AALTO UNIVERSITY SCHOOL OF ELECTRICAL ENGINEERING
Department of Automation and Systems Technology

Antti Kunnas

Embedded Control System for Agricultural Machinery implemented in Forage Harvesting

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo, 4 December 2011

Thesis supervisor:

Prof. Arto Visala

Thesis instructor:

M.Sc. Juha Backman

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical Engineering

Abstract of the Master's Thesis

Author: Antti Kunnas

Title: Embedded Control System for Agricultural Machinery implemented in Forage Harvesting

Date: 4 December 2011 Language: English Number of pages: 70+8

Department: Automation and Systems Technology
Professorship: AS-84 Automation Technology
Supervisor: Professor Arto Visala
Instructor: Master of Science Juha Backman

Efficient use of machines is especially important in forage harvesting due to the short harvesting period and expensive machinery. To achieve the best efficiency, a harvesting machine, such as a loader wagon, should be used with optimal loading. Whereas overloading the machine can cause blockages in the cut-and-feed unit, underloading consumes more time and reduces the quality of the resulting silage. In addition, the quality can be improved by optimizing the dosage of the additive. In this thesis, two ISO 11783 compatible electronic control units were implemented for optimizing the harvesting process. The electronic control units were developed using a tool chain designed for developing ISO 11783 systems. The main parts of the tool chain are Matlab Simulink with C code generation, PoolEdit with the associated parsers, and Visual Studio with a Windows CE embedded target. Mass flow of forage in the cut-and-feed unit of the wagon is estimated with a Kalman filter according to tractor speed, swath size, mass of the load, and moisture content of the forage. The mass flow estimation is used to derive the optimal spreading rate of additive so that the ratio of additive and forage is precisely what is desired. The speed of the tractor is controlled with a fuzzy controller according to the capacity of the machine, the mass flow estimation, and the swath size. As a result, the mass flow in the cut-and-feed unit is always optimal regardless of the swath size and thus no blockages will be formed. In the field tests, the performance of the system proved to be in line with the specification.

Keywords: ISO 11783, agricultural machinery, embedded systems

Aalto-yliopisto sähkötekniikan korkeakoulu	Diplomityön tiivistelmä

Tekijä: Antti Kunnas
Työn nimi: Maatalouskoneiden sulautettu ohjausjärjestelmä rehunkorjuussa
Päivämäärä: 4.12.2011 Kieli: Englanti Sivumäärä: 70+8

Laitos: Automaation- ja systeemitekniikka
Professuuri: AS-84 Automaatiotekniikka

Valvoja: Professori Arto Visala
Ohjaaja: Diplomi-insinööri Juha Backman

Koneiden tehokas käyttö on erityisen tärkeää rehunkorjuussa lyhyen korjuuajan ja koneiden kalleuden takia. Parhaan tehokkuuden saavuttamiseksi noukinvaunun kaltaisen korjuukoneen kuormituksen tulisi olla optimaalinen. Ylikuormitustilanteessa noukin voi tukkeutua, kun taas alikuormitus kuluttaa aikaa ja heikentää rehusilpun laatua. Laatua voidaan myös parantaa säilöntäaineen tarkalla annostelulla. Tässä diplomityössä kehitettiin kaksi ISO 11783 ohjainta parantamaan rehunkorjuun tehokkuutta. Ohjaimet kehitettiin käyttäen työkaluketjua, joka on suunniteltu ISO 11783 ohjainten kehitykseen. Työkaluketjun pääosat ovat Matlab Simulink C-koodin generoinnilla, PoolEdit ja siihen liittyvät parserit sekä Visual Studio ja Windows CE tietokonemoduuli. Rehun massavirtaa noukkimella estimoidaan Kalman-suotimella nopeuden, karhon koon, kuorman massan sekä kosteuden perusteella. Massaestimaatin perusteella säilöntäainetta voidaan levittää rehuun siten, että säilöntäaineen ja rehun suhde on optimaalinen. Traktorin nopeutta säädetään sumealla säätimellä koneen kapasiteetiin, estimoidun massavirran sekä karhon koon perusteella. Lopputuloksena massavirta pysyy optimaalisena riippumatta karhon koon muutoksista eikä tukoksia pääse syntymään. Peltotesteissä laitteiston todettiin toimivan määrittelyn mukaisesti.

Avainsanat: ISO 11783, maatalouskoneet, sulautetut järjestelmät

PREFACE

Throughout my thesis work I was provided with valuable help and support from various people. I owe a big thank you to all of them.

First, I would like to thank my supervisor, Professor Arto Visala, for giving me the opportunity to work on this thesis project in the Autonomous Systems research group. The project proved to be very interesting and instructive.

I could not have wished for a better instructor for my thesis than M.Sc. Juha Backman. He patiently advised me on every difficulty I had and provided answers to all my questions. I learned more from him than I have learned from anyone else during my studies. His help really was irreplaceable.

In addition, I would like to thank other people in the Department of Automation and Systems Technology who have somehow helped me during the project. Especially I would like to thank Timo Oksanen for offering me valuable help during the first weeks of the project.

A thank you goes also for the people in MTT Vakola; Antti Suokannas helped me to perform the field tests as well as advised me on the agricultural topics, and Raimo Linkolehto took care of the measurement equipment and data acquisition software. Special thanks to Ari Ronkainen and Riikka Nousiainen for offering me a ride to Vakola whenever I needed one.

Antti Kunnas

Espoo, 4 December 2011

TABLE OF CONTENTS

SYMBOLS AND ABBREVIATIONS

Symbols

x	State of the system
p	Probability density function
u	Control input
z	Measurement
f	State transition model
v	Process noise
Q	Process noise covariance
h	Measurement model
w	Measurement noise
R	Measurement noise covariance
$\hat{x}$	State estimate
ν	Innovation
$\hat{z}$	Measurement prediction
P	State prediction covariance
F	Jacobian of f
S	Measurement prediction covariance
H	Jacobian of h
W	Filter gain
r	Reference value
y	Process output
u	Control output
e	Error value
K_p	Proportional gain
T_i	Integration time
T_d	Derivation time
K_u	Ultimate gain
T_u	Ultimate period
Δ	Delta

Abbreviations

CAN	Controller Area Network
DA	Destination Address
ECU	Electronic Control Unit
EKF	Extended Kalman Filter
FPGA	Field-programmable Gate Array
GE	Group Extension
GPS	Global Positioning System
I/O	Input/output
IECU	Implement Electronic Control Unit
LEM	Light-Emitting Measuring
LMS	Laser Measurement System
NIR	Near-Infrared
P	Priority
PDU	Protocol Data Unit
PF	Protocol Data Unit Format
PGN	Parameter Group Number
PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller
PS	Protocol Data Unit Specific
PTO	Power Take-Off
PWM	Pulse Width Modulation
RPM	Revolutions Per Minute
RT CPU	Real-Time Central Processing Unit
SA	Source Address
SDK	Software Development Kit
TC	Task Controller
TECU	Tractor Electronic Control Unit
VT	Virtual Terminal
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

1 INTRODUCTION

For years electronics have been used to enhance the performance in agricultural machinery, yet without a standardized communication protocol much of the potential has been left unexploited. To utilize the full potential of electronics, the ISO 11783 (International Organization for Standardization) standard of electronic communications protocol for agricultural machinery was introduced. It allows the development of sophisticated control systems, such as automatic steering and cruise control, thus taking the automation in agriculture to a new level.

The Department of Automation and Systems Technology of Aalto University School of Electrical Engineering has been researching and developing the ISO 11783 standard and its applications since 2003; initially through the Agrix and the Farmix projects, and currently in the Agromassi (Assisting and adaptive agricultural machine) project, which this thesis is a part of. Agromassi is conducted in co-operation with MTT Agrifood Research Finland, the Department of Agricultural Sciences of University of Helsinki, and 11 agricultural machinery and software manufacturers including Arctic Machine, Elho, Junkkari, Kemira, Parker-Vansco, Potila, Suonentieto, Valio, Valtra, Vieskan Metalli, and Wapice. The project is a part of Effima program. TEKES (Finnish Funding Agency for Technology and Innovation) funds 70% of the work for the research centres and 30% for the companies. (FIMECC 2011)

The main objective of Agromassi is to develop assisting and adaptive features for tractor-implement systems, which will not only reduce the operator’s workload but also improve the efficiency and precision of the work process. The new features introduced in the project are categorized as mechatronic automation and control, management of the driving process, management of the cultivation process, and support systems for contract work. Developed methodologies and tested concepts in Agromassi will support future product developments in the associated companies. (Visala 2011)

1.1 Objectives of the thesis

The objective of this thesis is to optimize the forage harvesting process, which will not only improve the efficiency of the process, but will also improve the quality of the resulting silage. Since the quality has an important effect on the product of a dairy and beef farm, poor quality can cause significant financial loss for the farmer (Tella 2007).

Optimization can be divided into two different cases. The first case is to optimize forage collecting with a loader wagon. In this process, forage is first mown and raked into swaths. Then, a loader wagon is driven over the swath; the cut-and-feed unit of the wagon picks up the forage and cuts it into chop (Suokannas and Sipilä 2008). Underloading the machine reduces the efficiency of the work as well as the quality of the chop (Suokannas and Nysand 2006), whereas overloading causes blockages in the machine. To maintain a proper level of loading, the operator has to monitor several factors, such as the forage mass flow in the cut-and-feed unit, and control the speed of the machine accordingly. The objective of this case is to implement a controller which will estimate the mass flow and command the speed based on the estimation and the capacity of the machine (Visala 2011).

The second case is to optimize the application of forage additive. A proper ratio between additive and forage will improve the preservability properties of the silage and prevent excess consumption of the additive. The additive is sprayed when the forage enters the cut-and-feed unit, thus the dosing of the additive depends on the forage mass flow. For the operator it is difficult to estimate the mass flow and to maintain a proper additive flow accordingly; thus, the objective of this case is to implement a controller for regulating the additive flow based on the mass flow estimation provided by the controller in the loader wagon.

To accomplish the objectives described above, a tool chain for developing ISO 11783 control systems is utilized. Another objective of this thesis is to test and further develop the tool chain which was first introduced in Agromassi safety research.

This thesis is divided into four chapters. The first chapter explores the ISO 11783 standard alongside previous research regarding the objectives of this thesis. In addition, it investigates alternative tool chains for developing ISO 11783 control systems. Furthermore, it introduces the mathematical methods used in this thesis. The second chapter introduces the tool chain used in this thesis and explains the design of the controllers for the loader wagon and the additive applicator. The results of the tests performed with the controllers are presented in the third chapter, while the results and objectives are reviewed and discussed in the fourth chapter. Subsequently the conclusions in the fifth chapter summarize this thesis.

1.2 ISO 11783

For decades agricultural machinery has been used for various work performed on the farm. The most important of these machines is the tractor, to which different work machines, or implements, such as a seeding machine or a baler, are connected. Implements are usually powered by the tractor power take-off (PTO) and controlled by the hydraulic controls of the tractor. Thus, the tractor and the implement have to be compatible even if they are designed by different manufacturers, which is often the case. Standards for PTOs, hydraulic connections, and three-point hitches ensure the compatibility of the mechanical connections (ISO 2011a). However, over the last couple of decades, the number of electronics in agricultural machinery has been increasing, thus creating a demand for standardized electronic communications. Without a standard, the implements all have their own control boxes installed inside the tractor cabin, their own external sensors for measuring the state of the tractor, and lots of cable for connecting everything.

The demand for the standardized electronic communications protocol for agricultural equipment is met by the ISO 11783 standard which specifies the communication network between the tractor and implements. It allows the distribution of controls and sharing of information between them; thus, the controls can utilize the same sensor data as well as the same user interface device. As a result, the control systems have become simpler, yet more sophisticated.

1.2.1 History

The development of communication network for agricultural equipment first began in Germany in the late 1980s. The result was the DIN 9864 (Deutche Industrie Normen) (DIN 1991) standard, also known as LBS (Landtechnik BUS System), which had its main parts completed by the end of 1991. Simultaneously, the SAE J1939 (Society of Automotive Engineers) standard for heavy duty vehicles was developed in North America (SAE 2011). Both standards are based on Controller Area Network (CAN), yet they are totally incompatible. From the basis of the two standards, the development of the ISO 11783 standard of electronic communications protocol for agricultural equipment was started in 1992. It relies on SAE J1939 derived components for the basic communications structure with applications largely derived from DIN 9684. (Stone, et al. 1999) By 2011 the standard, consisting of 13 parts, is still being further developed; however, several manufactures are already using it (ISO 2011b). The ISO 11783 standard is sometimes called as ISOBUS which, however, is actually a specification based on the ISO 11783 standard (VDMA 2002). The specification is intended for manufactures to help implement the standard. It is inherently similar to the ISO 11783 standard.

1.2.2 Parts of the network

ISO 11783 supports applications both on self-propelled systems and on tractor-implement systems; however, only the latter is discussed here. An example of an ISO 11783 network is presented in Figure 1. It contains various electronic control units (ECU), a tractor bus and an implement bus. An ECU is a device for connecting to the bus and implementing control functions. A tractor bus handles the communications between the ECUs within the tractor; for instance, engine, transmission, and hitch ECUs. The realization of the tractor bus depends on the manufacturer, although it is recommended to comply with the standard (ISO 2005a). An implement bus connects devices such as implement electronic control units (IECU), a task controller (TC), and a virtual terminal (VT). The two buses are connected via a tractor electronic control unit (TECU), which among other functions relies the information between the buses. The wiring of the buses is composed of a single linear quad twisted cable connected to each ECU. Two of the wires are used to carry data, while the other two are used to provide power for the terminators at the end of the bus. The wiring solution, with the bus topology shown in Figure 1, allows a data transfer rate of 250 Kbit/s. (ISO 2002b)

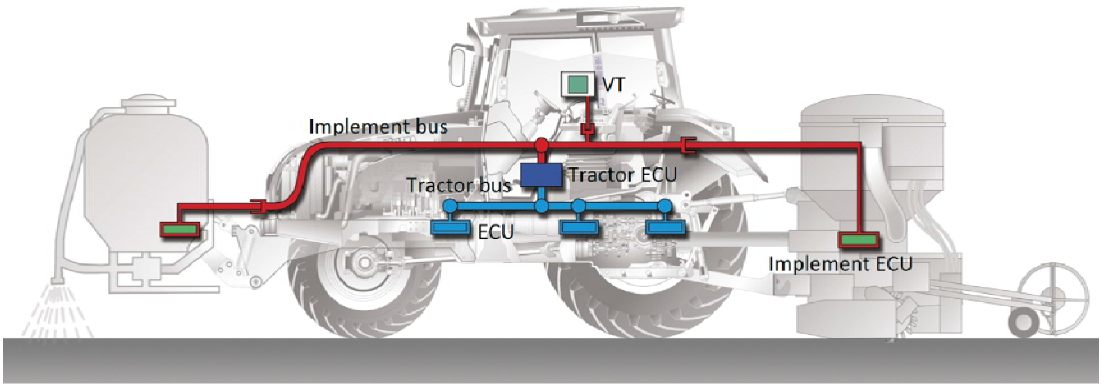

Figure 1: An ISO 11783 network consists of an implement bus, a tractor bus, and various ECUs. (Valtra 2011a)

1.2.3 Communication protocol

The basis of the ISO 11783 data link layer is the CAN 2.0B 29 bit protocol (Bosch 1991). ISO 11783 defines the interpretation of the 29 bits in the identifier of CAN frames as well as the interpretation of the data. Two types of identifier structures, or protocol data units (PDU), are defined as shown in Figure 2. In both type of PDUs, the least significant 8 bits define a source address which is the physical address of the ECU sending the message. The physical address is claimed when the ECU first connects to the bus. The first three most significant bits of the identifiers are defined as independent priority bits. The difference between the two PDU types is in the content of the PDU Specific (PS) field. The content is determined by the PDU Format (PF) field value; PF values between 0-239 imply it is a destination address, while values between 240-254 imply it is a group extension. A destination address allows the message to be sent to a particular ECU based on the physical address. The PDU fields, excluding Priority, Destination Address (DA), and Source Address (SA), are used to compute a Parameter Group Number (PGN), a unique numeric identifier for each group of parameters that may be contained in the data field. Some of the PGNs are left undefined in the standard and can be defined for proprietary usage by the manufacturer. The length of the data in one message frame is restricted to 64 bits, however for longer messages, ISO 11783 specifies a multi-packet transport protocol. (ISO 1998)

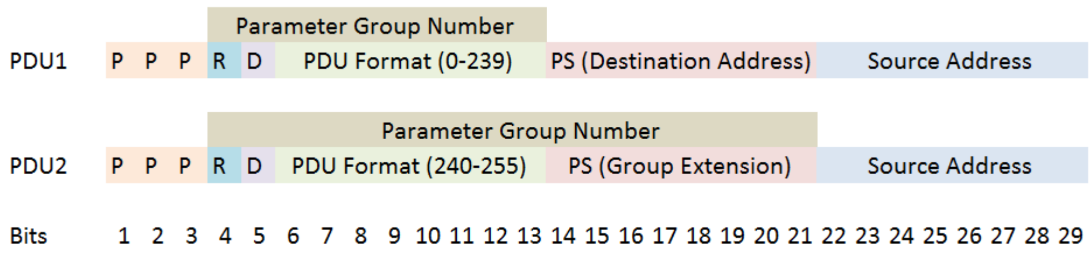

Figure 2: ISO 11783 defines two identifier frames, or protocol data units (PDU), where P stands for Priority, R for Reserved, D for Data Page, and PS for PDU specific bits.

1.2.4 Implement electronic control unit

One or several implement electronic control units are needed to control a single implement. For performing various functions, an IECU often has an input/output (I/O) for operating external devices, such as actuators or sensors installed in the implement. Connected to the implement bus, the controller can communicate with the other ECUs as well as with the virtual terminal; together the controllers in an implement form a working set with a shared user interface on the VT.

1.2.5 Tractor electronic control unit

A tractor electronic control unit provides a gateway between the tractor bus and the implement bus. It is responsible for transmitting information on the state of the tractor to the implement bus and relaying commands to the tractor. ISO 11783 specifies three tractor classes depending on the available features on the TECU. Class 1 TECUs provide the implement bus only with the basic tractor internal measurements, for instance speed, PTO, and hitch information. In addition to these, class 2 TECUs provide more advanced measurements, such as measurements on the horizontal force of rear hitch and hydraulic valve flow. Furthermore, class 3 TECUs are prepared to take commands from the implement bus. IECUs are allowed to command the tractor's rear-hitch, PTO, auxiliary valve control, as well as speed and steering. It is advised to provide new tractors with at least class 2 TECU. (ISO 2002a)

1.2.6 Virtual terminal

A virtual terminal is a special kind of ECU as it provides the network with a user interface. It has a graphical display, soft keys, and a means to retrieve user inputs. The display area is divided into two areas; the data mask area and the soft key area. The data mask covers most of the display area and is used to display various objects, such as buttons, number and string fields, geometrical shapes, meters and bar graphs, and bitmap graphics. The soft key area is used to display the soft key labels. In addition, the VT supports auxiliary inputs, such as joysticks or button panels. Two virtual terminals are shown in Figure 3. (ISO 2004)

Figure 3: A VT provides ISO 11783 systems with a graphical display, soft keys, and means to retrieve user inputs. (PRLog 2010) (Valtra 2011b)

The user interface description consists of a set of objects, such as the ones mentioned in the previous paragraph. The set of objects, or the object pool, is loaded to the VT by a working set master, which represents a working set formed by an ECU, an implement, or a group of implements. After loading the user interface, the working set can use the VT to display data by changing the content of an object, or to retrieve user inputs by reading the contents of an object. The objects are identified by unique object IDs. Several working sets can have their user interfaces loaded in the VT at the same time, however, only one can be active at a time. The active interface can be changed from a button. (ISO 2004)

The properties of the VT can vary with different models. However, a user interface should be readable on any VT regardless of the model. The working set master can query the terminal on the hardware details, such as the display resolution, the color depth, or the number of soft keys, and then scale the graphical objects in the object pool accordingly before loading the pool to the terminal. (ISO 2004)

1.2.7 Additional equipment

A task controller is another special kind of ECU connected to the implement bus; it provides task management for the machines connected to the ISO 11783 network. Tasks can be planned beforehand and loaded to the task controller which executes them on the field by commanding other ECUs. A task may be for instance to seed different parts of a field with different kinds of seeds. To execute position dependent tasks, a Global Positioning System (GPS) receiver is connected to the implement bus. The GPS message format is specified in the NMEA 2000 (National Marine Electronics Association) standard (NMEA 2000). In addition, the task controller is used to collect data during field operations for producing farm records. Other additional equipment on the bus is a sequence control and a file server. (ISO 2005b)

1.3 Previous research

Despite the fairly short age of the ISO 11783 standard, its applications have been researched in numerous studies. Various tools for developing ISO 11783 control systems have been experimented, however, only few development environments provide complete support for the standard. In addition, with regards to the objective of this thesis, a few methods for automating the forage harvesting process have been introduced; some of them utilize the ISO 11783 standard, while some rely on other technologies. The developing tools alongside the automating methods are investigated in this section.

