Department of Electrical Engineering and Automation

Standardization proposal on Implement guidance for ISO 11783 compatible tractor-implement systems

Timo Oksanen, Juha Backman

Aalto University

SCIENCE + TECHNOLOGY

WORKING PAPERS

Aalto University publication series
SCIENCE + **TECHNOLOGY** 2/2015

Standardization proposal on Implement guidance for ISO 11783 compatible tractor-implement systems

Timo Oksanen, Juha Backman

Aalto University
School of Electrical Engineering
Dept. of Electrical Engineering and Automation

Timo Oksanen
E-mail: timo.oksanen@aalto.fi
Phone: +358-9-47001

Aalto University publication series
SCIENCE + **TECHNOLOGY** 2/2015

ISBN 978-952-60-6165-8 (pdf)
ISSN-L 1799-4896
ISSN 1799-4896 (printed)
ISSN 1799-490X (pdf)
http://urn.fi/URN:ISBN:978-952-60-6165-8

March 2015
Unigrafia Oy
Helsinki
2015

Finland

Standardization proposal on Implement guidance for ISO 11783 compatible tractor-implement systems

Preface

This working paper presents a proposal for the standardization in ISO. The scope of the proposal is in agricultural machinery electronic communication and control. We propose the standard to be defined for the multibrand systems to control the steering of an implement(s) connected to the tractor.

The idea of the proposal was briefly summarized in the article ”Backman et al. 2013: Applicability of the ISO 11783 network in a distributed combined guidance system for agricultural machines, published in Biosystems Engineering, Vol 114(3)”.

Since that, we have studied the details of the idea, roughly from September 2013 to December 2014. The discussions with the group in ISO have motivated us to complete the proposal.

The inter-compatibility of guidance systems was proven to be very challenging in multi brand systems. The group focused on communication protocol faces a new challenge as the protocol meets the control science, inevitably. In this proposal, we have combined our knowledge on communication protocols to the understanding about dynamics and automatic control and also the requirements of multibrand systems in the market.

The standard defining the messaging for tractor-implement communication is prepared by ISO and known as ISO 11783 series. Our proposal is compatible with the existing standard and there are no backwards compatibility issues related to implement steering. We see that the proposal could found a base for a new part for ISO 11783, to define messaging and architecture for implement guidance.

Helsinki, 31th March 2015
Timo Oksanen & Juha Backman

Contents

List of Definitions and Abbreviations

Abstract Implement	Skeleton representing the virtual structure of an implement
CP	Connection Point (of the tractor and the implement)
DOF	Degree of Freedom (mechanical)
Essential DOF	1) Side offset and 2) Heading offset of Implement Tool
GCP	Guidance Control Point, the point that needs to follow path in position and orientation
GUIDANCE	Combined Guidance Controller; a navigation system that is connected to ISO 11783 network, capable of guiding both a tractor and an implement of the same mobile system.
IGPP	Implement Guidance Parameter Pool; a set of static parameters describing scale and dynamics of the Abstract Implement
Implement	The complete machine connected to tractor; e.g. trailed mower
Implement Tool	A mechanical part of implement that does the operation, e.g. coulters
TRP	True Rotation Point
VJP	Virtual Joint Position; describes implement mechanical structure
Side offset (ΔS)	Describes how CP moves sideways/horizontally relative to TRP
Heading offset ($\Delta\theta$)	Describes how Implement Tool rotates relative to vertical axis

1. Introduction

1.1 In short

The ISO 11783 standard series is defined for agricultural tractor-implement communication. The farmers know the standard by its market name: ISOBUS. The standard series contains currently 14 parts. The part 1 is the introduction and general requirements, the parts 2-5 & 12 define the lower layers of the communication stack equivalent to ISO/OSI model layers 1-4. The parts from 6 to 14 define both application and presentation layers and in many parts these are mixed, so the relationship of the parts is not always clear.

