.0B (Extended CAN). The main difference in the three standards is the length of the

identifiers that precede each message. All work presented in this publication is based on the CAN 2.0B standard.

The International Standards Organization (ISO) 11783 document was created to standardize electronic communication on agricultural tractors and equipment. The ISO 11783 standard is comprised of thirteen documents outlining specifications for the physical communication layer, network layout, and message priority 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 many parameters concerning the serial communication and hardware connections. Much of ISO 11783 follows the exact specifications of the Society of Automotive Engineers (SAE) standard J1939 in an attempt to make the two standards compatible (Stone et al., 1999b). SAE J1939 is an automotive network communication standard that also uses CAN 2.0B. The ISO 11783 standard was followed whenever applicable throughout this project.

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) also reported the use of a CAN-based network in 1994 (Young, 1993). Ag-Chem Equipment Company Inc. (Duluth, GA) patented the use of network-based control systems for multi-product application, which has been used in their Falcon control systems (Monson and Dahlen, 1995). John Deere (Moline, IL) 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 msec with proper prioritization. When bus loads increased 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 latency associated with passing through a network bridge during normal bus load was between 2.5 and 2.6 msec with a maximum latency of 3.6 msec for messages with a high priority.

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 researchers. Researchers at the University of Illinois showed that 16 cm steady state error straight-line accuracy could be achieved while traveling up to 6.8 m/sec (Stombaugh et al., 1999). Benson et al. (1998) showed that at slower speeds, addition of a geomagnetic direction sensor (GDS) could reduce the straight-line steady state error to 1 cm.

Machine vision has also been incorporated into control systems. Work by Carnegie-Mellon University and NASA demonstrated that vision-based systems could be used for autonomous control (Ollis and Stentz, 1996). A camera mounted on a hay-windrowing machine was used to identify the cut/uncut edge within the hay crop and guide the windrower to stay along this path. Benson et al. (2001) used a single cab mounted camera along with grain head mounted cameras to develop steering control systems for combine harvesters.

Row crop vision systems have been developed to identify individual plants and determine a path to stay within the plant rows. Near infrared sensors have been used to classify the soil and row

crops separately and provide a guidance directrix to a control algorithm (Reid and Searcy, 1987).

While research has been completed on autonomous vehicle guidance, few studies have focused on end of row turning methods. Noguchi et al. (2001) has shown 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).

Distributed control systems have been used 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 the wheat as determined from a near-infrared reflectance sensor.

Wei et al. (2001) proved that a CAN bus could be used as the foundation for a distributed weed control system. Successful testing was documented along with failures associated with error processing and recognition. Microcontroller use in distributed control systems was demonstrated by controlling low-pressure plant growth chambers at Texas A&M University (Brown and Lacey, 2002). Six microcontrollers were successfully integrated into a distributed control system to monitor and control pressure within a vacuum chamber based on a given set point.

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 field vehicle. This project also evaluated whether an inexpensive compass could assist with autonomous vehicle operations.

These project goals were accomplished through implementation of the following sub-objectives.

1. Design a modular distributed control system
2. Interface the electronic control units on the CAN bus with sensors and controls specific to the autonomous operation of a prime mover vehicle
3. Develop a digital steering control system for the an autonomous test vehicle
4. 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 were as follows:

1. A dedicated node was to be located at each control location.
2. All nodes were to be able to communicate to all other nodes in a multi-master communication system.
3. Each node was to have the capability to implement control routines and interface with feedback sensors.
4. An RS232 interface was to be used for task controller interaction.

There were several multi-master based communication systems available to use for this project including CAN, Inter-Intercomputer Communications (I2C), and Local Area Network (LAN). The CAN 2.0B protocol was chosen from this list 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 very 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 relatively low. A bus baud rate of 250 kbits/sec was used to maintain compliance with the ISO 11783 standard.

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

1. At least 4 Kbytes of program memory
2. At least 1 Kbyte of RAM
3. At least 3 channels of 10-bit analog to digital conversion
4. Internal CAN engine
5. Hardware pulse width modulation
6. Universal Asynchronous Receive and Transmit (UART) port.

The microcontrollers utilized in the distributed control system were PIC18F258 microcontrollers produced by Microchip (Chandler, AR). This chip provided capabilities that met or exceeded the requirements of the project, including an internal CAN module, 5 channels of 10-bit analog to digital conversion, 25 mA sink/source current loads, 8 x 8 single cycle hardware multiplier, 16-bit counter/timer, hardware pulse width modulation, 32,000 bytes of FLASH memory, 1600 bytes of SRAM, and 256 bytes of EEPROM.