1.3.1 Development environment

A basic requirement for the development environment of ISO 11783 systems is to provide a means for designing a user interface and program logic; and furthermore, connect the user interface events with the logic. Another requirement is the usability; few system engineers are familiar with all the bit level issues in the ISO 11783 standard, hence the development environment has to abstract the details of the standard from the designer. (Oksanen, Kunnas and Visala 2011)

1.3.1.1 ISOAgLib

OSB AG offers a tool chain for developing ISO 11783 systems. The tool chain is based on the open-source programming library ISOAgLib, which is written in C/C++. The library provides the functions defined by the standard as well as the established machine interfaces. The layered architecture of ISOAgLib allows functions to be abstracted from the hardware, thus hardware changes do not require major adjustments to the software. The library has been used in numerous ISO 11783 projects. (OSB 2011a)

In the tool chain, the ISO 11783 graphical user interface is designed with vt-designer, which is a commercial product of OSB AG. Vt-designer allows the graphical development, editing, and emulation of virtual terminal object pools. It stores the pool description in extensible markup language (XML) format that can be transformed into a binary file and loaded into a VT. (OSB 2011b)

1.3.1.2 PoolEdit

PoolEdit is an open-source, XML based, graphical editor for designing ISO 11783 graphical user interfaces. The editor interface is shown in Figure 4 where an example of an object pool is being edited; the appearance of the data masks is shown on the left, the tree view of the object pool as well as the attribute table of an object are shown on the right. Objects can be dragged to the tree view after which their position can be changed from the data mask view and attributes edited from the attribute table. PoolEdit was developed in Aalto University School of Electrical Engineering as a part of the Farmix project. (Öhman, Kalmari and Visala 2008)

PoolEdit stores the VT object pools in PoolEdit XML format which is based on ISOAgLib XML format with some differences. After finishing the design of the user interface, the object pool can be converted from PoolEdit XML into embedded XML format. The difference between the two formats is that the latter includes base64 encoded bitmap data which removes the need for separate image files. In addition, unique object IDs are automatically generated in the conversion. An object pool in embedded XML format can be transformed to ISO 11783 binary format and sent to a virtual terminal. (Öhman, Kalmari and Visala 2008)

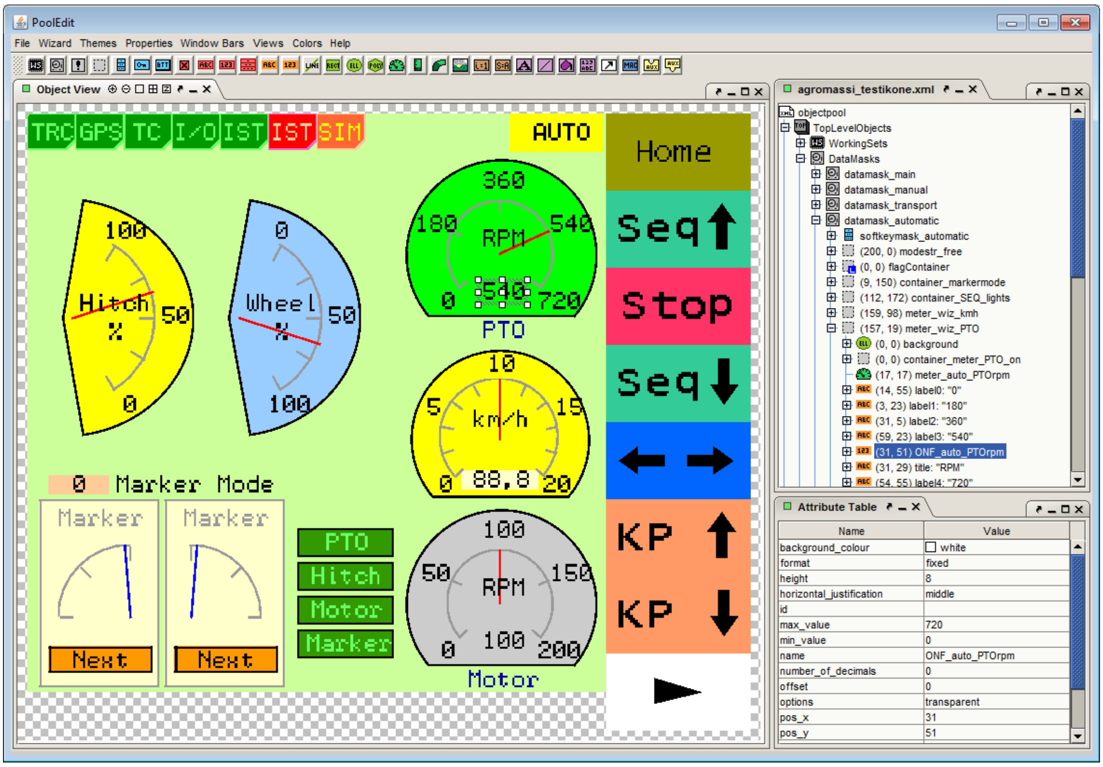

Figure 4: The editor interface of PoolEdit.

1.3.1.3 Constellation

In the Farmix project (Öhman and Visala 2006) (Öhman, Kalmari and Visala 2008), RTI’s Constellation was used with PoolEdit display editor to comprise a tool chain for developing ISO 118783 ECUs. Constellation is a unified modeling language (UML) based tool especially designed for building control systems. It provides a framework that allows utilizing reusable software components, such as for realizing ISO 11783 communications or handling I/O. Constellation is based on C++ programming language and runs on Windows and Linux operating systems. The ECU hardware consists of a PC with a Linux operating system, a CAN interface card, and a wireless local area network (WLAN) adapter which can be used for debugging the application in real-time. (Oksanen, Öhman, et al. 2005) The conclusion of the aforementioned research was that graphical tools, such as RTI’s Constellation, could be used to improve software quality and programmers productivity. However, RTI does not provide support for Constellation anymore, thus the development of Constellation based control systems has been ceased.

1.3.1.4 LabVIEW

National Instruments' LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is an example of a program that integrates user interface design, functionality design, and debugging into a single piece of software. It provides a graphical programming environment for developing measurement, test, and control systems using icons and wires that resemble a flowchart. LabVIEW programs can be run in standard PCs, real-time central processing units (RT CPU), and field-programmable gate arrays (FPGA). (National Instruments 2011)

MTT Agrifood Research Finland has used LabVIEW to implement a PC based ISO 11783 task controller, data acquisition, and even a controller for a seeding machine. However, the LabVIEW user interface designer cannot be utilized for designing ISO 11783 user interfaces, thus it is not possible to create an ISO 11783 ECU with LabVIEW alone.

1.3.1.5 CoDeSys

CoDeSys (Controller Development System) is a software tool for industrial automation technology. It consists of two parts: the programming system CoDeSys and the runtime system CoDeSys Control. The runtime system turns any intelligent automation device into an IEC 61131-3 controller programmable with CoDeSys. (3S-Smart Software Solutions 2011)

CoDeSys and PoolEdit display editor were used to develop an ISO 11783 compatible ECU for a course work in Aalto University School of Electrical Engineering. The hardware chosen for the ECU was Epec 2024 Universal Module which is a CANopen compatible programmable logic controller (PLC). It is designed for vehicle use in rough conditions and the manufacturer provides an ISO 11783 library for CoDeSys. (Epec 2011) During the development of the ECU, not only was the library found difficult to use, but also the IEC 61131-3 programming languages were considered inconvenient for programming ISO 11783 functions.

1.3.2 Forage harvesting automation

1.3.2.1 Forage collecting

John Deere and Pöttinger have developed an automation system for tractor-loader wagon combination used in forage collecting. The system controls speed according to various input parameters, such as swath size, tractor speed, PTO torque, and charging level of the loader wagon. The swath size is measured with nine ultrasonic sensors mounted in the front of the tractor. With the linkage of the swath size and the tractor speed, the volume flow of forage can be predicted. Furthermore, the mass flow can be derived by taking the PTO torque into account. The smaller the swath and torque are, the higher the requested speed, and vice versa. In addition, the implementation measures the charging level of the loader wagon. With higher charging levels, the deceleration and acceleration levels are decreased. (Hoyningen-Huene and Baldinger 2009)

John Deere and Pöttinger concluded that their automation increases the driving comfort as well as increases the productivity by 10%. Their field evaluation showed that the automation is able to prevent blockages in the loading process. The automation implementation has been made into a commercial product that went to market in 2010. (Hoyningen-Huene and Baldinger 2009)

1.3.2.2 Application of additive

To ease the regulation of additive application, a few automatic controls have been introduced. Junkkari provides their HP1000 additive applicator with a speed compensated work mode in which the additive flow is controlled according to the speed of the tractor. However, the mode does not take the swath size into account. To use the speed compensated mode, an additional speed sensor must be installed in the tractor. In addition, an optional switch will turn off the flow when the cut-and-feed unit of the wagon is not in the pick-up position. (Junkkari 2009)

CLAAS has designed a QUANTIMETER which uses light-emitting measuring (LEM) technology to measure the total volume of the collected forage. Together with moisture measurement it can be used to provide exact data on the current dry matter content and forage yield. The additive can then applied according to the actual throughput. (CLAAS 2011) (CLAAS 2009)

1.4 Mathematical methods

This section presents the basics of the mathematical methods utilized in this thesis. First of the methods, the Kalman filter, provides a method for estimating and filtering systems where the measurements are uncertain or the variable of interest is not directly measurable, e.g. in sensor fusion. The basic filter can be applied in linear systems, however, the extended Kalman filter (EKF) supports also nonlinear systems. The second method, fuzzy logic, is often applied in multiple variable systems where no exact equation defining the system exists. A fuzzy controller can be assimilated with the human way of thinking; values and controls are rather fuzzy than exact. Third of the methods, the Proportional-integral-derivative (PID) controller, is the most widely used feedback controller.

1.4.1 Extended Kalman filter

Kalman filters are based on linear dynamic systems discretized in the time domain. Assuming a state vector that completely describes the system, the state of the system depends only on the state at the previous instant of time, not on the entire past. Thus, the system can be modeled by a Markov process described by equation

$$p(x_{k+1}|x_k, x_{k-1}, x_{k-2}, \dots) = p(x_{k+1}|x_k), \quad (1)$$

where the random variable x_k denotes the state of the system at an instant k, while p is the probability density function.

A more general description of a Markov process is shown in Figure 5 where a control input u is affecting the next state and measurement z provides information on the present state. In the form of an equation, it is expressed as

$$p(x_{k+1}|x_k, x_{k-1}, \dots u_k, u_{k-1}, \dots, z_{k+1}, z_k, \dots) = p(x_{k+1}|x_k, u_k, z_{k+1}). \quad (2)$$

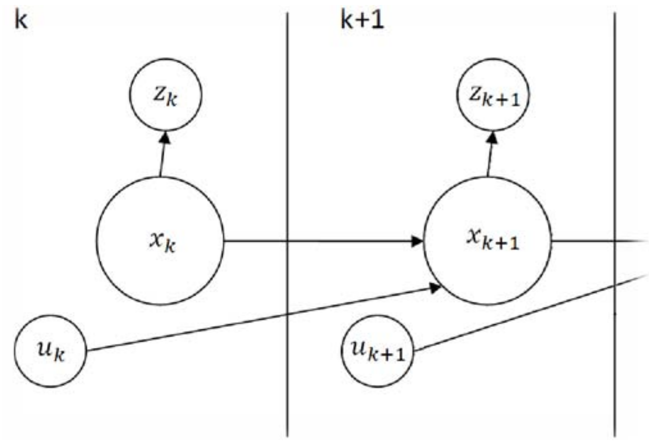

Figure 5: A state x_{k+1} in a Markow process is affected only by the state x_k and control u_k at the previous instant of time, not by the entire past. Measurement z_k provides measurement data of the present state.

The extended Kalman filter model assumes that the state of the system at time k evolves to the state at $k+1$ according to equation

$$x_{k+1} = f(x_k, u_k) + v_k, \tag{3}$$

where f is the state transition model applied to the previous state and to the control input, and v_k is the process noise drawn from a zero-mean normal distribution with covariance Q_k.

A measurement of the state at an instant k is described by equation

$$z_k = h(x_k) + w_k, \tag{4}$$

where h is the measurement model relating the measurement with the true state, and w_k is the measurement noise drawn from a zero-mean normal distribution with covariance R_k.

Assuming the initial state, process noise, and measurement noise are mutually independent, an extended Kalman filter can be applied to estimate the state of the system. The updated state estimate, or posteriori estimate, at time k based on measurements up to and including at time k, is denoted by $\hat{x}_{k|k}$. The state at an instant *k+1* can be predicted according to equation

$$\hat{x}_{k+1|k} = f\left(\hat{x}_{k|k}, u_k\right) \tag{5}$$

which is called a priori estimate, as it only includes measurement information of the state up to and including the previous instant of time.

Based on the estimated state, the prediction of the next measurement can be calculated by equation

$$\hat{z}_{k+1|k} = h\left(\hat{x}_{k+1|k}\right). \tag{6}$$

When the new measurement is obtained, the difference between it and its predicted value, the innovation ν, is evaluated with the equation

$$\nu_{k+1} = z_{k+1} - \hat{z}_{k+1|k}. \tag{7}$$

The state prediction covariance P can be calculated from the current state covariance according to equation

$$P_{k+1|k} = F_k P_{k|k} F_k' + Q_k, \tag{8}$$

where F_k is the Jacobian of *f* in $\hat{x}_{k|k}$.

Using the state prediction covariance, the measurement prediction covariance S is obtained with equation

$$S_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}' + R_{k+1}, \tag{9}$$

where H_{k+1} is the Jacobian of *h* in $\hat{x}_{k+1|k}$.

The filter gain W is calculated from the state and measurement prediction covariance with equation

$$W_{k+1} = P_{k+1|k} H'_{k+1|k} S^{-1}_{k+1|k}. \tag{10}$$

The updated state at the next instant of time is obtained as the sum of the predicted state and the correction term, product of the filter gain W and the innovation term ν, according to equation

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + W_{k+1}\nu_{k+1}. \tag{11}$$

The updated covariance associated with the next state is then computed according to equation

$$P_{k+1|k+1} = P_{k+1|k} - W_{k+1} S_{k+1} W'_{k+1}. \tag{12}$$

With the equations above, the probability density function estimate of the Markov process state is obtained as

$$p(x_k) \sim N(\hat{x}_k, P_k) \ . \tag{13}$$

Thus, the estimate is normally distributed with mean $\hat{x}$ and covariance P.
(Bar-Shalom, Li and Kirubarajan 2001)

1.4.2 Fuzzy logic

Fuzzy logic is based on the concept of a fuzzy set. Compared to a classical set, which either completely includes or excludes any given element, a fuzzy set can partially include an element. For instance a speed value that is not a member of a classical set of fast speed values, might have a partial membership in a corresponding fuzzy set, meaning it is fast to some degree. In other words, the truth of any statement in fuzzy logic is a matter of degree between completely false and completely true, whereas the truth value in classical logic is either false or true. An example with the described sets is shown in Figure 6, where the fastness of a speed value is considered from both points of view. In classical logic the speed of 2.7 m/s is fast to a degree of 0. However, in fuzzy logic it is fast to a degree of 0.5 and therefore regarded as quite fast. The curve that defines to what degree the inputs belong to the fuzzy sets is called a membership function. In the case of fuzzy logic, the resulting degree of membership is called a fuzzy value and the whole process is called fuzzification. (The MathWorks, Inc. 2011a) (The MathWorks, Inc. 2011b)

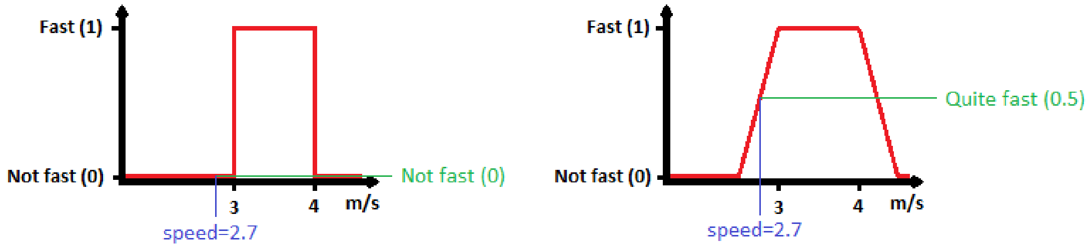

Figure 6: The membership function (red) defines to what degree the inputs belong to the fuzzy sets. A speed value of 2.7 m/s is considered fast to a degree of 0 in the traditional logic, however, in the fuzzy logic, it is fast to a degree of 0.5.

Logical operations with fuzzy values are similar to those with Boolean values. AND operations retrieve the minimum of the two values, OR operations return the maximum and NOT operations are the complement of the value. For example AND operation with fuzzy values of 0.4 and 0.7 will result in 0.4. The operators in fuzzy logic are called fuzzy operators. (The MathWorks, Inc. 2011a) (The MathWorks, Inc. 2011b)

The fuzzy logic comprises of linguistic if-then rules used with fuzzy sets and fuzzy operators. For example, input fuzzy sets “low speed value”, “low mass flow”, and output set “increase speed” can be used with AND operator to form a rule “if speed is low and mass flow is low then increase speed”. The result of the AND operator defines to what degree the speed is increased, in other words, a minimum operator is performed on the result of the fuzzy operator and the output fuzzy set. Several rules can be combined by taking the maximum of the output sets to form one output set. The formed output set can be defuzzified by taking the centroid of the set. A process of combining three rules is illustrated in the Figure 7. (The MathWorks, Inc. 2011a) (The MathWorks, Inc. 2011b)

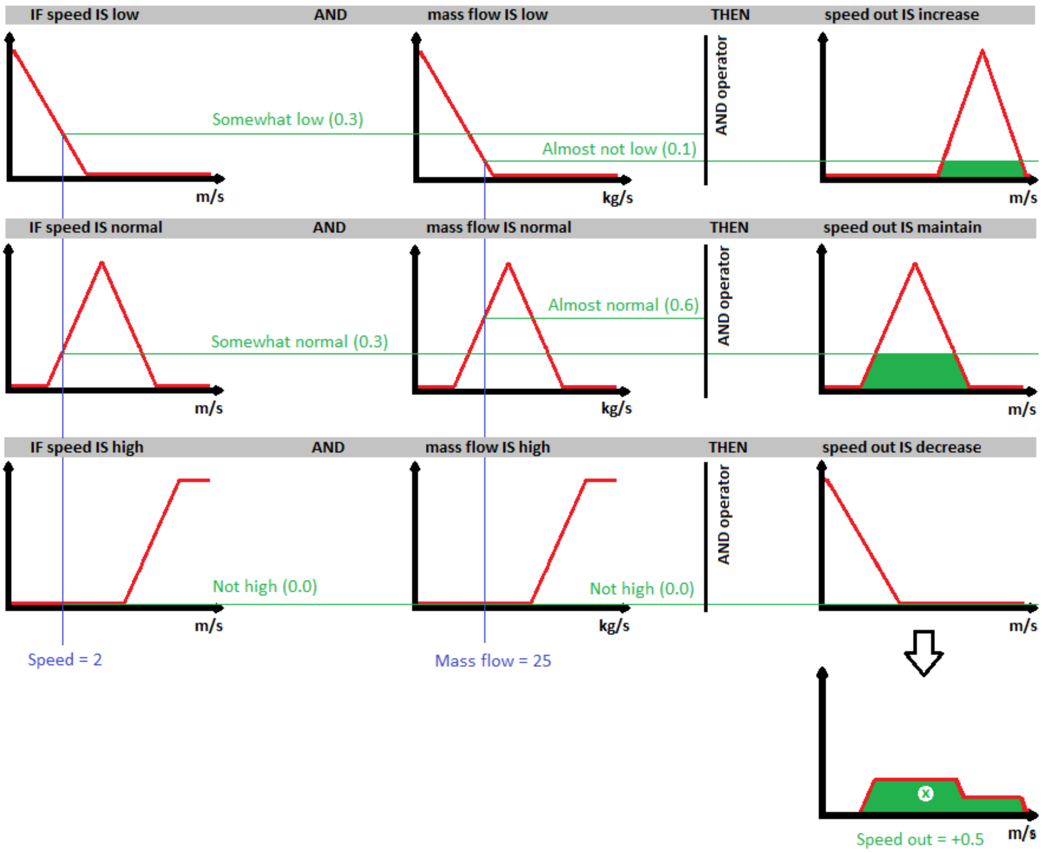

Figure 7: Three fuzzy rules are combined to return a single value output. First, a fuzzy operator acts upon the fuzzified input sets. The result defines to what degree the output set is applied. Then, the output sets are combined into a one single set from whose centroid the single value output can be obtained.

