Development of a Cooperative Tractor-Implement Combination

While driver assistance systems such as adaptive cruise control and lane-keeping assistants are increasingly handling longitudinal and lateral control of vehicles in road traffic [1], in the agricultural industry the trend towards driverless systems is making progress. This trend is also being supported at the "Chair for mobile machinery" of the Karlsruhe Institute for Technology (KIT) in conjunction with electronic drawbars for agricultural machinery developed by the companies AGCO and "geo concept" [2].

Currently, there is great interest in automated driving in the automotive industry. This topic is also gaining in importance in the development of agricultural machines. In a research project at the Karlsruhe Institute for Technology, a new system has now been developed, in which a manned agricultural machine drives ahead as a lead vehicle, and another one follows it autonomously and driverless. The goal of the project is to increase area performance and optimize machine utilization in agricultural business without incurring additional personnel costs.

Electronic Drawbars for Tractors

The electronic drawbar makes it possible to operate two tractors simultaneously with just one driver. This involves using two tractors of the same type, equipped with identical implements and highly precise RTK-GNSS receivers (Real-Time Kinematics) and to interlink them via data radio. While one tractor, which is configured as the master vehicle, drives ahead with a human driver, an unmanned vehicle configured as a slave follows with a prescribed longitudinal and lateral offset. The work settings of the master vehicle to implement the local work process are copied by the slave while still stationary. Good operability and a high level of safety are achieved by the use of environmental sensors to detect obstacles **(Figure 1)** and web-based geo-information.

System Functionality

Five driving modes are available to the driver in operating the electronic drawbar.

The "Parallel Driving" mode serves as the starting mode and working mode. Here the slave follows the driving track of the master and performs the same work task. If the slave follows exactly in the driving track of the master, then the "Tracking" mode is useful. The "Ignore" mode makes it possible to quickly decouple the slave to perform independent driving maneuvers with the master. When leaving this mode, the slave joins up with the master on its current path. The "Turn-Over driving" mode is available for a turning maneuver at the headland. If an obstacle is reported along the planned path, the slave seeks out a collision-free detour path in the "Evasion" mode.

Safety Concept and State Machine

Safety is a key aspect in the development phase of an unmanned vehicle. Based on ISO 25119, typical tools are used to identify risk sources and weaknesses related to the functional safety of a machine, and these risk sources and weaknesses are addressed by suitable measures. A risk analysis yields an Agricultural Performance Level (AgPL) for each scenario studied in the product life cycle, from which requirements can be derived for hardware and software. The electronic drawbar is based on a three-stage safety concept **(Figure 2)**.

> In the "Operational" state, there are no critical error messages or warning messages; the planned path is free of obstacles.
> The "Safety Stop" (SSt) state is understood as an active stop for driving and working drives.
> The highest level of the safety concept is the "Emergency Stop" (ESt) state. Here, the engine is shut off immediately, and the electronic parking brake is engaged.

The safety concept is implemented in the form of a state machine which is active as the central software module in both vehicles which sets the synchronized overall system state. **Figure 3** shows the state machine of the electronic drawbar. At the system start, the "Default" state is active initially.

The role of the tractor can be set as master or slave. After setting roles, an assignment must be made to both vehicles in the "Assignment" state, in which the radio link between the two vehicles is produced. After assignments have been made to the two vehicles, the system transitions to the "Safety Stop". The slave is now in the "Active Stop" state. Now the "Emergency Stop" can be triggered at any time. To initiate driving operation, after enabling by the driver the vehicles are coupled to one another in the "Docking" state, provided that their relative positions and orientations allow this. The system transitions to the "Operation" state, and the slave is able to drive autonomously. It starts in the standard driving mode "Parallel Driving", then any other driving mode can be activated. Risk of collision is detected by environmental sensors consisting of 2D laser scanners and 3D cameras, and detection leads to an immediate "Safety Stop".

Prototype Development at Distributed Sites

Implementation of a complex modular assistance system with external project partners requires a detailed specification of interfaces. Because the development platforms at individual partners are not harmonized, a distributed architecture was selected, in which different software and hardware modules communicate with one another over a proprietary edaugCAN bus and over TCP/IP Ethernet.