The microcontroller provided the logic operations for CAN communication, but could not drive the differential voltages required for physical implementation of the communication protocol. 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 nominal CAN bus voltage was set based on the operating voltage of the transceiver chip. For this project, the nominal bus voltage was set at 2.5 volts, thus following the ISO 11783 specification.

The transceiver also acted as a buffer for the ECU and prevented transient voltage spikes on the CAN bus from reaching the microcontroller. The slope that the transmission bits rise and fall (slew rate) was also controlled via an external resistor network on the transceiver. The slew rate setting for a particular bus was 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.

The bus cable that connected each ECU consisted of nine wires. Four wires followed the physical layer standard of ISO 11783. Two additional wires provided a 12 volt supply voltage and ground to the ECUs. The three remaining wires were unused in the initial design, but provided potential for future expansion. The ECU power supply lines were insulated 16 gauge stranded copper wire. All other communication wires were 22 gauge twisted pair stranded copper wire.

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 would require 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 (Figure 1).

Figure 1. Controller area network layout

The final generic node design (Figure 2) included two CAN bus connections on one end of the enclosure. The opposite end of the enclosure housed one or two connections for interfacing with external sensors.

Figure 2. Electronic control unit

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

A CAN to RS232 bridge was designed to interface the CAN message information 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 and retransmit the message via an RS232 link to the handheld data collection unit at a baud rate of 256 kbits/sec. 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 Garmin (Kansas City, MO) 76 GPS receiver was used in simulation mode to provide 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 (Figure 3) was implemented on each node to collect analog input data and transmit the measured values on the CAN bus.

Figure 3. Microcontroller program flow chart used during initial testing

The GPS node was unique in that data were acquired through the on-chip serial port rather than the analog to digital converter. There was also no Pause statement for the GPS node. Its 0.5 Hz timing was controlled by the GPS output from the GPS receiver.

The message transmissions were logged via the CAN-RS232 bridge onto a task computer, thus modeling the final system. 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 hours 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 hour period. No transmission errors were found to have occurred over the 24 hour period. The period of the stored data was only 2 seconds, so it was possible that transmission errors did occur, but that they were recognized by the ECUs and the errant messages were retransmitted according to ISO 11783 protocol.

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 a tremendous amount of noise occurring on both CAN bus wires, but the messages still transmitted without errors (Figure 4). This test demonstrated the robustness of CAN's low voltage differential signaling.

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

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 vehicle for this project was an 18.6 kW hydrostatic drive tractor (Figure 5). Manual linkages were used to actuate the steering axle, transmission speed, and three-point hitch location. All driver amenities were removed to reduce the gross vehicle weight and to lower the vehicle center of gravity.

Figure 5. Test vehicle for autonomous guidance

The following list outlines the minimum required control and sensing points to convert the test vehicle from manually operated to a limited functionality autonomous machine.

1. Electronic control of the steering axle with position feedback
2. Electronic control of the hydrostatic transmission swash-plate with position feedback
3. Electronic control of the three-point hitch control lever with position feedback
4. Acquisition of vehicle position from a GPS receiver
5. Acquisition of vehicle heading from 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 10k 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. The supply voltage to the potentiometer was sourced from the node. By providing the supply voltage from the same source as the microcontroller, a single ended analog to digital conversion could be used with little worry of bias resulting from different ground potentials. Initial calibrations were performed to determine the maximum operating ranges for each of the actuators.

A Trimble (Sunnyvale, CA) AgGPS 214 GPS receiver with WAAS correction provided vehicle positioning with sub-meter accuracy. An ECU was designed to monitor the NMEA output of the 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

traversing a circular path, GPS data points were acquired by the task computer and recorded to a text file. The vehicle completed three circular paths for each steering position tested. The test was performed on a firm sod surface to reduce the effect of wheel side slip. The data points were post-processed in a GIS package to determine the mean diameter of each circular path (Figure 6).

Figure 6. Steering actuator calibration

A kinematic model was used to define the vehicle's steering angle based on the radius of the circular path (Grofum and Zoerb, 1970). The model showed that the radius of each circle was directly related to the steering angle of the vehicle (Equation 1).

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

Where: $R1$ = effective turning radius of the vehicle
 WB = vehicle wheel base

From Equation 1 and the multiple circular paths conducted during the calibration, an equation relating the actuator feedback counts to the actual steering angle of the vehicle was determined. This equation was as follows:

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

Where: P = Actuator Position (Counts)
 ϕ = Steering Angle in Degree

For these tests, the value of CenterPosition was 470. Regular re-calibration 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 (Figure 7). The R^2 values for the linear regressions were .998 and .999 for counterclockwise and clockwise turns, respectively.