1.4.3 PID controller

A PID controller can be interpreted by the block diagram, as shown in Figure 8. In the diagram, e is the error value denoting the difference between reference value r and process output y, u is the control output, K_p is the proportional gain, T_i is the integration time, and T_d is the derivative time. (Åström 2002)

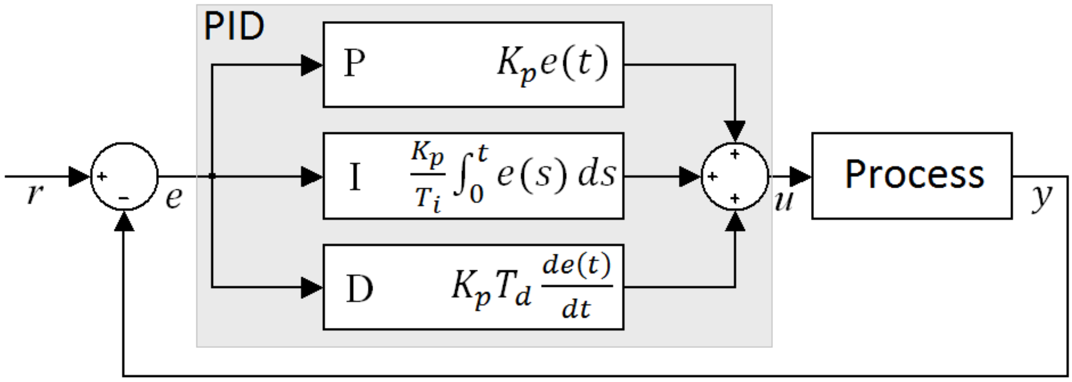

Figure 8: A PID controller consists of a proportional, an integrative, and a derivative term. The controller is inputted with the error value *e*, which is the difference between the process output *y* and reference value *r*. The process is controlled with the control output *u*.

The proportional term (P) is proportional to the error at the instant *t*, which is the “present” error. The P term is the basis of the controller. The integral term (I) is proportional to the integral of the error up to the instant *t*, which can be interpreted as the accumulation of the “past” error. Thus, it can be used to remove the accumulated error in the system. The derivative term (D) is proportional to the derivative of the error at the instant *t*, which can be interpreted as the prediction of the “future” error. Thus, it can be used to prevent future errors such as overshoot and boost when the reference value changes. (Åström 2002)

Output saturation combined with integral term can cause problems with a basic PID controller. The saturation occurs when an actuator reaches its limit, such as maximum speed, which breaks the feedback loop. Then the system runs as an open loop because the actuator will remain at its limit, independent of the process output. If a controller with integrating action is used, the error will continue to be integrated, which may cause the integral term to become very large, to “wind up”. It is then required that the error has opposite sign for a long period before the system returns to normal. A common method to deal with the wind up is to restrict the integral term from increasing when the output saturates. (Åström 2002)

1.4.3.1 Tuning of PID controller

The Ziegler-Nichols frequency response method is a well known and simple method for tuning PID controller. However, since the parameters are found based on oscillation, it can only be applied on processes that can handle the oscillation of the controller. The first step of the method is to make the controller purely proportional by setting $T_i = \infty$ and $T_d = 0$, after which the gain is increased until the process starts to oscillate. The gain when this occurs is called the ultimate gain K_u and, respectively, the period of oscillation is called the ultimate period T_u. Parameters for the PID controller are given by Table 1. (Åström 2002)

Table 1: PID parameters for the Ziegler-Nichols frequency response method.

Control Type	K_p	T_i	T_d
P	$0.5K_u$		
PI	$0.4K_u$	$0.8T_u$	
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$

Another method by Ziegler and Nichols is based on a process information in the form of the open-loop step response. The step response is characterized by only two parameters, *a* and *L*, which can be found by drawing a tangent at the point where the slope of the response has its maximum. The parameters are then given by the intersections between the tangent and the coordinate axes as shown in Figure 9. The controller parameters can be found in Table 2. (Åström 2002)

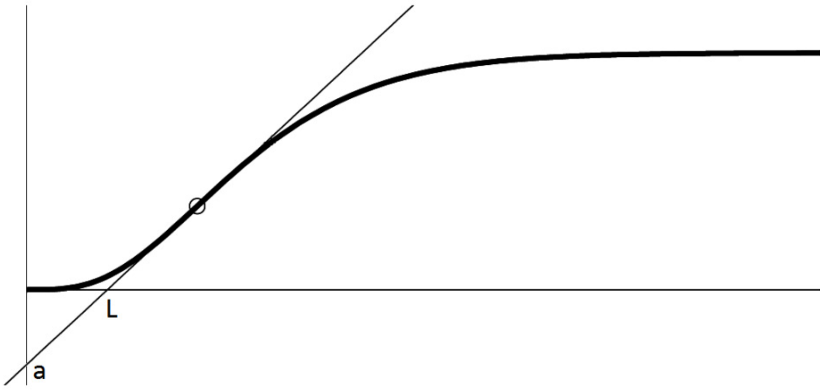

Figure 9: A step response is characterized by parameters determined from the intersection of the axes and the tangent drawn at the point where the slope has its maximum; parameter *L* can be found from the intersection at x-axis, and *a* at y-axis.

Table 2: PID parameters for the Ziegler-Nichols step response method.

Control Type	K_p	T_i	T_d
P	1/*a*		
PI	0.9/*a*	3*L*	
PID	1.2/*a*	2*L*	*L*/2

2 METHODS

This chapter describes the methods for reaching the objectives of this thesis. First, the utilized tool chain for developing ISO 11783 control systems is presented. Next, the developed control systems for both the loader wagon and the additive applicator are discussed in detail.

2.1 Tool chain

In order to simplify the development process of an ISO 11783 compatible ECU, various tool chains have been introduced, some of which were presented in section 1.3.1. The tool chain used in this thesis utilizes software integration; its main parts are Matlab Simulink with C code generation, PoolEdit with associated parsers, and Visual Studio with a Windows CE embedded target. (Oksanen, Kunnas and Visala 2011)

2.1.1 Simulink

MathWorks's Simulink is a tool for modeling, simulating and analyzing dynamic systems. It provides a graphical user interface for building models as block diagrams. The usage of Simulink can be expanded with tool boxes such as the Real-time Workshop toolbox, allowing the generation of the model into C code to be used in an embedded target, and the Stateflow toolbox used for developing state machines and flow charts. (The MathWorks, Inc. 2011c)

2.1.2 Simulink framework

The tool chain includes a Simulink framework providing a basis for the logic of the ISO 11783 ECUs. A common requirement for all ECUs is to communicate with the other devices on the bus. Thus, to allow the reuse of the program logic for the common parts, the framework includes ready-to-use function blocks from the basic block for reading a single ISO 11783 message to the block for handling the whole address claiming process. For instance, Figure 10 illustrates a function block for retrieving information on the tractor state; the messages containing the data are read inside an S-function with C code according to a specific PGN, after which the data is handled in the state chart. As all the required ISO 11783 communications is programmed inside the blocks, they can be added to the ECU's Simulink model without considering the details of the standard. A basic structure of an ECU's model is shown in Figure 11. It receives and sends ISO 11783 messages, manages address and VT connection, and handles the work mode selection. Control actions can be added to the model depending on the type of the ECU. Furthermore, the framework supports runtime debugging with the external mode of the Simulink; the behavior of the model can be examined when it is running in the target.

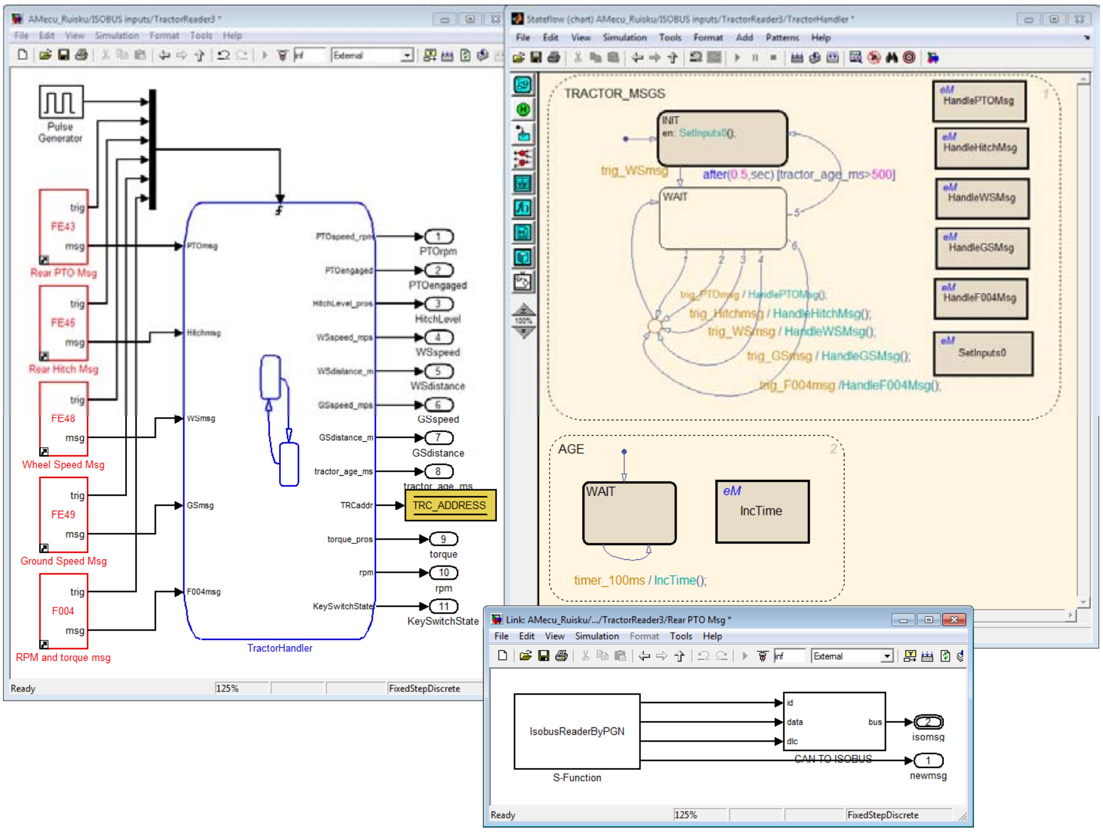

Figure 10: Realization of a function block for retrieving information on the state of the tractor. While messages with specific PGNs are read in the S-functions with C code, the data of the messages is handled in the state chart.

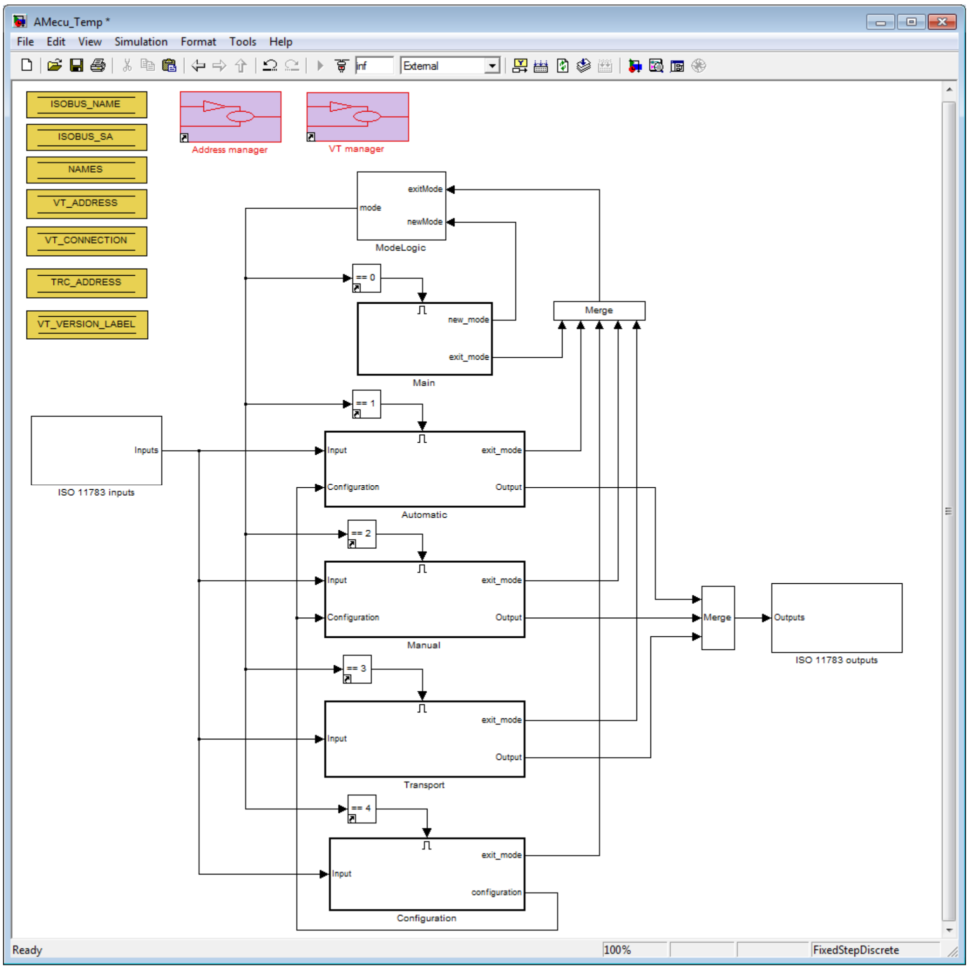

Figure 11: A basic structure of an ECU's Simulink model. It receives and sends ISO 11783 messages, manages address and VT connection, and handles work mode selection.

2.1.3 PoolEditParser and PoolEditObjectParser

PoolEditParser and PoolEditObjectParser are used for processing the object pools created with PoolEdit which was presented in section 1.3.1.2. PoolEditParser is a tool with which a PoolEdit embedded XML file can be converted into a C-file header. For the tool chain used in this thesis, PoolEditParser has been modified to produce a Matlab m-file script defining the object pool as a byte array. In addition, the same parser generates enumerations for the object IDs to ease the usage of the objects in the later phases. (Oksanen, Kunnas and Visala 2011)

PoolEditObjectParser was designed as a part of this thesis for generating a Simulink library from the embedded XML object pool. The parser was developed in Visual Studio with C#. An example of a generated library is shown in Figure 12. The generated library contains function blocks corresponding to an object in the pool; the parser reads the object's type, ID, and some additional attributes from the XML file and creates a function block from a template. The templates for different objects, such as soft keys or output number fields, contain all the required communication procedures with the VT. For instance, an output number field function block will update the corresponding object in the VT screen with the value the block is inputted with. Thus, the user can insert the blocks into the Simulink model without considering the details of the ISO 11783. The realization of an output number field function block is illustrated in Figure 13. (Oksanen, Kunnas and Visala 2011)

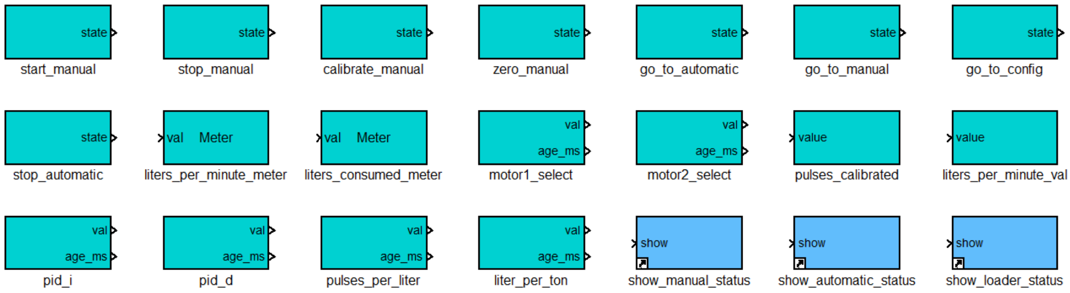

Figure 12: A Simulink library generated from a PoolEdit embedded XML file. Each block corresponds to a user interface object visible on the VT screen.

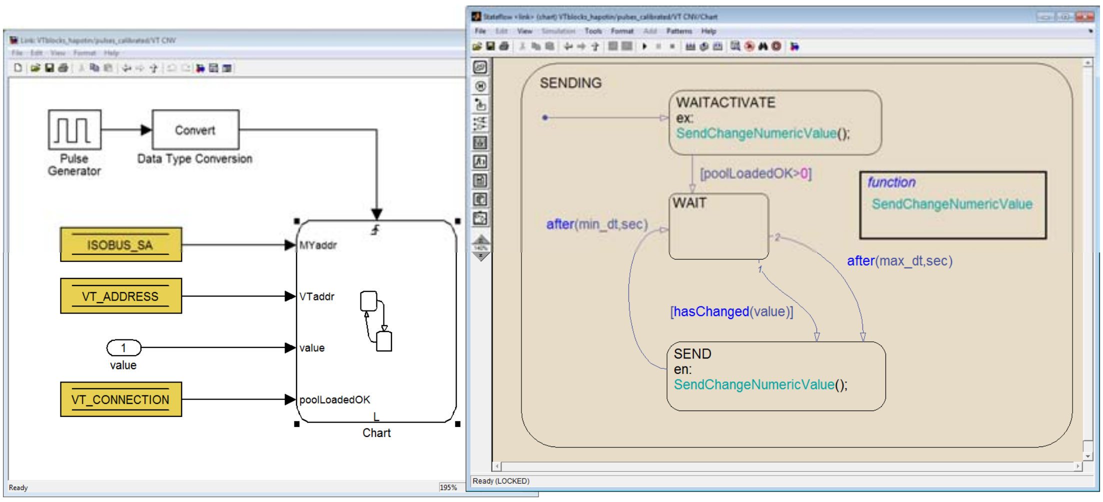

Figure 13: Realization of a function block for updating the output number field. The block updates the value shown on the VT screen every time the value changes.

2.1.4 C code generation and runtime

The Simulink model is converted into C code with the Real-Time Workshop toolbox. The code is integrated into short runtime code in Visual Studio handling CAN hardware abstraction, direct I/O, and real-timing of the control system. The default sample time for the system is 10 ms, however, it can be changed depending on the machine requirements.

The compiled program is deployed to an embedded target based on a Toradex Colibri with a Marvell ARM XScale PXA320 processor. It runs at 806 MHz and it comes with a pre-installed Windows Embedded CE 6.0 real-time operating system (Toradex AG 2010a). The Colibri computer module is used with a Toradex Protea carrier board. The combination, shown in Figure 14, includes a variety of features, such as integrated CAN transceiver, Ethernet, 1GB integrated flash memory, RS232 and I2C communications (Toradex AG 2010b). The chosen hardware provides an embedded solution with easy programmability, a powerful processor, and a possibility for remote debugging via Ethernet. The manufacturer supplies drivers for CAN and I2C communication as well as a software development kit (SDK) for Visual Studio.

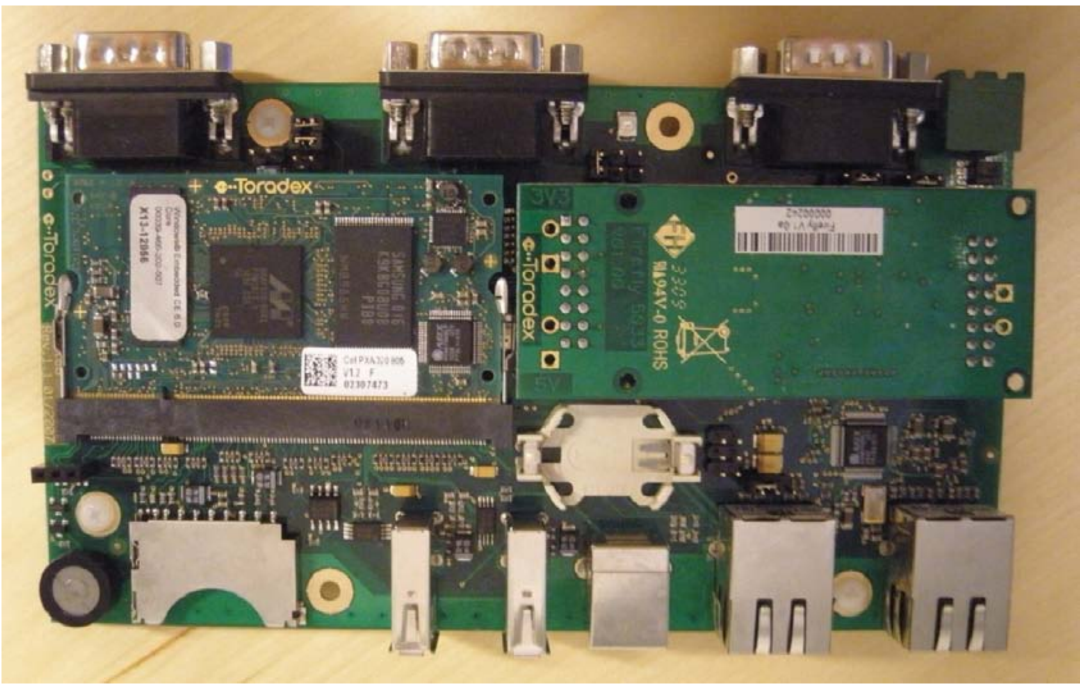

Figure 14: A Toradex Colibri computer module installed in a Toradex Protea carrier board.

2.2 Optimizing forage harvesting process

Methods for optimizing the forage harvesting process are presented in these two sections. The first section introduces the control system for optimizing the forage collecting process, and the second section presents the system for optimizing the application of forage additive.