Hardware Architecture

While the project partner was responsible for implementation of navigation functions (NAV), access to proprietary and customized Geo data servers (GIS) and access to vehicle control (EXT) via the main controller and user interface for the standard terminal (HMI), KIT implemented the state machine (ZST), and environment detection (UFS) on a rapid prototyping platform.

Figure 1: Prototype setup for testing environmental sensors of the slave tractor

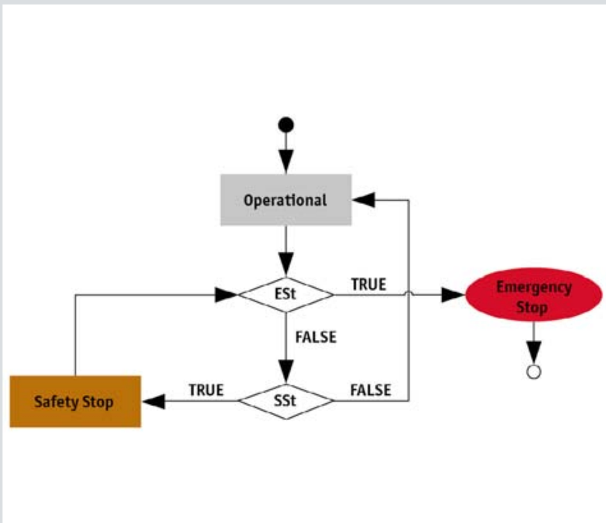

Figure 2: Safety concept of the electronic drawbar

Data Communication

The physical medium used to exchange system and operating data within the vehicles over the proprietary CAN protocol was the tractor's ISOBUS. By choosing 11-bit identifiers, both buses can be operated in parallel without any problems. Each hardware module has its own CAN-ID, under which all relevant signals are transmitted via multiplexing. The bandwidth-intensive environment data is transmitted over TCP/IP-Ethernet. A low number of nodes and network load permit real-time sending and receiving of data. Radio communication between the vehicles is in full-duplex mode. The modems used are linked via RS232 interfaces.

Master and Slave continually exchange data on position and movement states, the system state, settings of working drives, the vehicle state and environmental information **(Figure 4)**. Pragmatic organization of the interfaces enables uncomplicated implementation and validation of subfunctions. Timeouts and checksums were used as safety mechanisms for diagnosis of typical data transmission errors. This guarantees the data integrity and activity of the nodes. If a timeout is exceeded on the CAN bus or in the radio path, this immediately leads to an "Emergency Stop", because it is no longer possible to assure control over the vehicle.

Remaining Bus Simulation to Support Software Development

In the development of ECUs, the remaining bus simulation (RBS) is a powerful tool that is used to support debugging and to validate the software under the most realistic constraints possible. Not only in the automotive industry, but in mobile machinery as well, ECUs, sensors and actuators communicate in different networks simultaneously. Generally, comprehensive software tests are not possible until the ECU's interfaces can be supplied with realistic values. To avoid having to perform time-consuming and complicated software tests in the real system in the development phase, an RBS can be used to virtually simulate the remaining network environment of an ECU. Software tests performed in this way are cost-effective, fast and noncritical in terms of safety.

CANoe

Vector's CANoe software flexibly supports the development of network communication. The structure of a CAN network is fully simulated in CANoe with the help of network nodes, in which the communication behavior of the nodes is stored. In addition, user interfaces, signal generators and other communication interfaces can be created as network nodes. The event-driven or interactive behavior can be simulated in CAPL, a C-like programming language [3] or, for example, with the help of compiled Simulink models. In a database, all messages that occur, including signal structure, are stored and assigned to individual network nodes. CANoe offers multibus capability, i.e. it supports both proprietary and standardized protocols such as CAN, J1939 or ISO11783 as well as FlexRay and Ethernet. For modeling and time-synchronous evaluation, it offers a set of stimulation and tracing functions with many different