ISO 11783-7 defines communication protocol for tractor guidance, i.e. how a guidance system may command the steering wheels of the tractor. The standard is defined to create a link between two manufacturers. A farmer is the typical integrator of the system, to plug-and-play requirements have been taken seriously in all parts of the ISO 11783 standard series.

However, ISO 11783 series does not contain any messaging that would allow the so called functionality Implement Guidance or Implement Steering. There are available Implement Steeering product in the market, but they are non-ISOBUS related systems, that are made by one brand. However, it appears that the Implement Guidance will be one of the future functionalities that supplement ISOBUS, especially in the implement that require constant steering, like potato harvesters or precision navigation needs in row crop production.

In this paper, we discuss the requirements for the standardization and present the proposal for the functionality and messaging. Some details are clear while the other require more discussion as the architecture to be followed in ISO 11783 is a topic for large group discussion.

The underlying idea is to follow the same principle as in the guidance of tractor: to abstract the actual steering actuators and use a simple interface that is powerful enough to describe all Essential degrees of freedom (DOF). In this proposal, the selected variables are: side offset and heading offset of Implement Tool. The definition of both comes from the geometry of an implement; the implement wheels (or other similar non-side-slipping part) is positioned upwards direction and two variables describe how CP moves laterally and how Implement Tool rotates. This corresponds to steady-state condition where CP is towed infinitely upwards.

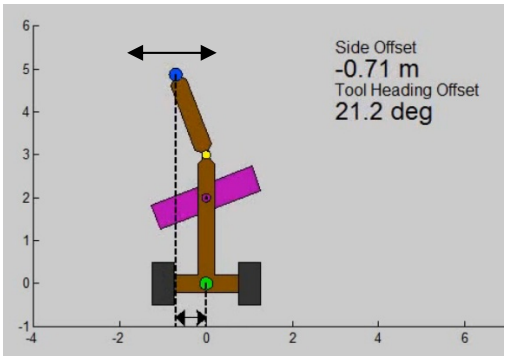


Figure 1. Side offset of towed implement.

In case of **any** implement, see Figure 1: wheels point **always** upwards, TRP always in origin or coordinate system, **side offset** measures how CP moves respect to TRP laterally (projected distance along x-axis) and tool **heading** offset measures what is the angle between Implement Tool and y-axis. See details in Appendices A-C.

The proposal supports natively and easily both Essential Degrees of Freedom (DOF), so both position and orientation of implement can be controlled; but it is possible to utilize only one of them in case only one actuator is utilized.

For the IMPLEMENT manufacturer the interface is **simple**. The required functions are: the closed loop control of actuators and conversion equations between actuators positions (angle or length) to real-time parameters of Abstract Implement (side offset & heading offset). Numerous examples of common types of implements will be presented in the standard with validated equations.

For the GUIDANCE manufacturer the interface is **simple**. All implements are abstracted to the same level (skeleton), and a common kinematic model can be used to describe the behaviour. Development of guidance control algorithms can be done without knowing the actual mechanical structure of implement. This is a key to enable ISOBUS plug-and-play.

The GUIDANCE collects the constant parameters of IMPLEMENT by requesting Implement Guidance Parameter Pool (IGPP), that is transferred by using ISO TP. IMPLEMENT must support IGPP, but a few parameters may be marked as *not available*. IGPP is different from DDOP and this proposal is designed in a way that TC stack is not needed at all. Therefore ISOBUS Implement Guidance interface is completely independent from ISOBUS Task Controller.

1.2 Motivation and requirements

The ISOBUS GUIDANCE system consists of three main components: a Tractor, an Implement and a Combined Guidance Controller (GUIDANCE); all of which may be manufactured by **different** manufacturer; e.g. A, B and C. The system may also contain other ISO 11783 components, like Virtual Terminal (VT) or Task Controller (TC). GUIDANCE may be integrated in TC, or have the functions of TC, or be completely independent of TC.