2.2.1 Forage collecting

The loader wagon is one of the tractor implements used for collecting pre-dried forage from the field. In the collecting process, the forage is first mown and raked into swaths. Then, the loader wagon is driven over the swath; the cut-and-feed unit of the wagon picks up the forage, cuts it into chop, and stuffs it into the cargo space (Suokannas and Sipilä 2008). Providing the unit with a sufficient forage mass flow is important as it affects the quality of the resulting chop. Not enough forage on the cut-and-feed unit will allow a greater number of straws to pass the knives without being cut. The length of the chop is an important quality factor contributing to the preservation properties of the resulting silage (Suokannas and Nysand 2006). In addition, shorter chop will take less space, thus allowing heavier loads (Suokannas 2006). In contrast to insufficient mass flow, too much forage on the cut-and-feed unit will cause blockages. Clearing the blockage will take several minutes and it can be an unpleasant task, especially if additive has been sprayed on the forage.

2.2.1.1 Objective

Collecting forage with a loader wagon can be a wearing task. The operator has to constantly monitor the mass flow, size of the swath, and performance of the machine in order to control the speed of the tractor optimally. The objective was to ease the operator’s work by implementing an intelligent controller for estimating the mass flow and controlling the speed based on the estimation and the capacity of the machine. As a result, the mass flow is maintained at optimal level and blockages are prevented.

2.2.1.2 Equipment and measurements

The tractor used in this thesis is a Valtra T132 evaluation version equipped with a research implementation of an ISO 11783 class 3 TECU. The loader wagon is Krone ZX 45 GL.

To provide the loader wagon ECU with all the necessary data for mass flow estimation and speed control, some additional sensors were installed on the tractor and the loader wagon. The installation and calibration of the sensors were performed by researchers of MTT Agrifood Research Finland. Thus only a short explanation of their work is presented here. The sensors and other devices are shown in Figure 15.

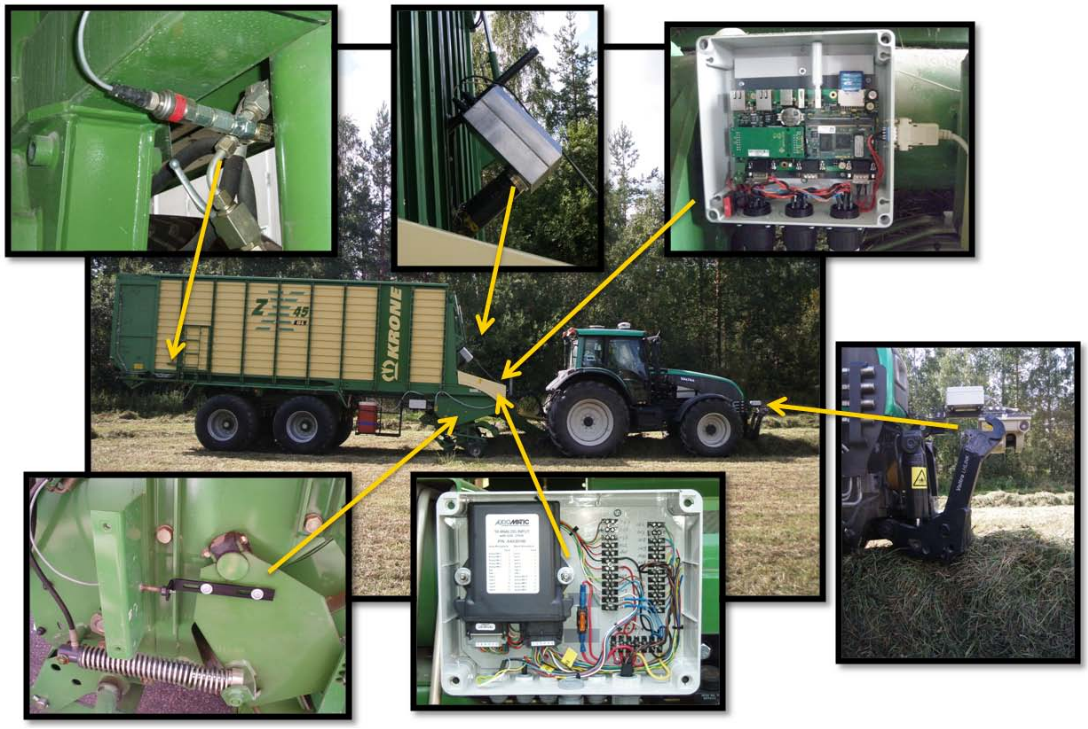

Figure 15: The tractor–loader wagon combination with additional devices. Devices on the top row are a pressure sensor, a near-infrared photometric analyzer, and a loader wagon ECU. Two other pressure sensors are not shown in the figure. Devices on the bottom row are a sensor for monitoring the position of the cut-and-feed unit, an I/O module for relaying the readings of the analog sensors to the ISO 11783 bus, and a laser scanner for measuring the swath size.

The hydraulic suspension of the wagon was utilized in measuring the load mass; one pressure sensor was set up to measure the pressure on the suspension of the drive shaft, and two others to measure the pressure on the rear axles, one on each side. With calibration coefficients, the total mass can be calculated from the voltage readings of the sensors. In addition, two other sensors were installed on the loader wagon. One is to signal whether the cut-and-feed unit is in the pick-up position or not, and the other is measuring the speed of the scraper floor. The sensors are all connected to an Axiomatic AX030100 I/O module which reads the signals, constructs an ISO 11783 proprietary message containing the data, and relays it to the bus.

For measuring the moisture percentage of the forage, an MCT series near-infrared (NIR) photometric analyzer was installed on the front wall of the wagon and connected directly to the loader wagon ECU via RS232 bus (Process Sensors Corporation 2007). The density of the swath can be derived from the moisture percentage; it is affected by the amount of water stored in the forage, the amount of dry matter, and the tightness of the forage in the swath.

A laser scanner was mounted in the front of the tractor for measuring the cross-sectional area of the swath. Sick Laser Measurement System (LMS) 221 provides distance measurement throughout the 180° scanning field (SICK 2003), thus it is used to measure the distance from the scanner to the ground. The measurement width was adjusted to 4 meters. The distance measurements are transformed into Cartesian coordinates to obtain the shape of the ground plane, from which the cross-sectional area of the swath is determined. The value of the area is then relayed to the bus using a proprietary message. This is illustrated in Figure 16.

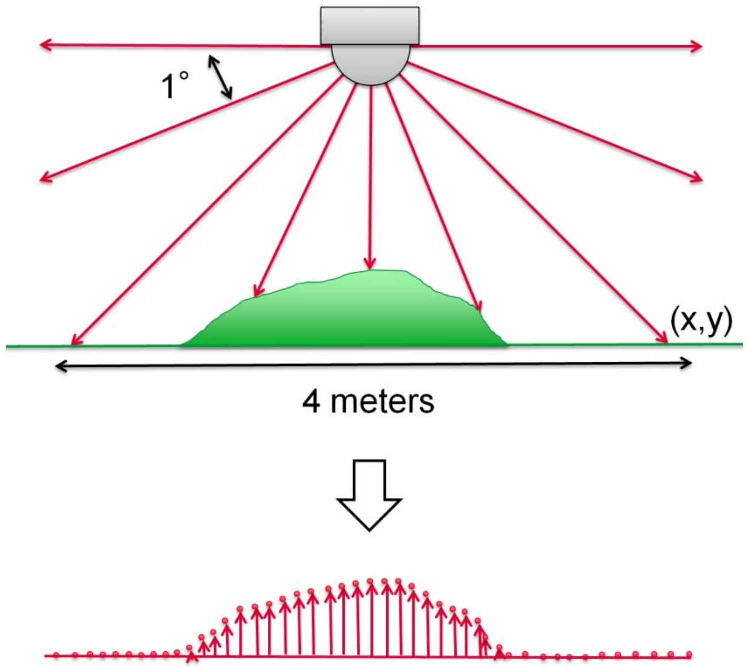

Figure 16: The laser scanner measures the distance to the ground. The distance measurements are transformed into Cartesian coordinates to obtain the shape of the ground plane, from which the cross-sectional area of the swath can be determined. The measurement width is adjusted to 4 meters.

2.2.1.3 Loader wagon ECU

The loader wagon ECU was implemented using the tool chain and hardware described in section 2.1. A block diagram of the program running in the ECU is presented in Figure 17. Its main parts are a Kalman filter implementing the mass flow estimation and a fuzzy controller implementing the speed control. The Kalman filter requires three inputs, first of which is the volume flow of forage in the cut-and-feed unit. The flow is a product of the cross-sectional area of the swath and the speed of the tractor. Since the laser scanner is measuring the swath in the front of the tractor and the point of interest is the swath in the front of the cut-and-feed unit, the cross-sectional area measurement has to be delayed. The delay time can be calculated from the speed of the tractor and the distance between the laser scanner and the cut-and-feed unit. The other two inputs are the total mass of the collected forage and the density of the forage. The mass can be calculated from the measurements of the three pressure sensors with the calibration coefficients, and the density can be derived from the moisture measured with the NIR photometric analyzer.

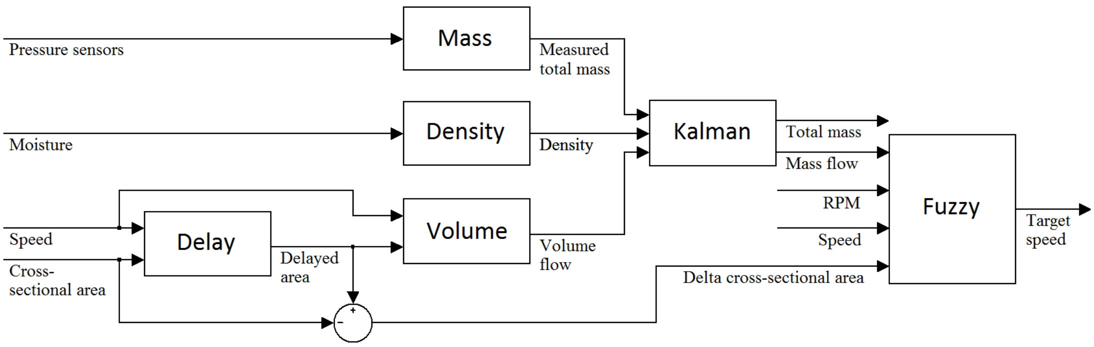

Figure 17: The main parts of the control logic are a Kalman filter and a fuzzy controller.

The fuzzy controller takes engine revolutions per minute (RPM) and tractor wheel speed as inputs from the ISO 11783 bus. Other data it utilizes comprises of the mass flow, which is available from the Kalman filter, and the change in the cross-sectional area, which is the difference between delayed and non-delayed measurement of cross-sectional area.

2.2.1.4 Mass flow estimation

The mass flow estimation was implemented with an extended Kalman filter. The filter is based on a model with three states: total mass x_1, mass flow x_2, and density x_3. Thus, the mass flow estimation is available directly as a state. During every increment of time, the total mass increases according to the mass flow, whereas the mass flow and density are independent of other states. The state transition model is interpreted as

$$f(x) = \begin{bmatrix} x_1 + x_2 \\ x_2 \\ x_3 \end{bmatrix}, \qquad (14)$$

where x_2 has been scaled with the increment of time.

The filter has three measurement inputs: total mass z_1, volume flow z_2, and density z_3. The volume flow measurement is related to the division of mass flow state and density state, and total mass measurement and density measurement are related to the corresponding states. The measurement model is

$$h(x) = \begin{bmatrix} x_1 \\ x_2/x_3 \\ x_3 \end{bmatrix}. \tag{15}$$

Jacobians of the models are respectively

$$F_k = \frac{\partial f}{\partial x}\bigg|_{x_k} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and} \tag{17}$$

$$H_k = \frac{\partial h}{\partial x}\bigg|_{x_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/x_{3,k} & -x_{2,k}/x_{3,k}^2 \\ 0 & 0 & 1 \end{bmatrix}. \tag{17}$$

2.2.1.5 Speed control

The speed control for the loader wagon was designed with Matlab Fuzzy Logic Toolbox which includes editor for creating fuzzy sets with associated membership functions and rules. The main purpose of the speed control is to prevent blockages and maintain a sufficient forage mass flow. In order to achieve these purposes, there are four measurements that need to be taken into consideration: mass flow, changes in the swath size, tractor speed, and engine RPM. The mass flow should be maintained within reasonable boundaries by controlling the speed, i.e. the speed should be increased when the flow is low and vice versa. In addition, the controller should react to changes in the swath size; decreasing the speed when there is an increase in the swath size can prevent the mass flow from rising. Furthermore, the higher the speed, the stronger the decrease of the speed should be. Finally, the engine RPM is affected by the loading of the cut-and-feed unit in the loader wagon. A decrease in the RPM is a sign of overloading and might result in a blockage. Thus, if the RPM decreases, the machine should slow down in order to reduce the mass flow and lighten the load in the cut-and-feed unit.

The rules for the fuzzy controller were derived by the ideas introduced above. Several rules with various fuzzy sets were required to achieve a smooth control of the speed. The rules and fuzzy sets can be found in APPENDIX A and APPENDIX B.

The input signals of the fuzzy controller are scaled to a constant level of -1 to 1. By adjusting the scaling coefficients, the mapping of inputs to the membership functions can be adjusted during runtime. Without the coefficients, runtime adjustments would not be possible because Matlab does not allow changes inside the fuzzy controller. Scaling coefficients are as follows:

- *Target mass flow* is a value to which the controller tries to settle the flow.
- *Mass flow standard deviation* determines how much the flow can differ from the target value.
- *Normal RPM* is a value that RPM should stay above.
- *RPM standard deviation* determines how much RPM can go below the normal value.
- *Median speed* is a value that is in the middle of the speed range.
- *Speed standard deviation* determines how much speed can differ from the median.
- Δ *swath size range* determines the allowed difference between swath sizes in front of the tractor and in front of the cut-and-feed unit.
- *Output scaling* determines the speed output range of the controller.

2.2.1.6 User interface

The user interface of the loader wagon ECU consists of main, automatic and configuration tabs. The main tab is for selecting the other tabs as well as for shutting down the ECU for test purposes. The main tab is shown in Figure 18.

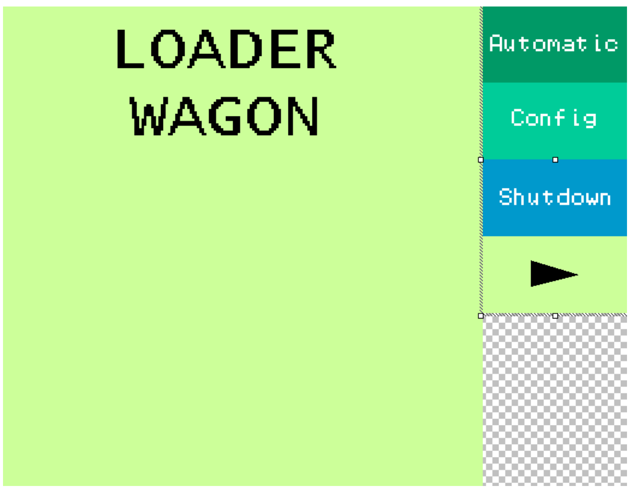

Figure 18: From the main tab of the loader wagon ECU, it is possible to access the other tabs as well as shutdown the ECU.

From the automatic tab, the user can start or stop the automatic mode. After the automatic mode is turned on, the ECU takes control over the tractor speed. The automatic mode can be stopped by either pressing the “Stop” soft key, going back to the main tab, or by engaging the external stop switch connected to the bus. In addition, there are soft keys for calibrating the total mass measurement. Pressing the “Calibrate total mass” soft key sets the current total mass value as the zero level. Calibration can be undone by pressing the “Reset Calibration” soft key. The Kalman filter can be reset from the “Reset Kalman” soft key which, however, is only for test purposes. The automatic tab is shown in Figure 19.

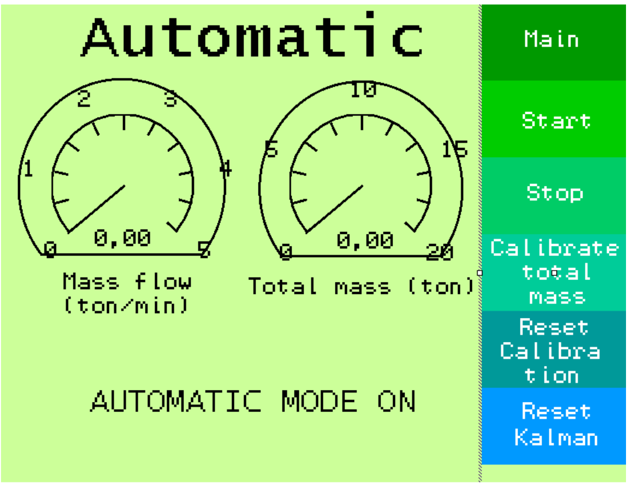

Figure 19: The automatic tab of the loader wagon ECU contains a few actions; activation of the automatic mode, calibration of the total mass measurement, and resetting the Kalman filter.

The configuration tab shown in Figure 20 contains the scaling coefficients for the fuzzy controller as discussed in the previous section.

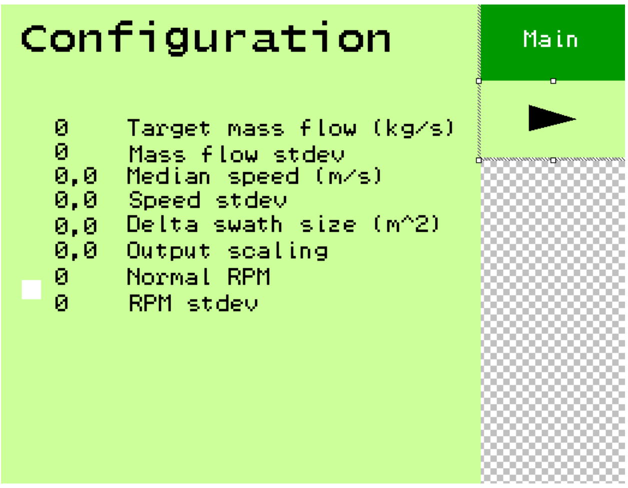

Figure 20: The configuration tab of the loader wagon ECU contains the scaling coefficients for the fuzzy controller.

2.2.2 Application of forage additive

An additive applicator is an implement for applying additive on forage when it is collected by a forage harvester, such as a loader wagon or a baler. A proper ratio of additive and forage will improve the preservability properties of the silage and prevent excess consumption of the additive. The additive is sprayed when the forage enters the cut-and-feed unit of the harvester, thus the flow of additive depends on the forage mass flow.

2.2.2.1 Objective

For the operator it is difficult to estimate the mass flow and to maintain a proper additive flow accordingly. Usually, the operator will set a constant flow of additive and then control the speed of the machine to maintain a steady flow of forage to match the additive flow with a proper ratio. After loading the machine, the operator can compare the mass of the load with the consumed additive to check the ratio and, on the next load, make changes to the flow or speed accordingly. However, the method can be far from optimal since it is based on rough estimations.

The objective was to modify an additive applicator to meet the ISO 11783 standard and to implement a controller for regulating the flow of additive so that the operator only needs to select the ratio of additive and forage. The controller is able to realize the ratio based on the forage mass flow estimation provided by the loader wagon ECU.

2.2.2.2 Equipment

In this thesis, a Junkkari HP1000 additive applicator was mounted on the side of a loader wagon as shown in Figure 21. The nozzles were installed on the cut-and-feed unit of the wagon in order to spray additive on the forage when it enters the unit. The original setup of HP1000 consists of two motors controlling two diaphragm pumps, a flow meter for measuring the output flow, and a control box for controlling the applicator. The pumps are able to maintain a flow between 2-20 liters per minute and the accuracy of the flow meter is 0.1 liters per minute. (Junkkari 2009)

Figure 21: The additive applicator with an additive container mounted on the side of the loader wagon. The nozzles are installed on the cut-and-feed unit.

2.2.2.3 Additive applicator ECU

To implement the previously discussed functions with HP1000, some modifications were executed. The original control box was removed and replaced by an ECU which was implemented using the tool chain and hardware described in section 2.1. An additional I/O board was designed for interfacing the applicator with the Toradex board. Furthermore, the original motor controller, which is controlling both pump motors, was replaced by one that can be controlled by the ECU. A block diagram of the system is shown in Figure 22.

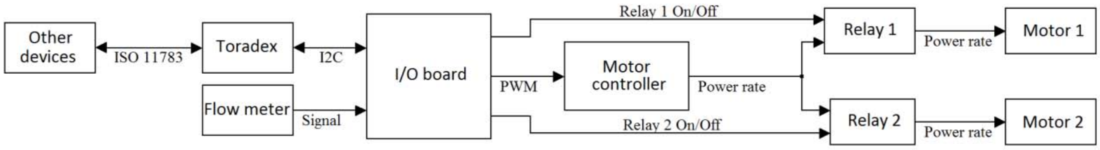

Figure 22: Wiring diagram of the additive applicator

The core of the I/O board is an ATmega168 microcontroller which communicates with the Toradex board via I2C bus. The microcontroller acts as an I2C slave while the Toradex is a master, thus the flow of communication is determined by the Toradex while the controller only responses when requested (Atmel Corporation 2011). The program for the controller was written in C language and was developed in AVR Studio 4.

The I/O board is connected to the motor controller which is controlled with pulse width modulation (PWM) method. The controlling PWM pulses are generated by the microcontroller according to the commands from the Toradex. The pulse width ranges from 0 to 100% determining the power rate respectively. In addition to power rate control, the pump motors can be switched on or off with relays which are controlled by the Toradex via the microcontroller.