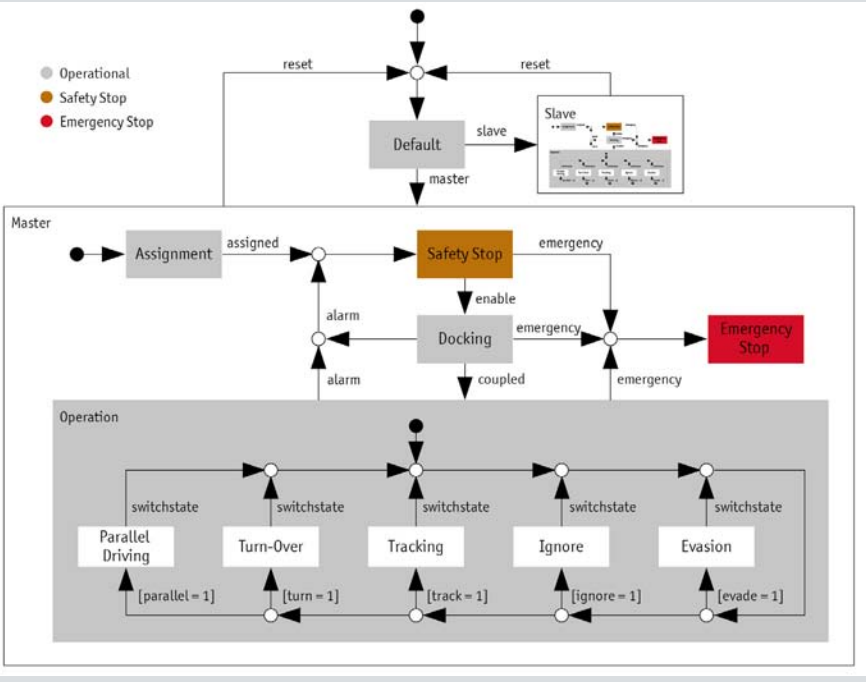

Figure 3: Model for defining the operating state

filter options, graphic signal display and an overview of important communication parameters such as bus load and error messages. Any desired ECUs can be operated virtually or as real ECUs by activating or deactivating the network nodes in CANoe. The bus interface is via an interface from the company Vector. CANoe can also run test scenarios fully automatically and report on the results. These test scenarios may be defined in table form or in XML, CAPL [3], C# or graphically.

Modeling the Electronic Drawbar in CANoe

CANoe is well-suited to this project configuration, especially in early validation of ECUs, because coordinating testing dates and options with the test vehicles is often critical. The approach using RBS enables parallel development paths, which can later be easily merged.
A comprehensive model of the electronic drawbar was developed at KIT that could be used to simulate the individual ECUs as well as the entire communication with the partner vehicle. The ZST, NAV and HMI software modules were modeled as network nodes here. The depth of detail of the individual modules varies within the model. While all messages controlling the system state were simulated in their full functionality, data integrity could be validated sufficiently by assigning constant test parameters in the remaining messages. The use of CANoe in conjunction with the model-based programmable of the remaining messages dSPACE platform proved to be especially helpful. After the exchange of platform-specific I/O blocks, the messages for the developed Simulink models can be immediately compiled for CANoe and integrated in the remaining bus model. Since the serial interfaces of the PC could be accessed in CANoe, the radio path could also be integrated in unmodified form and be tested for data integrity.

Summary

The electronic drawbar has not only proven to be complex with regard to the innovative aspect of object detection and integration of geo-information in a harmonious safety and operating concept. In a continually changing prototype system, the specification and maintenance of data interfaces is a supportive task that must be taken seriously by all parties. When multiple companies are cooperating on a project, a clearly specified modular system and software architecture quickly proves to be an important foundation for quick progress in the project. However, rapid prototyping systems in conjunction with RBS software represent a project-assisting toolbox for efficiently validating subfunctions and entire ECUs so that they are successful in subsequent field testing.

Translation of a German publication in HANSER automotive, September 2014.