Figure 7. Steering axle actuator calibration

Each actuator used in this design was a ball-screw linear actuator. Due to the large current requirements of each actuator, 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. An LMD18201 H-bridge produced by National Semiconductor (Santa Clara, CA) was used for the transmission and hitch actuators. This H-bridge could supply up to 3 amps of continuous current at up to 55 volts. The LMD 18201 did not have sufficient current capabilities for the steering actuator, which could draw up to 15 amps continuously. For this application an OSMC3 (www.robot-power.com) motor controller was used. This controller could provide well in excess of 15 amps continuously. Digital output pins on the ECU controlled the H-bridge circuits and allowed forward and reverse motion and locking of the electric actuators. It was found that the actuators would overshoot the desired location due to the inertia built up in the motor and ball-screw mechanism. H-bridge braking was incorporated into the ECU control routine, which eliminated actuator overshoot.

Several methods were available to acquire the current vehicle heading including computing a value based on the incoming GPS parameters. The most practical solution was to utilize a simple board level digital compass. This allowed for rapid sampling of the heading state (4 hertz) as opposed to the limited sampling capabilities of the GPS receiver. It also allowed for easier implementation because the microcontroller interfaced to the compass through a simple serial communication link.

A Vector 2X digital compass produced by Precision Navigation (Santa Rosa, CA) 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 and it could potentially use the heading information internally. 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. The sensor was calibrated to remove hard iron distortions in the form of a constant offset that resulted from the host vehicle. Non-constant magnetic fields from objects such as AC motors could not be calibrated out of the system and were physically avoided. Additional noise suppression was incorporated by mounting the compass 12" in front of the forward most part of the vehicle on a non-magnetic wooden surface. The sensor was interfaced to the steering node via an SPI communication link. Figure 8 depicts the complete network layout used to satisfy the system requirements.

Figure 8. Final network layout

The steering, three-point hitch, and transmission nodes were the actuation points for the distributed control system and were required to accept a control set-point and implement a feedback control routine to correctly position the actuator. The ECU was programmed to continuously run the same control routine, thus it monitored the CAN receive buffers for a new set-point message and conducted a feedback control routine to maintain the actuator at the desired location (Figure 9).

Figure 9. ECU control routine

Development of a Guidance Algorithm for Autonomous Vehicle Control

Each of the sensors and nodes depicted in Figure 8 were installed on the test vehicle. CAN bus messages were defined to efficiently transfer feedback data into the task computer and transmit control commands to ECUs. All messages required for autonomous vehicle operation were sent each time a new GPS message was received. The GPS update rate for this project was 2 Hz. Average bus load during normal transmission with a bus speed of 250 kbits/sec, a GPS signal rate of 2 Hz, and an average message length of 150 bits was 0.72%. This was very desirable and allowed for future expansion of the system.

The vehicle ground speed during autonomous testing was set at 3 km/hr (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 (Table 1).

Table 1. Message priority assignments (Reprinted from ISO 11783-3 5.2.1)

Priority Field Number	Priority Level	Default System
000	0 - Highest	
001	1	
010	2	
011	3	Control Systems
100	4	
101	5	
110	6	Proprietary Information
111	7 – Lowest	

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 (Redmond, WA) 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 (Figure 10).

Figure 10. Task computer flow chart

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

1. Initialize the system by defining a desired path of travel
2. Receive feedback information from the GPS receiver through the CAN bus
3. Determine the appropriate steering axle position for autonomous guidance
4. Transmit the appropriate control commands to the steering and transmission ECU
5. 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 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 accuracy 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 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 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.

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 method. 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.

$$C_{PID} = CP + [P_g \times E_K] + [I_g \times (E_K + E_{K-1} + E_{K-2})] + [D_g \times (E_K - E_{K-1})] \quad (3)$$

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)
- E_{K-2} = Second Previous Average Error (cm)

As equation 3 shows, when all error values were driven to zero, the output will simply be the calibrated value for the center location of the steering actuator, thus the vehicle will travel straight. In order to prevent random errors in the GPS signal from drastically affecting the vehicle response, three consecutive error values were averaged together and used as the current average error value. The two previous average error values were calculated in a similar manner, thus five data points were used to determine each steering control command.

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 meters. The P & D 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 (Figure 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 after a 15 meters travel distance (Figure 11).

Figure 11. Step response curve for $P = 0.75$ and various D gains

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 (Figure 12).

Figure 12. Steady state performance of PID controller for various integral gains

As expected, the standard deviation of steady state errors increased with an increase in integral gain (Figure 12). 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 0.83 m/s. Further testing using the digital PID controller design was conducted at a forward ground speed of 1.7 m/s. 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.

Figure 13. PID controller step response data for various integral gains and high ground speed

Analyses showed that at a higher ground speed of 1.7 meters per second, an integral gain of 0.8 was still optimal (Figure 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. Unfortunately, 1.7 m/s was the maximum forward speed of the test vehicle; thus, further high speed testing could not be completed.