The flow meter of the additive applicator uses a paddlewheel for measuring the flow. When the wheel is turning with the flow, the meter generates pulses which are received by the microcontroller. The controller calculates the pulse frequency and sends it to the Toradex.

The circuit board was designed with EAGLE Layout Editor. The board uses optocouplers for separating the ECU from the electronics of the additive applicator in order to prevent the voltage peaks, which motor controlling might cause, from reaching the ECU. The completed board with components in place is shown in the Figure 23. A schematic of the board can be found in APPENDIX C.

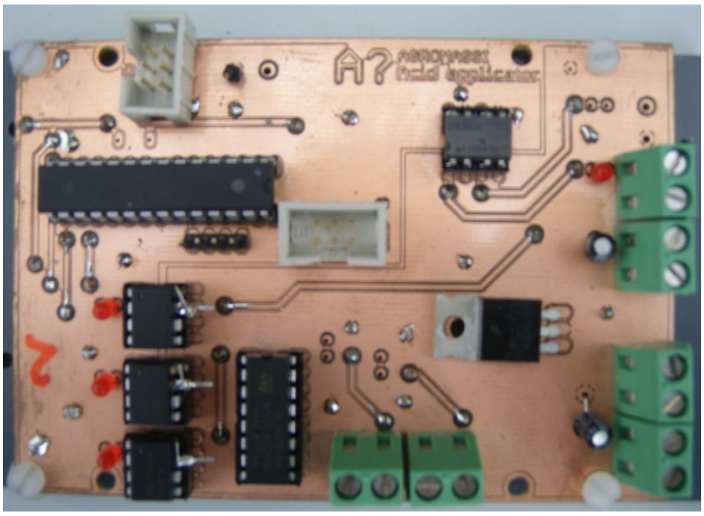

Figure 23: The I/O board with components in place.

2.2.2.4 Control logic

The additive flow can be calculated from the pulse frequency with a calibration coefficient. The frequency is provided by the microcontroller and the coefficient can be found inputted in the VT. Alternatively, the coefficient can be calculated automatically; the operator can input the actual consumption to the VT and the program will calculate a new coefficient by comparing the inputted consumption with the consumption calculated with the previous coefficient.

The additive applicator can operate in manual or automatic mode. In the manual mode the operator can select the motors that are used and set the power rate for them. Motor selection and power rate are sent to the microcontroller which will set the motors on accordingly and drive them with the provided rate.

In the automatic mode, the target additive flow is determined by the forage mass flow and by the number of liters set to be applied on one ton of forage. The value of mass flow is available on ISO 11783 bus and the liters-per-ton value can be found from the VT. Both the target flow and the measured flow are inputted to a PID controller which will set the power rate for motors so that the measured value meets the target value. The PID controller was explained in section 1.4.3; here, the target flow is the reference signal r and the measured flow is the process output y. Furthermore, the automatic mode turns off the flow when the cut-and-feed unit is not in pick-up position and vice versa. A block diagram of the automatic mode is presented in Figure 24.

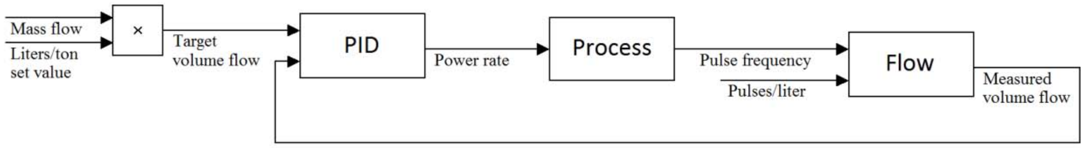

Figure 24: The target flow is inputted to a PID controller which controls the process in order to match the measured flow with the target.

One pump is able to maintain a flow up to 12 liters per minute; hence both pumps are used simultaneously only when the target flow exceeds this limiting value. To prevent the other pump from switching on and off at the limiting flow, it switches on at 12 liters per minute and turns off when it reaches 11 liters per minute. Since both motors are driven with the same motor controller, it is not possible to adjust the power rates separately. As a result, the transition from using one pump to two pumps cannot be performed smoothly by increasing the power rate of the other pump starting from zero, but by taking it in use with the same rate as the other one, which causes a sudden increase in the output flow. However, the PID will quickly adjust the rate to a proper level. Transition from using two pumps to one pump will have a similar effect respectively.

The minimum power rate of the motor is 8% (Junkkari 2009), which corresponds to a flow of 2 liters per minute. However, even lower flow rates are often required. When the output of the PID is lower than the minimum power rate, the motors are set off. This will accumulate the error in the PID, which eventually will cause the output to rise above the minimum and thus one motor will be set on. Once the additive is flowing, the accumulated error will quickly decrease and the output will drop below the minimum, after which the error starts to accumulate again. This will result in switch-like behavior of the flow.

2.2.2.5 User interface

The user interface of additive applicator consists of main, automatic, manual and configuration tabs. The main tab is for selecting the other tabs as well as for shutting down the ECU for test purposes. The main tab is shown in Figure 25.

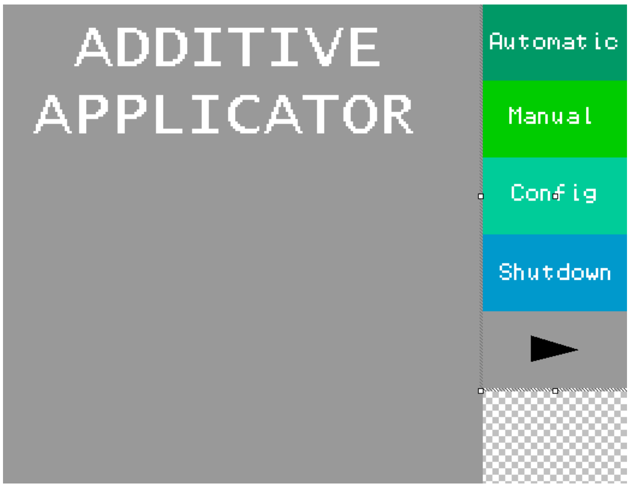

Figure 25: From the main tab of the additive applicator ECU, it is possible to access the other tabs as well as shutdown the ECU.

From the automatic tab, the user can start or stop the automatic mode. After the automatic mode is turned on, the ECU will control the additive flow according to the mass flow and the "Liters per ton" value. The automatic mode can be stopped by either pressing the "Stop" soft key, going returning main tab, or by engaging the external stop switch connected to the bus. The automatic tab is shown in Figure 26.

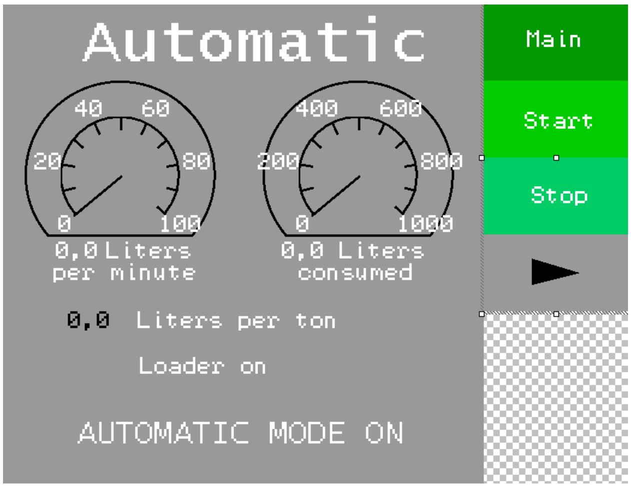

Figure 26: The automatic mode can be activated from the automatic tab of the additive applicator ECU; the controller applies additive according to the liters-per-ton value.

The manual tab, shown in Figure 27, is for controlling the additive applicator manually. The manual mode can be enabled by pressing the "Start" soft key after which the applicator operates with the selected motors and power rate. The manual mode can be stopped by either pressing the "Stop" soft key, returning to the main tab, or by engaging the external stop switch connected to the bus. The flow meter can be calibrated by inputting the actual consumption, measured e.g. with a scale, corresponding to the total consumption the meter is showing and by pressing the “Calibrate counter” soft key. The total consumption can be reset from the “Reset counter” soft key.

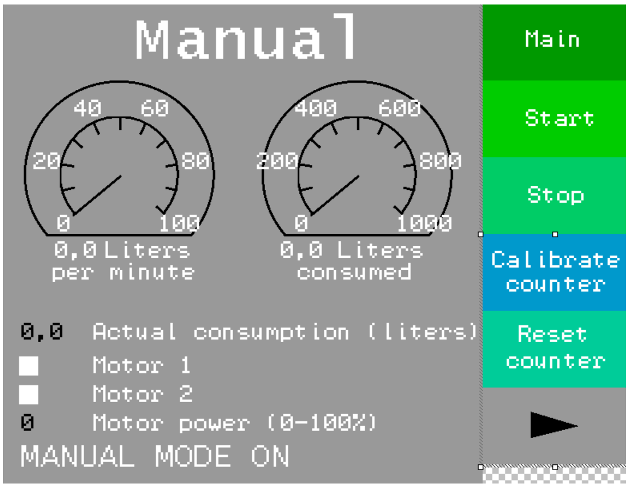

Figure 27: The manual tab of the additive applicator ECU contains a few actions; motors can be turned on separately, the power rate can be chosen manually, and the flow meter can be calibrated.

From the configuration tab, the parameters for PID controller and flow meter can be set. The calibration performed from the Manual tab can be reset from the “Reset calibration” soft key. The tab is shown in Figure 28.

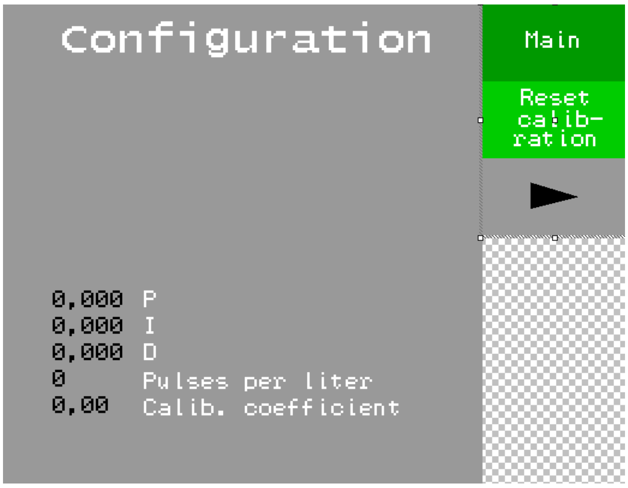

Figure 28: Parameters for PID controller and calibration coefficients for the flow meter can be changed from the configuration tab.

3 RESULTS

Testing of the system was performed on the test track and fields of MTT Agrifood Research Finland. In this chapter, the results are first presented separately for mass estimation, speed control, and additive application. Then, the performance of the whole system in automated harvesting is presented for two loads. Finally, data from a manual drive is provided for the comparison with the automated drives.

3.1 Mass flow estimation

The parameters for the Kalman filter were obtained by trial and error; various parameter values were tested with various sets of data. The most suitable process noise variance for the states was 5 kg for the total mass, 1 kg/s for the mass flow, and 0.2 $\mathrm{kg/m^3}$ for the density. Thus, the covariance for the process noise was set to

$$Q = \begin{bmatrix} 5^2 & 0 & 0 \\ 0 & 1^2 & 0 \\ 0 & 0 & 0.2^2 \end{bmatrix}. \tag{18}$$

The chosen measurement noise variance was 153 kg for the total mass, 1 $\mathrm{m^3/s}$ for the volume flow, and 2000 $\mathrm{kg/m^3}$ for the density. The variance for the density was set high since the measurement was known to be incorrect. Thus, the covariance for the measurement noise was set to

$$R = \begin{bmatrix} 153^2 & 0 & 0 \\ 0 & 1^2 & 0 \\ 0 & 0 & 20000^2 \end{bmatrix}. \tag{19}$$

The mass estimation was tested in the field with the obtained covariances. The behavior of the Kalman filter inputs and states are shown in Figure 29 and Figure 30. In the first figure, total mass measurement, total mass estimation and accumulated mass are shown. As the accumulated mass is based on the estimated mass flow, it shows how well the estimation matches the total mass. Peaks in the total mass measurement are erroneous readings; as can be seen in Figure 30, there is no volume flow that would cause such peaks. Whereas the weighting system installed in the wagon implies that the final mass of the load is 12000 kg, the reference measurement with a scale produced a value of 8350 kg, thus resulting in an error of 44%.

However, investigating Figure 30 for the volume flow reveals that the increase of the total mass at approximately 525 seconds is not possible; no corresponding volume flow can be seen. Thus, a more correct value for the total mass is 9000 kg with which the error decreases to 7.2%.

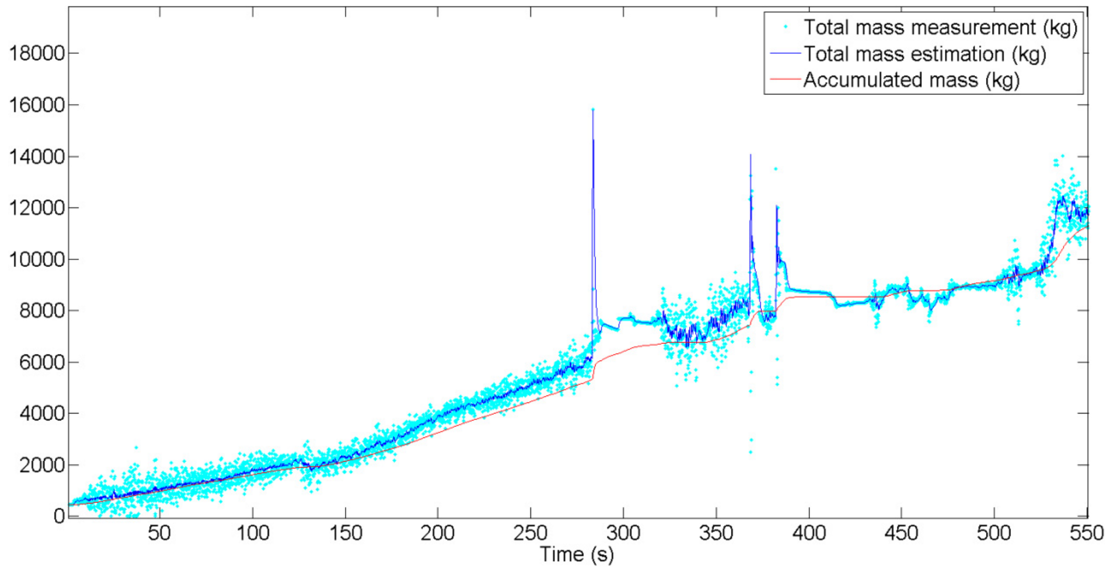

Figure 29: The performance of the Kalman filter. The sharp peaks and a steep increase in the total mass measurement (cyan) are erroneous readings. The final mass is approximately 12000 kg, however, the reference measurement with a scale produced a value of 8350 kg.

The estimated mass flow, density measurement, density estimation, and volume flow measurement are shown in Figure 30. The erroneous readings in the total mass measurement cause peaks in the mass flow and density estimations as well; mass flow has to go high to match the increase in the total mass, and density must go high to get the high mass flow with low volume flow. Since the variance for the density measurement was set as high, the density estimation is little affected by the measurement.

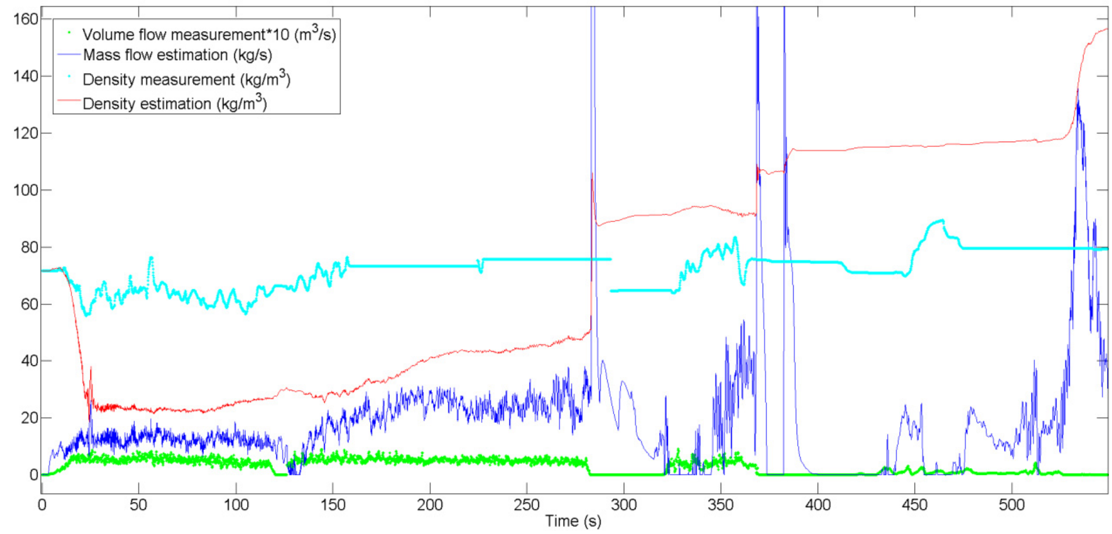

Figure 30: The performance of the Kalman filter. The density estimation (red) does not follow the measurement (cyan) because the measurement variance was set as high.

3.2 Speed control

Speed controller was first tuned on the test track with a flat asphalt surface, thus the conditions differed significantly from the field conditions. The unavailable measurements on the test track were swath size, total mass and moisture measurements. The swath size measurement was simulated with a C# program providing the bus with the appropriate ISO 11783 message. The total mass measurement was replaced by accumulation of the mass flow and the moisture measurement was ignored by setting the density to a constant of 50, thus the Kalman filter was not utilized as it was designed to. Furthermore, the absence of the swath meant that the machine had no loading. Overloading was simulated by manually decreasing the engine RPM.

The fuzzy controller was tuned by a trial and error principle; various rules with different membership functions and scaling coefficients were tested. The final rules and membership functions can be found from APPENDIX A and APPENDIX B**.** The scaling coefficients for the controller were set as follows unless said otherwise: target mass flow was set to 40 kg/s and corresponding standard deviation to 5 kg/s, normal RPM was set to 2000 and standard deviation to 200, median speed was set to 2.5 and standard deviation to 2.0, Δ swath size range was set to 0.2, and output scaling to 0.4.

Figure 31 illustrates a test run with simulated swath area measurement. The mass flow stays around 40 kg/s regardless of the swath size which varies quite randomly with a constant noise. It can be seen that when the speed is high, the system is more sensitive to changes in the swath size; with high speed values, the controller has little time to react to changes, which results in mass flow peaks.

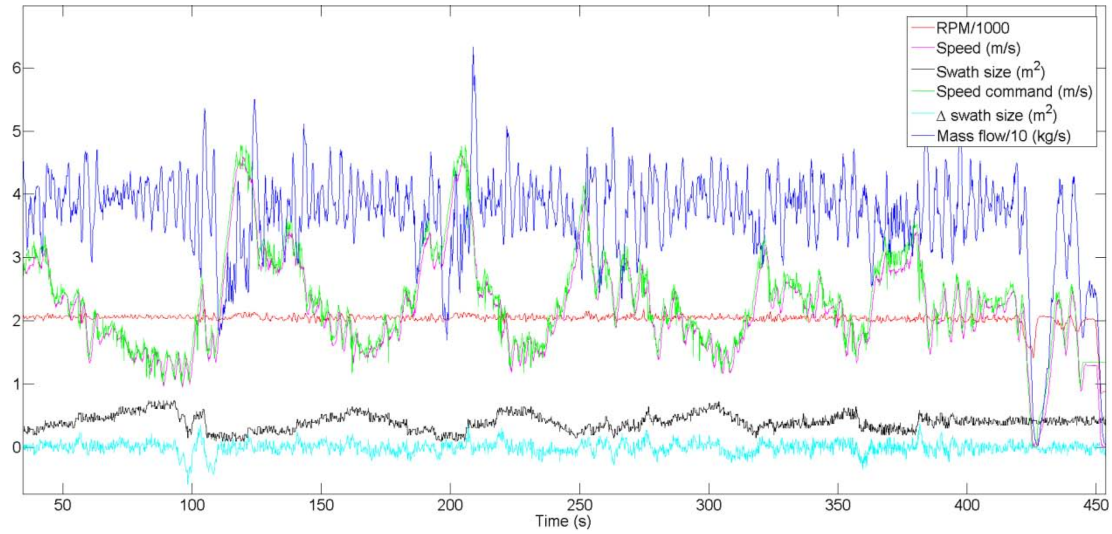

Figure 31: The speed (magenta) is changing according to the swath size (black) in order to maintain the mass flow (blue) at the desired level.

Figure 32 shows a test run where the swath size was increased in steps of 0.2 m^2. In contrast to the previous test, the noise of the swath size measurement was lower and the Δ swath size range was set to 0.1 m^2. An increase in the Δ swath size, meaning the swath size at the cut-and-feed unit is about to increase, causes a steep decrease in the speed command. In addition, the speed is affected by the mass flow so that a flow exceeding 40 kg/s by a certain degree will decrease the speed whereas values below 40 kg/s will increase the speed.