Image rights:
Lead Figure: Vector Informatik GmbH
Figure 1 – 4: KIT

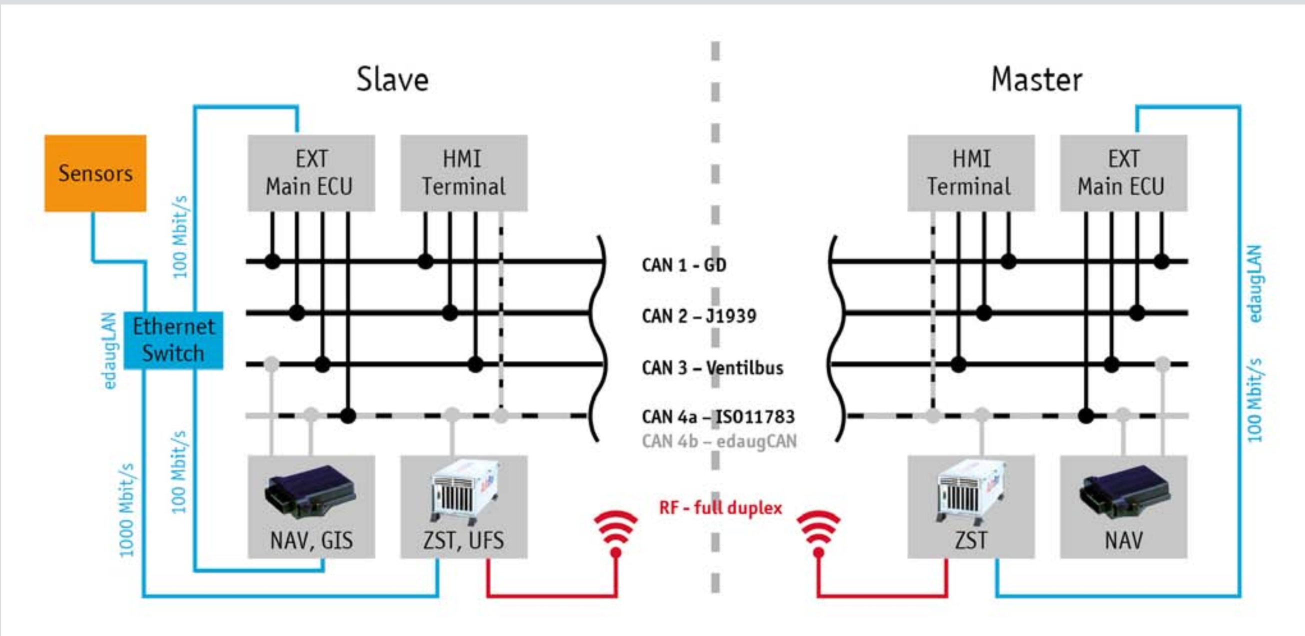

Figure 4: Communications architecture in prototype

Literature:

[1] Winner, H.; Hakuli, St.; Wolf, G. (2012). Handbuch Fahrerassistenzsysteme ("Handbook of Driver Assistance Systems"), 2nd edition, ISBN 978-3-8348-1457-9. Vieweg + Teubner Verlag, Springer Fachmedien Wiesbaden GmbH.
[2] Zhang, X.; Geimer, M.; Noack, P.O.; Grandl, L. (2010). Development of an intelligent master-slave system between agricultural vehicles. pp. 250-255. IEEE. Intelligent Vehicles Symposium. 6/21-6/24/2010. San Diego, CA, USA
[3] Marc Lobmeyer/Roman Marktl: Programming ECU Tests More Efficiently – Tips and Tricks for the Use of CAPL. In: CAN Newsletter (2014), H. 2, pp. 42-43

Links:

Homepage KIT: www.kit.edu/english/
Homepage Vector: www.vector.com
Product Information CANoe: www.vector.com/canoe

Bernhard Jahnke (Graduate Engineer)
is a scientific assistant at the Chair for Mobile Machinery of the Karlsruhe Institute for Technology (KIT).

Hans-Werner Schaal (Graduate Engineer)
is Business Development Manager in the areas of Automotive Ethernet, Car2x and open CAN protocols such as J1939 and ISO11783 at Vector Informatik GmbH.