Demonstrate 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 (Figure 14) represented a common path was 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.

Figure 14. Field operation schematic

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

1. Develop an algorithm to determine when the headland area had been reached
2. Develop an algorithm to calculate the next guidance line based on the current guidance line and defined vehicle swath width
3. 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 thus requiring initiation of a turning routine. 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 set-points were compared. If the vehicle had passed into the headland area, then the distance between its current location and the point which 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 (Equations 4 - 7).

This newly defined line was parallel to the initial line and contained new A-B points that were perpendicular to the initial points (Figure 15).

Figure 15. Guidance line translation

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

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

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

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

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
- $\theta_1 = \tan^{-1}(m_1)$

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 beginning 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 a digital compass and determined when the vehicle had changed direction by 180 ± 15 degrees, 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.

During the time it took the vehicle to maneuver 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. Also, the hitch node recognized the start and finish of the headland turning message and raised or lowered the implement appropriately.

This control system was tested in a field operation simulation. Two initial set-points 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 (Figure 16). It was found that while the deadband of $\pm 15^\circ$ 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 made.

Figure 16. Field path demonstration

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 with a high degree of accuracy. 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 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, MI.: ASAE.
- Benson, E. R., J. F. Reid, and Q. Zhang. 2001. Machine vision based steering system for agricultural combines. ASAE Paper No. 01-1159. St. Joseph, MI.: ASAE.
- Bosch, R. 1991. CAN Specification, Version 2.0. Stuttgart, Germany: Robert Bosch GmbH.
- Brown, D., and R. E. Lacey. 2002. A distributed control system for low pressure plant growth chambers. ASAE Paper No. 023078. St. Joseph, MI.: ASAE.
- CAN-CIA. 2002. Application of Controller Area Networks. Erlangen, Germany.: CAN in Automation. Available at: www.can-cia.de. Accessed 17 February 2003.
- DIN. 1993. Agricultural Tractors and Machinery; Interfaces for Signal Transfer. DIN Standard 9684. Berlin, Germany.
- Doebelin, E.O. 1998. *System Dynamics: Modeling, Analysis, Simulation, Design*. New York, NY: Marcel Dekker, Inc.
- Grovum, M. A., and G. C. Zoerb. 1970. An automatic guidance system for farm tractors. *Transactions of the ASAE* 13(5):565-573,576.
- Han, S., and Q. Zhang. 2001. Map-based Control Functions for Autonomous Tractors. ASAE Paper No. 01-1191. St. Joseph, MI.: ASAE.
- Hofstee, J. W., and D. Goense. 1999. Simulation of a controller area network-based tractor-implement data bus according to ISO 11783. *Journal of Agricultural Engineering Research* 73:383-394
- ISO. 1998. Tractors and machinery for agriculture and forestry – Serial control and communications data network – Part 3: Data link layer. ISO Standard 11783. 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. 01-1196. St. Joseph, MI.: ASAE.
- Ollis, M., and A. Stentz. 1996. First results in vision-based crop line tracking. In *Proceedings of the IEEE Robotics and Automation Conference*, 951-956 Minneapolis, MN.
- Reid, J. F., Q. Zhang, N. Noguchi, and M. Dickson. 2000. Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture* 25:155-167
- Reid, J. F., and S. W. Searcy. 1987. Vision-based guidance of an agricultural tractor. *IEEE Control Systems*. 7(12):39-43
- SAE. 1998. Part7, Vehicle Application Layer. SAE Standard J1939. *SAE Handbook*. Warrendale, PA: SAE.
- Stombaugh, T. S., E. R. Benson, and J. W. Hummel. 1999. [Guidance control of agricultural vehicles at high field speeds](#). *Transactions of the 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, MI.: ASAE.
- Stone, M., K. McKee, C. Formwalt, and R. Benneweis. 1999b. ISO 11783: an electronic communications protocol for agricultural equipment. ASAE Distinguished Lecture #23, Agricultural Equipment Technology Conference. Louisville, Kentucky.
- Tian, L., J. F. Reid, and J. W. Hummel. 1999. Development of a precision sprayer for site-specific weed management. *Transactions of the ASAE* 42(4):893-900.
- Wei, J., N. Zhang, N. Wang, D. Oard, Q. Stoll, D. Lenhert, M. Neilson, M. Mizuno, and G. Sing. 2001. Design of an embedded weed-control system using controller area network (CAN). ASAE Paper No. 013033. St. Joseph, Mich.: ASAE.
- Willrodt, F. L. 1924. Steering attachment for tractors. US Patent No. 1506706.
- Young, S. C. 1993. Electronic control system for GENESIS™ 70 series tractor. SAE Paper No. 941789. Warrendale, PA.: SAE