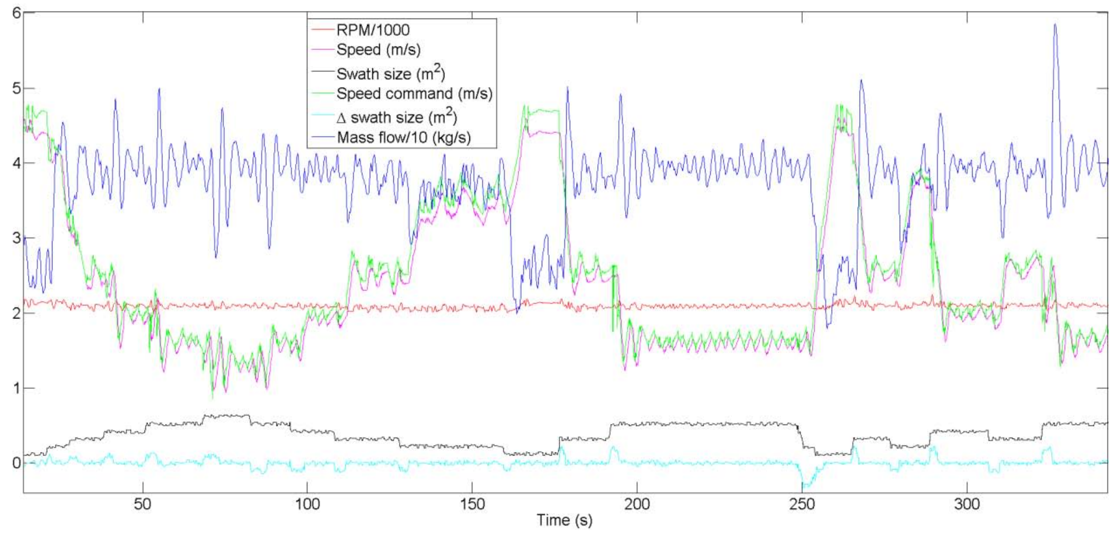

Figure 32: Step responses with simulated swath size measurement (black).

The affect of overloading is shown in Figure 33. The engine RPM was manually decreased below the normal value of 2000 to simulate an overloading scenario. A decrease in RPM causes the speed to decrease in order to lighten the loading of the machine.

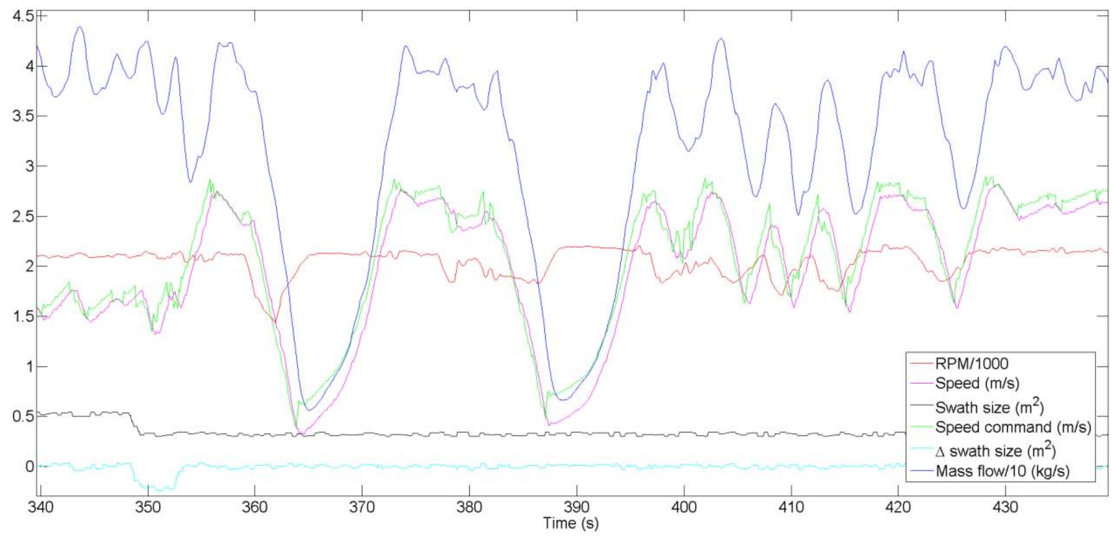

Figure 33: Overloading was simulated by manually decreasing the engine RPM (red). The speed controller responds by decreasing the speed (green) in order to lighten the loading of the machine.

After optimizing the speed controller at the test track, the controller was tested in a field with real swath size, total mass, and moisture measurements. A test run with the target mass flow set to 30 kg/s is shown in Figure 34. In contrast to the test track tests, the machine was under heavy loading, however, the performance of the speed controller did not differ from that of at the test track; the mass flow varies within the chosen boundaries while the speed increases or decreases according to the swath size.

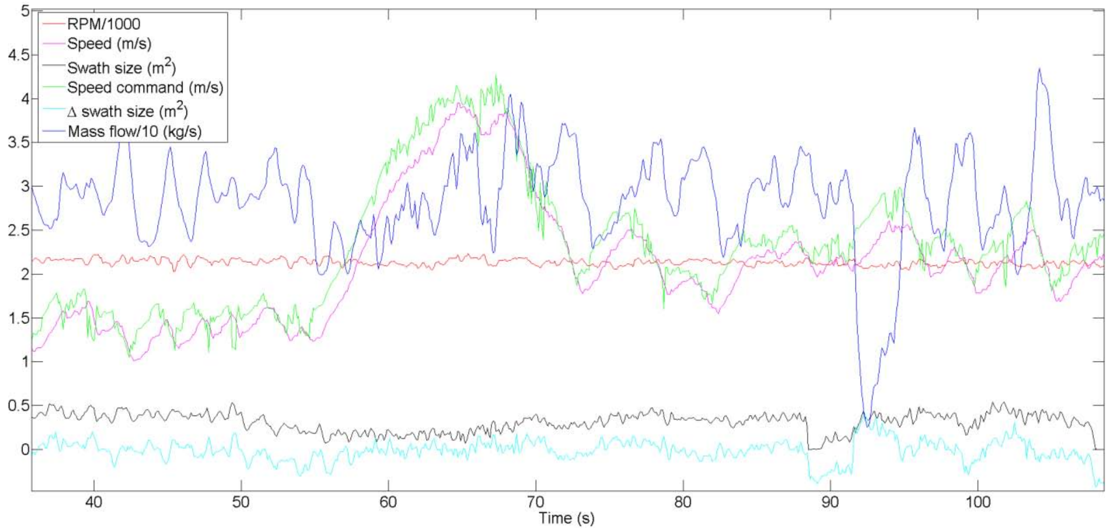

Figure 34: In the field tests, the target mass flow was set to 30 kg/s and the corresponding standard deviation to 5 kg/s. The speed (green) increases when the swath size (black) decreases in order to maintain the mass flow (blue) at the desired level.

A step input with a real swath is shown in Figure 35. The automatic speed control was turned on before driving the machine over the swath, thus causing it to accelerate to the maximum speed. When the machine gets over the swath and laser detects it, the controller then immediately decreases the speed. After the swath reaches the cut-and-feed unit, the excess of mass flow starts to decrease the speed. It took 7 seconds to decrease the speed and reach the target level of mass flow, however, the cut-and-feed unit was not blocked.

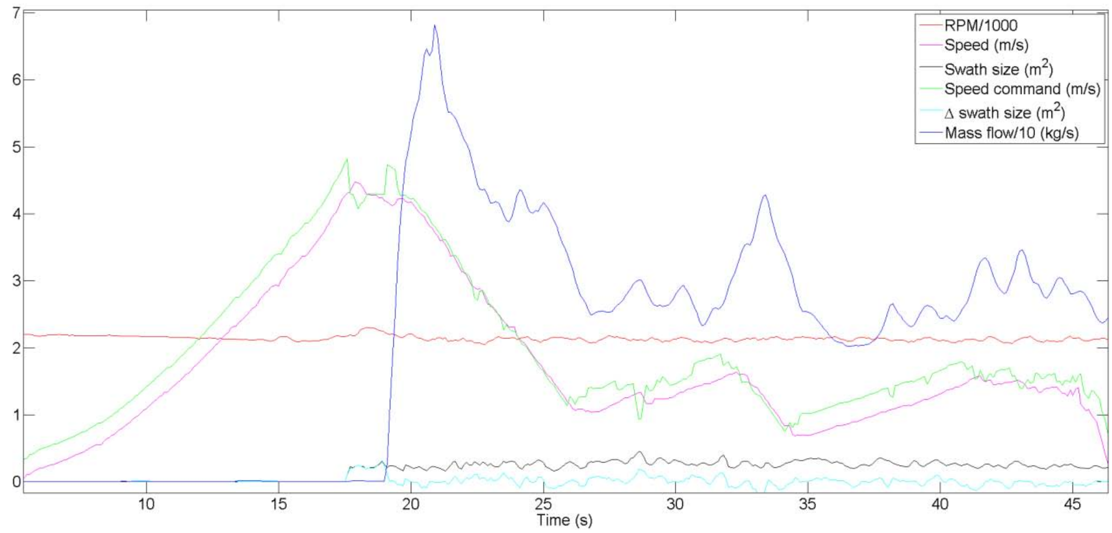

Figure 35: A step response in the field. The tractor hit the swath at 4 m/s, however, the controller was able to decrease the speed (magenta) before the mass flow (blue) rose high enough to block the machine.

3.3 Flow control

3.3.1 Tuning the controller

As explained in the section 1.4.3.1, the first step in tuning a PID controller with Ziegler-Nichols frequency response method is to make the controller purely proportional and increase the P term until the process oscillates. Here, while the process was inputted with simulated mass flow, the P term was increased from 10 to 60 with increments of 10. The target flow, which is derived from the inputted mass flow, as well as the output flow are shown in Figure 36. The gap between the curves is caused by the inability of a proportional controller to remove the error. The columns shown in the figure each represents the output with different P term, from 10 to 60 respectively. Damped oscillation can be seen already when the P term is 50, however, the undamped oscillation occurs with a P term of 60. A close-up of the oscillation is shown in Figure 37 where the oscillation period can be seen to be 0.4 seconds. Thus, the ultimate period $T_u = 0.4$ and the ultimate gain $K_u = 60$. The PID controller parameters can be obtained from Table 1 as follows: $K_p = 36$, $T_i = 0.2$, and $T_d = 0.05$. The gains for the three terms are: $K_p = 36$, $\mathrm{K_i} = \mathrm{K_p}/\mathrm{T_i} = 180$, and $K_d = K_p T_d = 1.8$

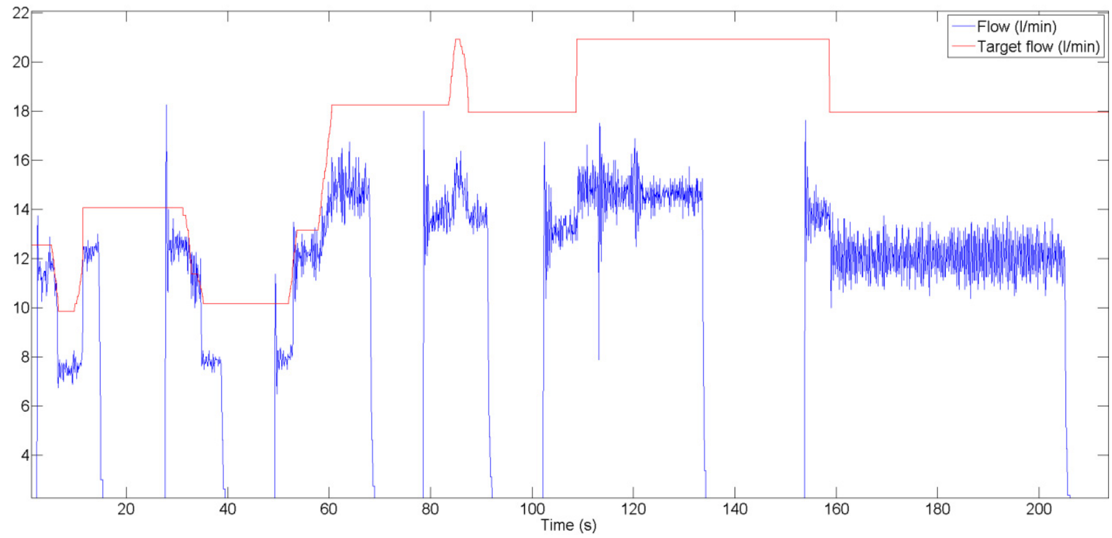

Figure 36: The PID controller was tuned with Ziegler-Nichols frequency response method; the controller was made purely proportional and the P term was increased in steps of 10 until the system starts to oscillate. The oscillation can be seen in the last column when the P term is 60.

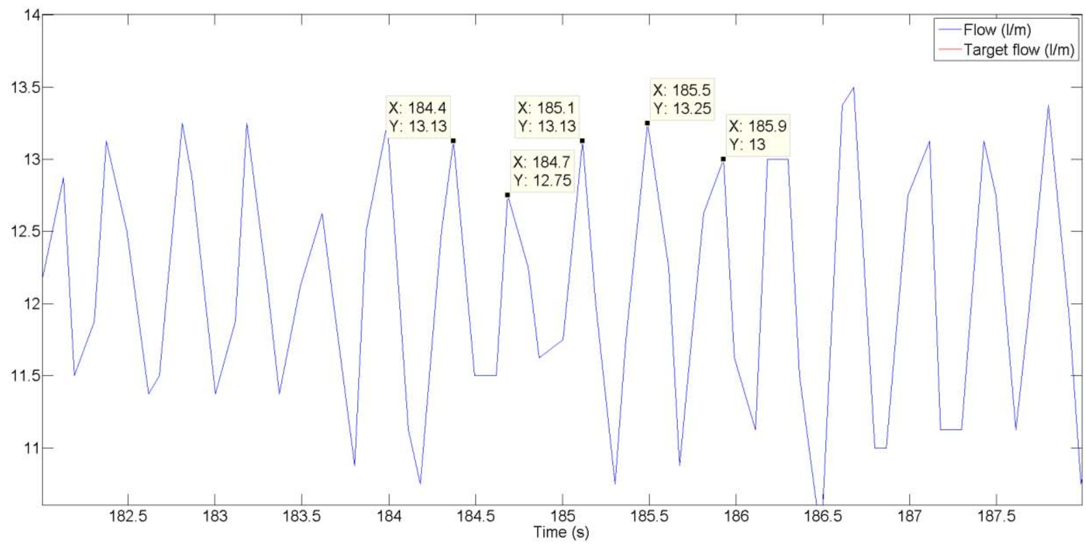

Figure 37: The oscillation period is 0.4 seconds.

3.3.2 Automatic mode

With the PID gains set as previously calculated, the automatic mode of the additive applicator was tested by inputting a simulated mass flow proprietary message on the bus, resulting in a curve seen in Figure 38. As described in section 2.2.2.4, when the target flow is below the minimum flow, the controller switches the minimum flow on and off, which will result in average flow matching the target flow. The actual consumption was measured with a scale to be 11.1 liters which is 2.7% higher than the value of 10.8 liters the controller assumed it to be.

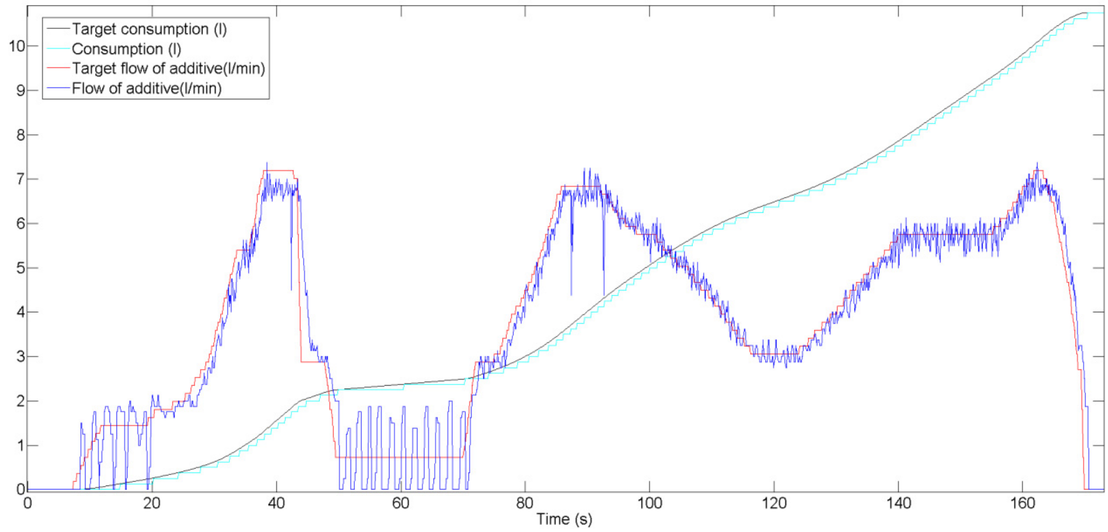

Figure 38: The PID maintains the flow (blue) according to the target flow (red). When the target goes below the minimum flow, the minimum flow is switched on and off to produce an average flow corresponding to the target flow.

3.4 Automated harvesting

The performance of the whole system during two of the harvested loads is presented in the following sections. All the coefficients and parameters were set as previously presented with the exception of the target mass flow set to 30 kg/s. The dosage of additive was set to 3 liters per ton of forage.

3.4.1 The first load

During the first load, the tractor was first driven manually until at approximately 50 seconds, it was stopped and then the speed controller was turned on. At 100 seconds, the machine was again stopped and the automatic mode was switched off due to some problems; however it was switched on again at 200 seconds. Collecting the first swath was finished at approximately 340 seconds, after which the tractor was stopped, and collecting the second swath was started at 350 seconds. This sequence is best seen in Figure 41, but is noticeable in other figures as well.

The performance of the Kalman filter is shown in Figure 39, Figure 40, and Figure 41. The first figure illustrates the total mass measurement, total mass estimation, and accumulated mass. Whereas the reference measurement with a scale produced a value of 8620 kg for the total mass, the weighting system in the loader wagon measured a mass of 8100 kg, thus resulting in an error of 6.0%. At 75 seconds, the total mass decreases, however, the mass flow is restricted from going negative, which causes a gap between the accumulated mass and total mass. The change of swath at 350 second can be seen as a step in the total mass, meaning the measurements before and after the change are not consistent but have some offset.

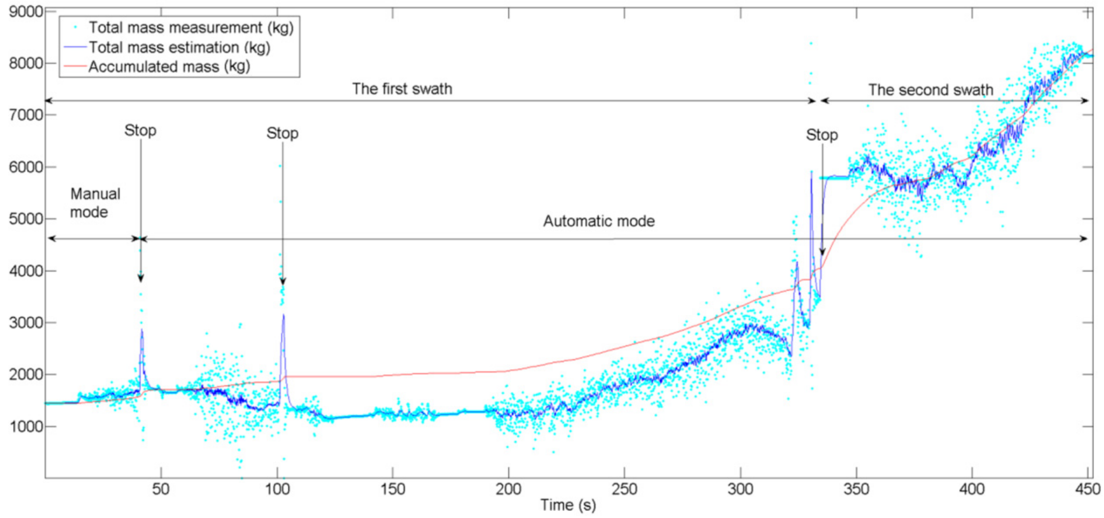

Figure 39: The swath is changed at 350 seconds. Measurements before and after the change are not consistent but have some offset, thus causing a step in the mass measurement (cyan). The mass was measured with a scale to be 8620 kg.

Figure 40 shows the behavior of the density estimation. It can be noted that when total mass decreases at 75 second, the density drops close to zero as the filter tries to explain the decrease in the total mass measurement. Mass flow estimation and other parameters for the speed controller are shown in Figure 41.

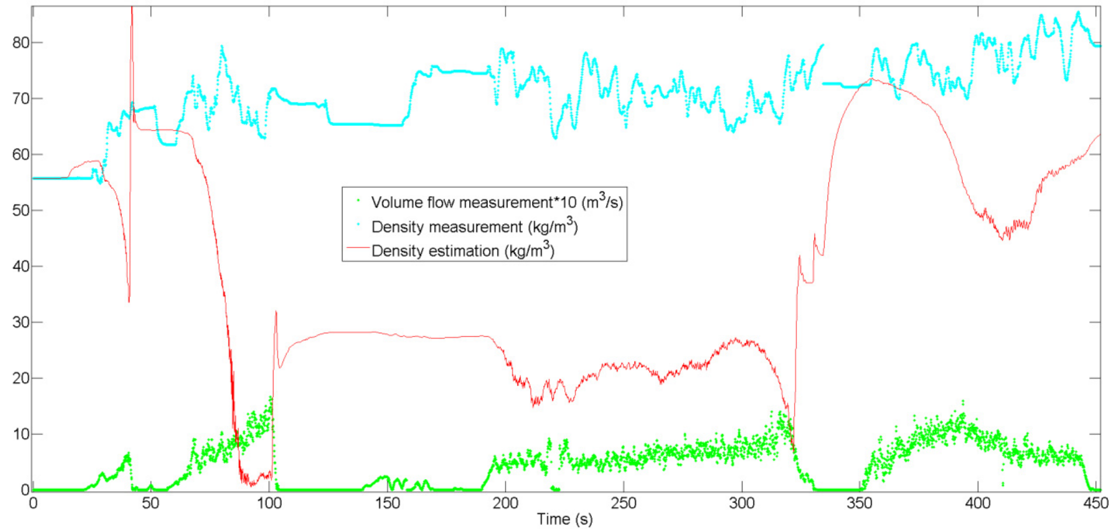

Figure 40: The density estimation (red) decreases to near zero when total mass measurements start to decrease in Figure 39.

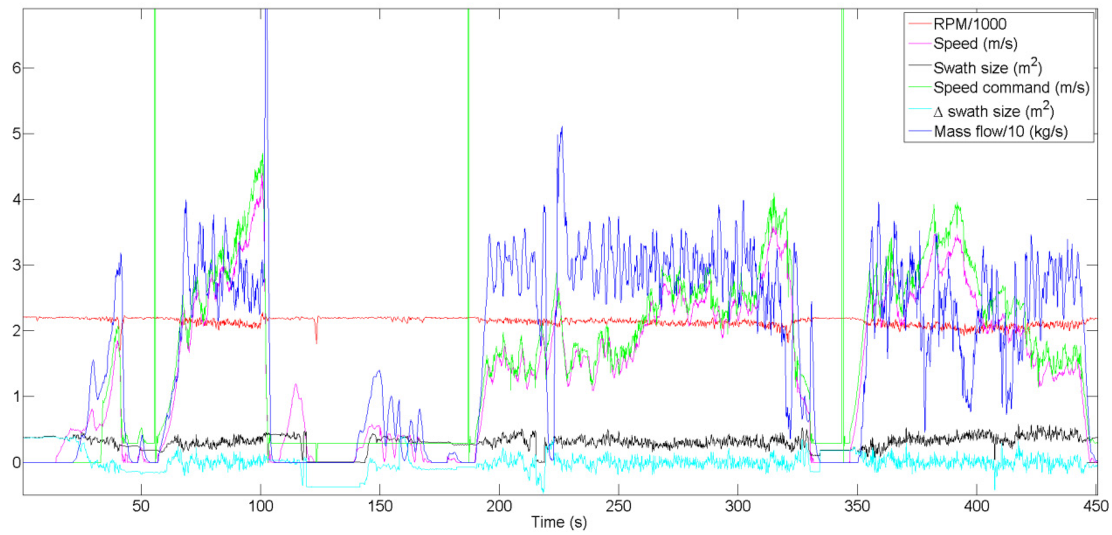

Figure 41: the tractor was first driven manually until at approximately 50 seconds, it was stopped and the speed controller was turned on. At 100 seconds, the machine was again stopped and the automatic mode was switched off due to some problems; however it was switched on again at 200 seconds. Collecting the first swath was finished at approximately 340 seconds, after which the tractor was stopped, and collecting the second swath was started at 350 seconds.

The behaviour of additive flow is presented in Figure 42. The accumulated consumption follows the target consumption to 23.3 liters. However, the reference measurement with a scale indicates the actual consumption is 24.8 liters, which results in an error of 6.0%. As the load was weighted to be 8620 kg and dosing of additive was set to 3 liters per ton, the correct consumption would have been 25.9 liters. Thus, the error in ratio of forage and additive is 4.2%.

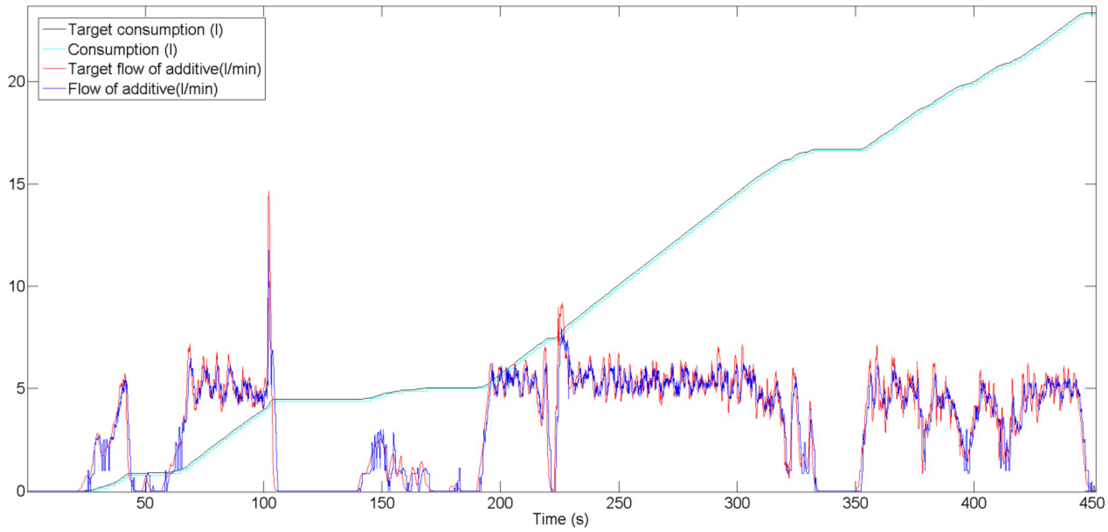

Figure 42: The consumption was measured with a scale to be 24.8 liters, which results in an error of 6.0% compared to the value of 23.3 liters seen in this figure.

3.4.2 The second load

The second load was collected from two swaths. The change of swath can be seen in Figure 43, Figure 44, and Figure 45 at 300 seconds; it caused a step in the total mass measurement. Whereas the reference measurement with a scale produced a value of 7320 kg for the total mass, the weighting system in the loader wagon measured a mass of 7180 kg, thus resulting in an error of 1.9%. The mass estimation and speed control performed as before.

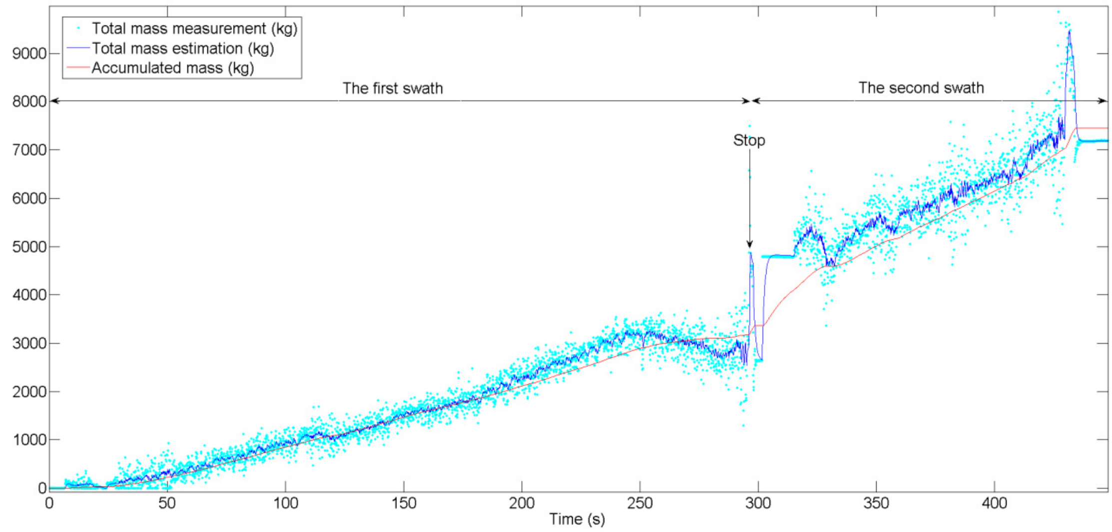

Figure 43: Measurements before and after the change are not consistent but have some offset, thus causing a step in the mass measurement (cyan). The mass was measured with a scale to be 7320 kg.

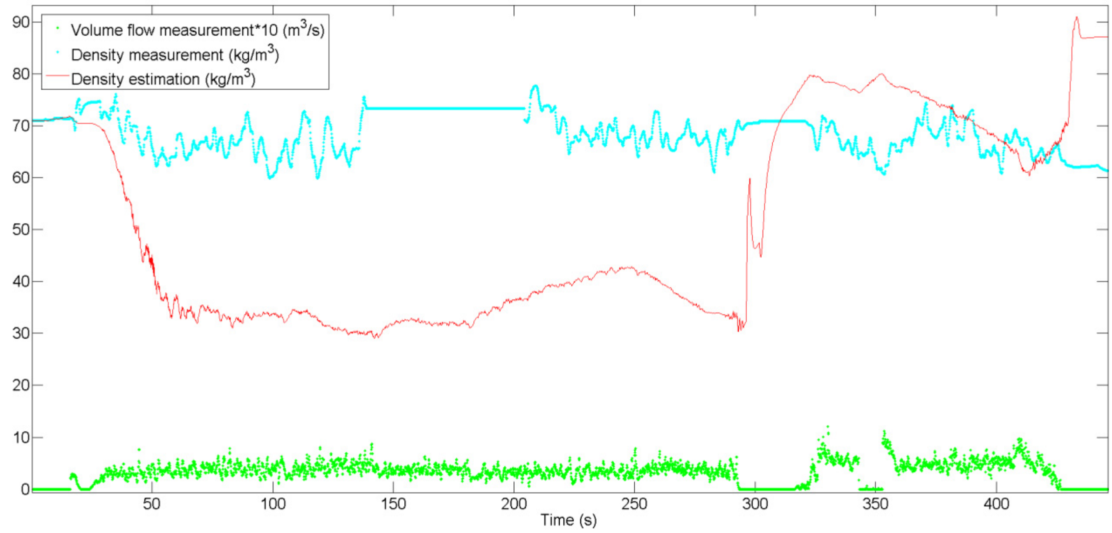

Figure 44: The behavior of volume flow measurement (green), density measurement (cyan), and density estimation (red) during the second load.

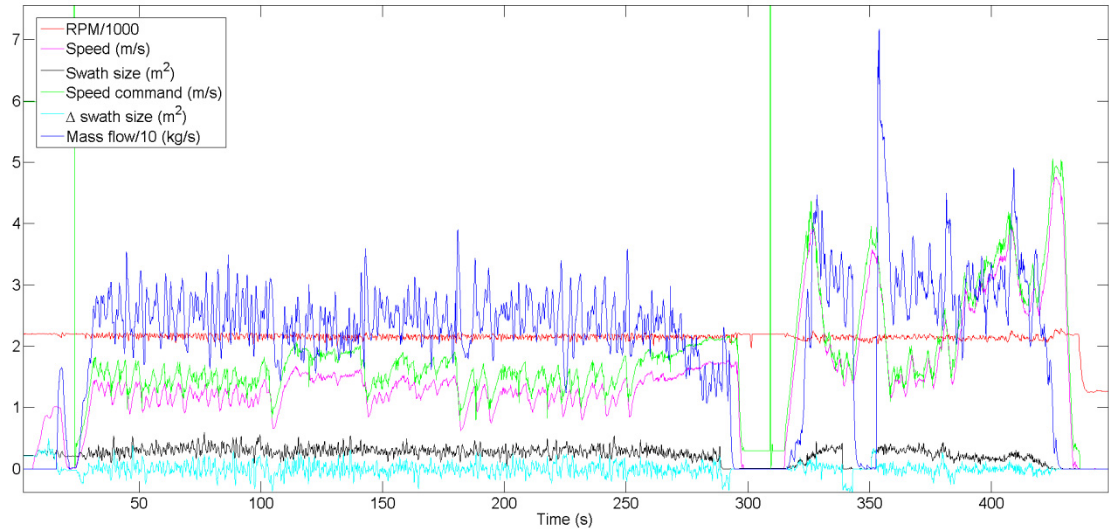

Figure 45: The speed controller was able to maintain the mass flow (blue) at the desired level.

The performance of the additive applicator is shown in Figure 46. Prior to harvesting the second load, the flow meter of the additive applicator was calibrated according to the measurements of the first load. Despite the calibration, the error in the flow measurement was 16.3%; whereas the scale indicated a consumption of 23.9 liters, the flow meter implied it was 27.8 liters. Since the load was 7320 kg and dosing of additive was set to 3 liters per ton, the correct consumption would have been 22.0 liters. Thus, the error in ratio of forage and additive is 26.4%.

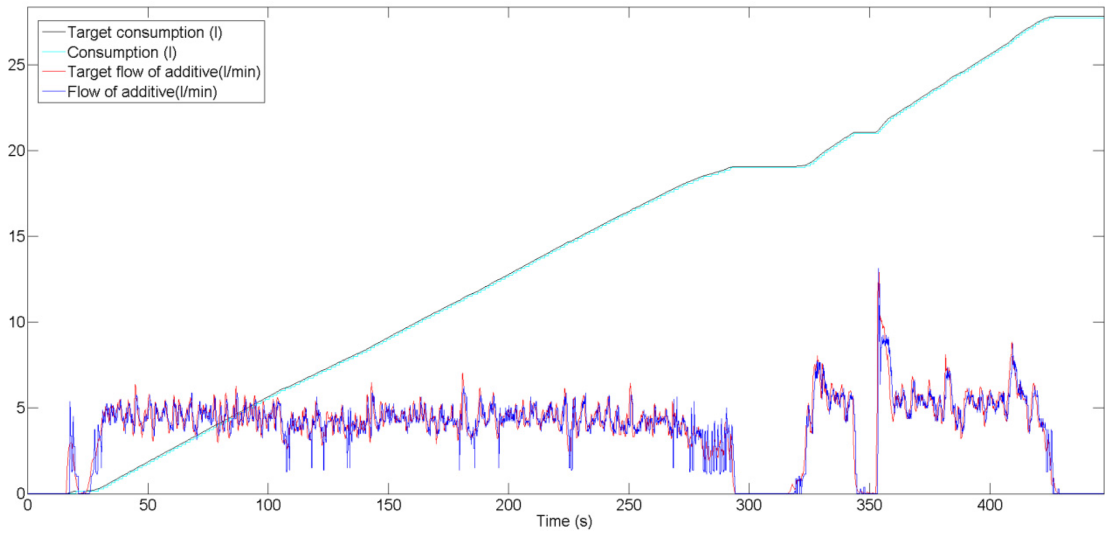

Figure 46: The consumption was measured with a scale to be 23.9 liters, which results in an error of 16.3% compared to the value of 27.8 liters seen in this figure.

3.5 Manual harvesting

In comparison with the automated forage harvesting process, this section presents two examples of the process performed in a conventional way without any automation. For the first drive, the progress of the harvesting process is presented in Figure 47 and Figure 48. First of the figures presents the progress of forage collecting the forage, and the latter presents the process of applying additives. Similarly, the data for the second drive is presented in Figure 49 and Figure 50.

The additive flow was set to a constant flow of 9 liters per minute. In the first drive, the consumption was 4.3% greater than the target, and in the second drive, the error was 111.1%.

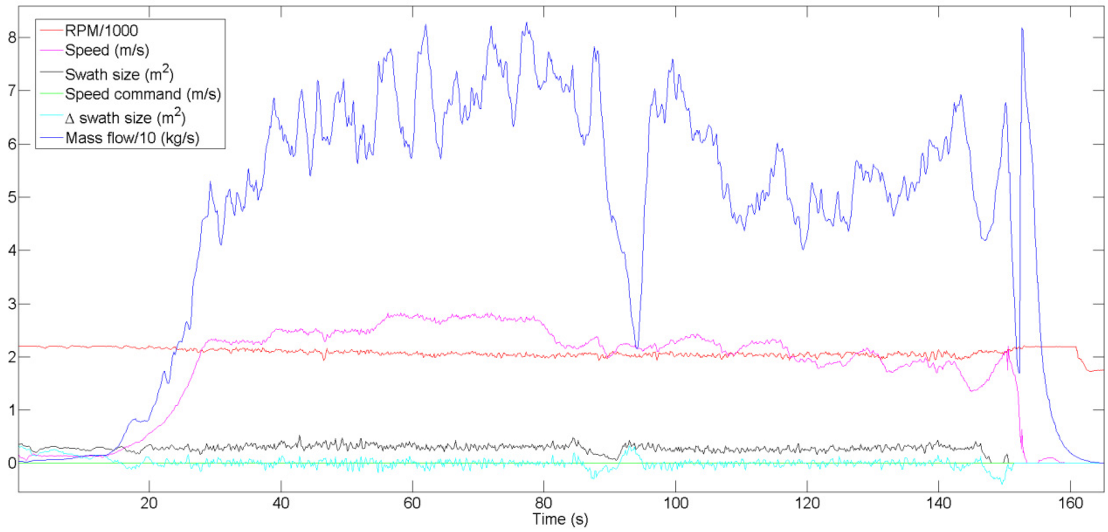

Figure 47: The speed (magenta) is controlled manually by the operator.

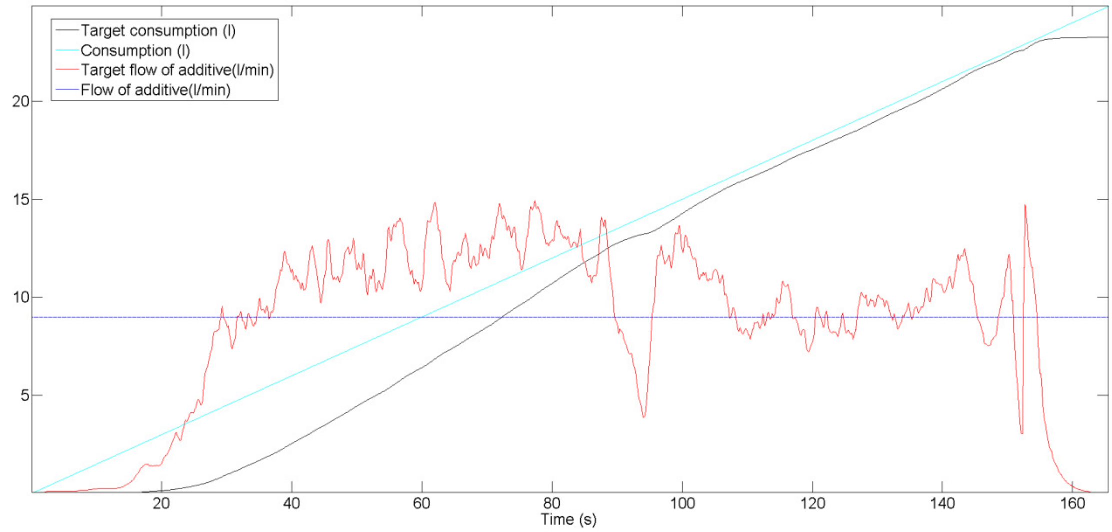

Figure 48: The additive flow (blue) was set to 9 liters per minute. The error in consumption is 4.3%.

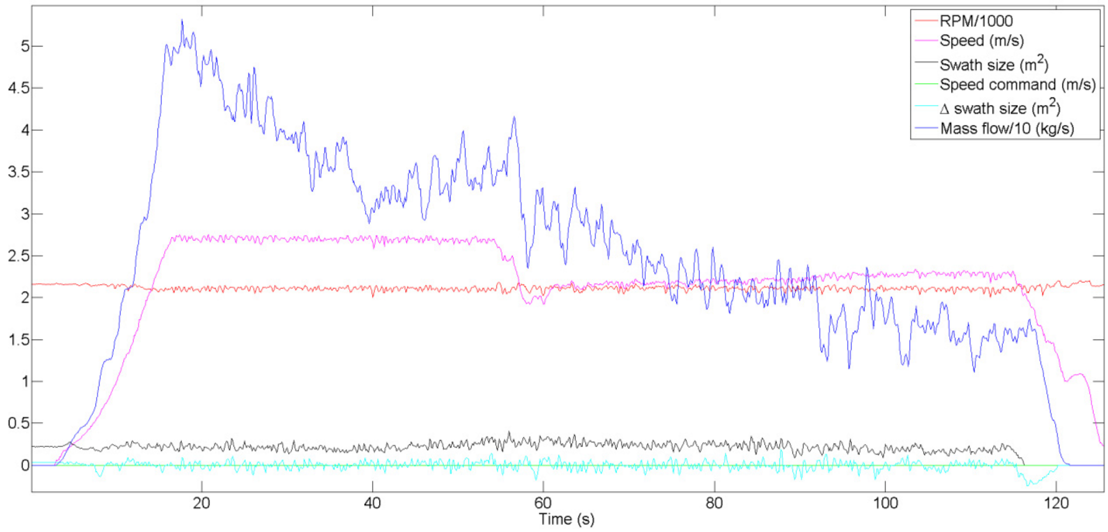

Figure 49: The speed (magenta) is controlled manually by the operator.

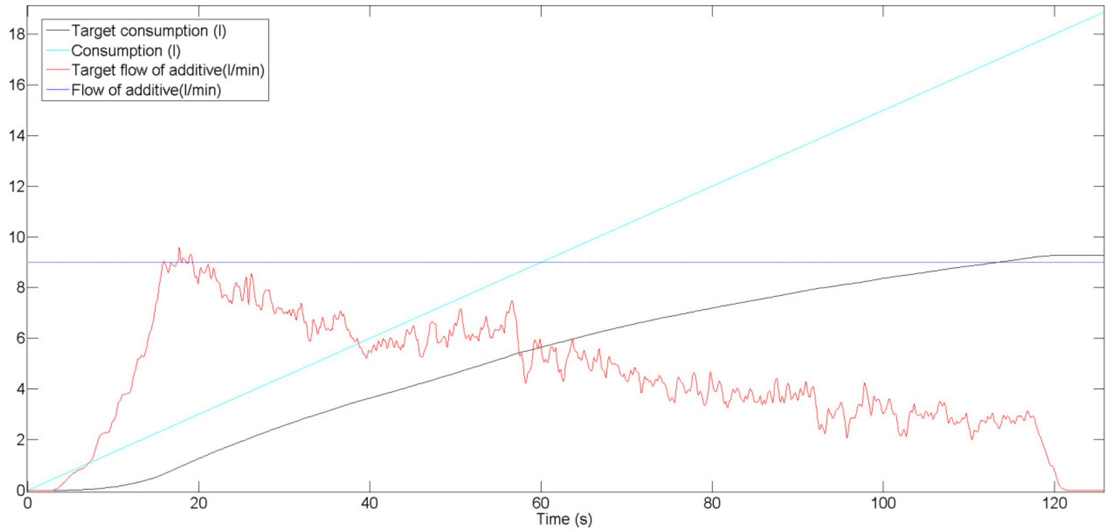

Figure 50: The additive flow (blue) was set to 9 liters per minute. The error in consumption is 111.1%.

4 DISCUSSION

The concept of optimizing forage harvesting process presented in this thesis was proved to be working. The Kalman filter provided the system with good estimates for most of the time. However, some faults in the measurements caused problems for the filter. As the speed controller is trying to maintain a steady mass flow, its performance is dependent on the mass flow estimation provided by the Kalman filter. Assuming a correct estimation for the flow, the performance of the speed controller was in line with the specification. It was able to maintain the target mass flow within the defined boundaries and to prevent blockages even when there was a large sudden increase in the swath size, as in Figure 35. Furthermore, the mass flow estimation is used by the additive applicator for controlling the additive flow. The applicator was able to maintain a flow according to the estimation. However, the precision of the ratio of forage and additive finally depends on the precision of the mass flow estimation. In section 3.4, the errors for the ratio were 4.4% and 26.4%; comparing these with the errors of 4.3% and 111.1% in section 3.5 for the manual drives, the result is satisfactory.

With further development, the system could indeed be used for optimizing the forage harvesting process. It could improve time efficiency, ease the operator’s work, prevent excess consumption of additives, and improve the silage quality. Some proposals regarding the development are presented in the following sections.

4.1 Mass flow estimation

The performance of the Kalman filter is dependent on the measurements. With faulty measurements, the filter is not able to produce estimations corresponding to the real states. Here, the volume flow measurement was accurate in all situations, however, the total mass and density measurements proved to be challenging. The calibration of the pressure sensors allowed quite accurate measurements when the wagon was steady. For instance, the final mass measurements in Figure 39 and Figure 43 have only 1.9-6.0% error compared to the reference value obtained by weighting the load with a scale. However, declination changes of the wagon on the bumpy fields caused the weighting system to produce erroneous measurements; the weighting system provides different readings if the wagon is leaning left or right, or back or forth. An example can be seen in Figure 29 at 525 seconds, implying an increase of 2000 kg in total mass without any volume flow. Furthermore, similar errors can be seen in Figure 39 and Figure 43 where the total mass measured before and after changing the swath differ significantly. When changing the swath, no forage was collected; only the position of the wagon was changed. The change of declination also caused the total mass to decrease, as in Figure 43 at 250 seconds. In addition to the declination of the wagon, the acceleration of the machine affected the mass measurement; by comparing Figure 39 and Figure 41, it is obvious that the sharp peaks in the mass measurement are due to strong braking.

The issues with mass measurement could be solved by either improving the existing measurement system, or by introducing a new system. A possible improvement would be a more careful calibration taking the declination of the wagon into account. Furthermore, sharp peaks in the mass measurement could be removed by filtering the measurements according to acceleration. Another solution is to build a mathematical model of the wagon. The model could relate the pressure sensor readings with the movements of the wagon, thus removing the unwanted effects in the mass measurement. As for the new measurement system, the pressure sensors could be replaced with strain gauges which allow the mass to be measured from the force it is causing to the structure of the wagon. Furthermore, as discussed in section 1.3.2.1, John Deere and Pöttinger utilized PTO torque for calculating the mass flow; mass measurement was not needed at all. However, the tractor used in this thesis did not provide accurate torque measurement, thus a different tractor would have been required for using the same approach.

Another issue was determining the density of the forage. In principle, it should be possible to determine it from the moisture with certain assumptions, however, a proper equation was not found. The issue was solved by simply raising the variance of the density measurement, thus ignoring the measurement.

4.2 Speed control

The speed controller had some minor issues that are needed to be solved. One of issues was that there was slight oscillation in the speed command, which can be seen in Figure 32 between 200 and 250 seconds. However, the oscillation was not noticeable in the tractor cabin. The oscillation is a result of some faults in the design of the fuzzy controller and could be removed by fine tuning the fuzzy rules and the membership functions of the controller.

Another issue was caused by curves in the swath. In order to keep the cut-and-feed unit over the swath, the tractor often needs to be driven off the swath during the curves. When this happens, the laser is not over the swath and is thus measuring the swath size to be zero, as can be seen in Figure 34 at 90 seconds. As a result, the controller assumes that there is a gap in the swath. A solution might be to make the measurement area of the laser wider. This however, might have side effects. In addition, the controller should distinguish if the swath collection is actually completed and turn off the automatic mode in that case.

Furthermore, the user interface of the loader wagon ECU requires redesigning. The configuration parameters of the fuzzy controller, as shown in Figure 20, are difficult for the user to understand, however, they might require adjustment depending on for instance the tractor properties. A more user friendly solution would be to offer a few different working modes, such as a “Low power mode” for tractors with little power. In the “Low power mode” normal speed and target mass flow would be set to slightly lower values, thus decreasing the power requirement.

4.3 Additive applicator

Due to the poor accuracy of the flow meter, the actual additive flow was not what the controller measured it to be. As discussed in section 3.4, the error in flow measurement was 6.0-16.3%. The issue might be solved by a more accurate flow meter or through a better calibration. Since the calibration coefficients seem to be dependent on the flow rate, different coefficients for low rates might be the solution.

Another improvement for the applicator would be to control the dosage of the additive according to the moisture of the forage. In this implementation, the operator sets the dosage according to the mass of forage. However, the optimal dosage is actually more dependent on the amount of water in the forage. The moisture percentage of the forage is already being measured, thus with a few minor changes in the program, the controller could calculate the target additive flow according to the mass flow of water instead of mass flow of forage.

4.4 Tool chain

The two electronic control units implemented in this thesis were among the first developed with the tool chain presented in section 2.1. The tool chain proved to be suited well for developing ISO 11783 control systems. It allows easy and fast developing of both the user interface and the control logic. In this thesis, as discussed in section 2.1.3, PoolEditObjectParser was successfully developed for further automating the development process. However, some deficiencies were found from the Simulink framework; for instance, the tool chain does not perform all the procedures according to the standard when initializing communication with the VT. Furthermore, the framework does not contain function blocks for handling all the messages defined in the standard. However, more blocks are being developed as they are required. At the moment, three more projects are being developed with the tool chain.

5 CONCLUSIONS

The objective of this thesis was to optimize forage harvesting process by utilizing the ISO 11783 communication network. Two electronic control units were developed using a tool chain based on Matlab Simulink with C code generation, PoolEdit with the associated parsers, and Visual Studio with a Windows CE embedded target.

The first of the controllers was developed for optimizing forage collecting with a loader wagon. The controller estimates the mass flow of forage and controls the speed of the machine based on the estimation and the capacity of the tractor. The objective was to maintain the mass flow at a proper level thus preventing blockages in the cut-and-feed unit of the wagon and saving time. The mass flow estimation was implemented using a Kalman filter with measurements of density, volume flow and mass of the collected forage. The speed control was implemented using fuzzy controller based on tractor speed, mass flow, change in the swath size, and engine RPM. The second controller was developed for optimizing the additive application with an additive applicator; it regulates the additive flow according to the estimated mass flow so that the ratio of forage and additive is what desired.

The performance of the system proved to be in line with the specification. The speed controller was able to maintain the mass flow at the desired level and the additive applicator spread the additive with an accurate ratio. With further development, the control system has the potential to make the forage harvesting process significantly more efficient.

BIBLIOGRAPHY

3S-Smart Software Solutions (2011). *CoDeSys.* http://www.3s-software.com/ (accessed September 05, 2011).

Atmel Corporation (2011). "ATmega168 Datasheet." May 2011.

Bar-Shalom, Yaakov, X. Rong Li, and Thiagalingam Kirubarajan (2001). "Estimation with Applications to Tracking and Navigation." 208-217. John Wiley & Sons.

Bosch, Robert, Gmbh (1991). "CAN Specification, Version 2.0." Germany.

CLAAS (2011). *JAGUAR 980-930.* http://www.claas.com (accessed September 5, 2011).

CLAAS (2009). "QUANTIMETER optical sensing system." *Resource: Engineering & Technology for a Sustainable World*, June 2009.

DIN (1991). "DIN 9864."

Epec (2011). *Epec 2024.* http://www.epec.fi/52.html (accessed September 5, 2011).

FIMECC (2011). "Updated program of EFFIMA."

Hoyningen-Huene, Martin von, and Markus Baldinger (2009). "Tractor-Implement-Automation and its application to a tractor-loader wagon combination."

ISO (2011a). *ICS 65.060.10: Agricultural tractors and trailed vehicles.* www.iso.org (accessed October 26, 2011).

ISO (2005a)."ISO 11783 Part 1: General standard for mobile data communication."

ISO (2002b)."ISO 11783 Part 2: Physical layer."

ISO (1998)."ISO 11783 Part 3: Data link layer."

ISO (2004). "ISO 11783 Part 6: Virtual terminal."

ISO (2002a). "ISO 11783 Part 9: Tractor ECU."

ISO (2005b). "ISO 11783 Part 10: Task controller and management information system data interchange."

ISO (2011b). *ISO/FDIS 11783.* http://www.iso.org (accessed September 19, 2011).

Junkkari (2009). *HP-1000 Acid applicator manual.*

National Instruments (2011). *LabVIEW.* http://www.ni.com/labview (accessed September 5, 2011).

NMEA (2000). "NMEA 2000:The network standard for interfacing marine electronics devices."

Oksanen, Timo, Antti Kunnas, and Arto Visala (2011). "Development and Runtime Environment for Embedded Controller supporting ISO 11783 Standard." *18th IFAC World Congress, IFAC 2011.* Milano. 2895-2900.

Oksanen, Timo, Matti Öhman, Mikko Miettinen, and Arto Visala (2005). "ISO 11783 - Standard and its implementation." *IFAC.*

OSB (2011a). *ISOAgLib.* http://isoaglib.com/ (accessed September 5, 2011).

OSB (2011b). *vt-designer.* http://www.vt-designer.com (accessed September 5, 2011).

PRLog (2010). "New ISO standard terminal from Parker." April 08, 2010. http://www.prlog.org/10616169-new-iso-standard-terminal-from-parker.html (accessed November 1, 2011).

Process Sensors Corporation (2007). "Multiple Constituent Transmitter; MCT Series; Installation and Operation Manual." July 2007.

SAE (2011). "SAE J1939 - Recommended Practice for a Serial Control and Communications Vehicle Network."

SICK (2003). "LM S221/220 Tecnical Information." July 2003.

Stone, Marvin L., Kevin D. McKee, C. William Formwalt, and Robert K. Benneweis (1999). "ISO 11783: An Electronic Communications Protocol For Agricultural Equipment." *Agricultural Equipment Technology Conference.*

Suokannas, Antti (2006). "Karhotus tehostaa korjuuta isoilla koneilla." *Maito ja me.* no. 2006/3. May 17, 2006.

Suokannas, Antti, and Anna Sipilä (2008). "Esikuivatetun säilörehun korjuumenetelmät." *Nurmitieto.*

Suokannas, Antti, and Matts Nysand (2006). "Silpun pituus." *Koneviesti*, January 13, 2006: 46-49.

Tella, Riku (2007). "Säilörehu - tilan vahvin vai heikoin lenkki." *Maito ja me.* no. 2007/3. May 2, 2007.

The MathWorks, Inc (2011a). *Foundations of Fuzzy Logic.*

The MathWorks, Inc (2011b). *Fuzzy Interference Systems.*

The MathWorks, Inc (2011c). *Simulink.* http://www.mathworks.se/products/simulink/ (accessed October 26, 2011).

Toradex AG (2010a). "Colibri XScale PXA320 Datasheet."

Toradex AG (2010a). "Protea Datasheet."

Valtra (2011a). "Valtra Image Bank." https://pictures.valtra.com/kuvat/6744hq.jpg (accessed November 1, 2011).

Valtra (2011b). "Valtra Image Bank." https://pictures.valtra.com/kuvat/6822.jpg (accessed November 1, 2011).

VDMA (2002). "ISOBUS Specifikation, Implementation Level 1." *VDMA Landtechnik.* Frankfurt am Main.

Visala, Arto (2011). "Agromassi project description."

Åström, Karl Johan (2002). "Control System Design." 216-234.

Öhman, Matti, and Arto Visala (2006). "Design and Implementation of Machine Control Systems with Modern Software Development Tools." *Field and Service Robotics. Corke, P. and Sukkarieh, S.*, 2006: 377-388.

Öhman, Matti, Jouko Kalmari, and Arto Visala (2008). "XML Based Graphical User Interface Editor and Runtime Parser for ISO 11783 Machine Automation Systems." *IFAC World Gongress 2008.* South Korea.

APPENDIX A: MEMBERSHIP FUNCTIONS FOR THE FUZZY CONTROLLER

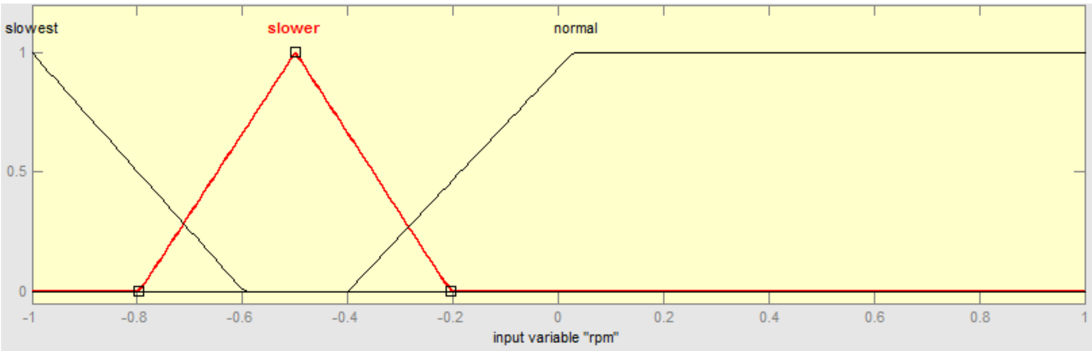

Figure 1: Membership function for RPM

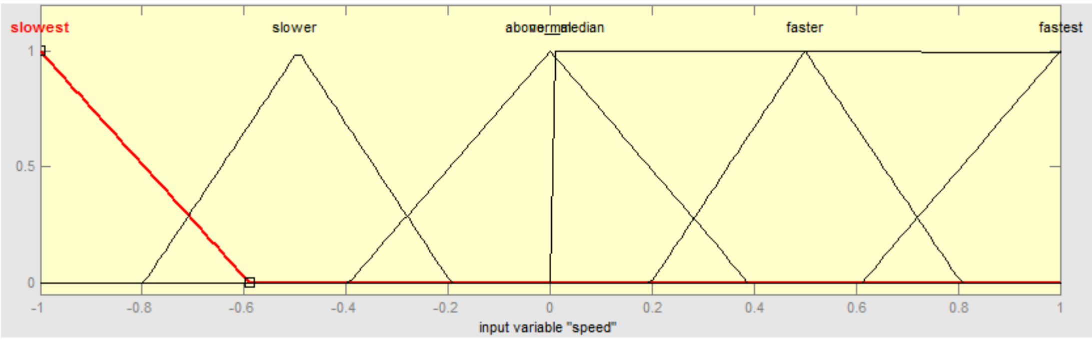

Figure 2: Membership function for speed

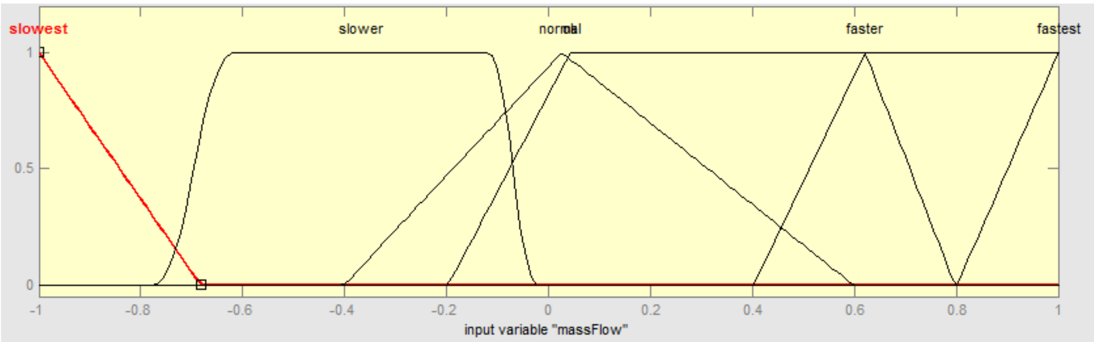

Figure 3: Membership function for mass flow

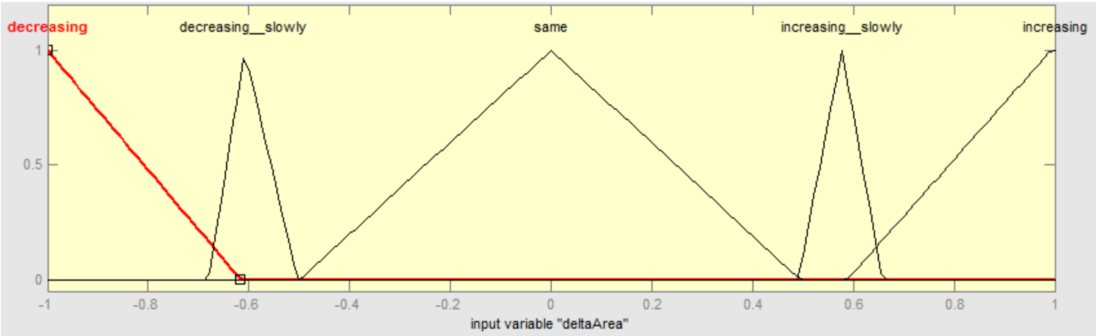

Figure 4: Membership function for Δ swath size

APPENDIX B: RULES FOR THE FUZZY CONTROLLER

- 1. If (rpm is normal) and (speed is not fastest) and (massFlow is slowest) then (speedOut is increase) (1)
- 2. If (rpm is normal) and (speed is not fastest) and (massFlow is slower) and (deltaArea is not increasing) then (speedOut is slight__increase) (1)
- 3. If (rpm is normal) and (speed is not slowest) and (massFlow is fastest) then (speedOut is decrease) (1)
- 4. If (rpm is slowest) then (speedOut is decrease) (1)
- 5. If (speed is not slowest) and (massFlow is ok) and (deltaArea is increasing) then (speedOut is decrease_a_lot) (1)
- 6. If (speed is not slowest) and (massFlow is ok) and (deltaArea is increasing__slowly) then (speedOut is decrease) (1)
- 7. If (rpm is slower) then (speedOut is slight__decrease) (1)
- 8. If (rpm is slowest) and (massFlow is ok) then (speedOut is decrease) (1)
- 9. If (rpm is slower) and (massFlow is ok) then (speedOut is slight__decrease) (1)
- 10. If (speed is above__median) and (deltaArea is increasing) then (speedOut is decrease__more) (1)
- 11. If (speed is above__median) and (massFlow is not slowest) and (deltaArea is increasing__slowly) then (speedOut is decrease_a_lot) (1)
- 12. If (rpm is normal) and (massFlow is normal) and (deltaArea is same) then (speedOut is maintain) (1)

APPENDIX C: SCHEMATIC FOR THE I/O BOARD