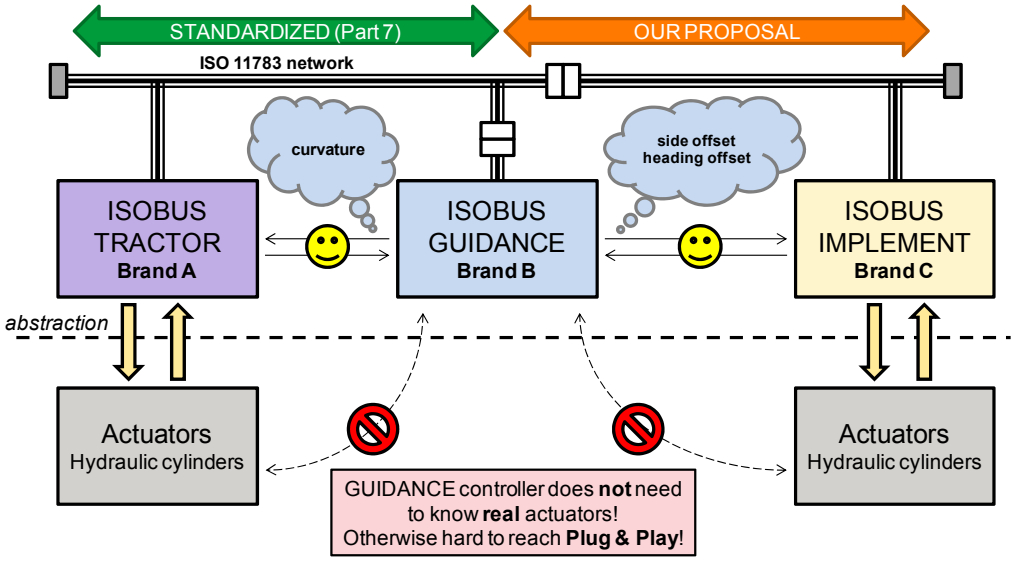


Figure 2. Big picture: the guidance works on abstract level; the real actuator are not considered in GUIDANCE. This is the way to reach ISOBUS plug&play.

Figure 2 shows the Big Picture of our proposal. In a tractor, the actuators are abstracted to curvature, so the GUIDANCE controller does not have to know anything about the real actuators, but just command & measure the curvature in navigation. This proposal follows the same way, trying to abstract an implement (as an entity) to a common framework which is the same for all implements. Therefore, GUIDANCE does not have to know anything about the actuators of the implement, but it may limit understanding only on the virtual model, or the abstract implement, or virtual skeleton of the implement.

In a typical configuration, the GUIDANCE is located in the tractor cabin; it may have it's own GNSS receiver(s) or use the GNSS receiver(s) connected via ISO 11783 network; it may have it's own display/HMI or use Virtual Terminal via ISO 11783 network. GUIDANCE is considered as an **extension** to the current commercial guidance systems available for agricultural machines; to control more Degrees of Freedom (DOF) of the connected system than the one in the tractor (front wheel steering, rear wheel steering, articulated steering, four wheel steering or skid steering).

The current implement steering systems work independently from tractor guidance; or they use proprietary communication to combine these. The main motivation of this proposal is to allow a GUIDANCE controller of Brand A to command both Brand B tractor and Brand C implement synchronously; in order to provide accuracy, performance and manouverability for the connected system. The only realistic way is to do this over ISO 11783 network.

The system bandwidth is limited by ISO 11783 to 10Hz. This applies also to tractor guidance. The combined guidance system may not increase busload remarkably. Preferably 10+10 single frame messages per second between GUIDANCE and Tractor (already in Part 7) and 10+10 single frame messages per second between GUIDANCE and Implement (see sketch in Chapter 2). Therefore this proposed implement guidance interface increases **bus load** only about 1%; both in case of a single actuator and in case of multiple actuators in the implement.

A connected system (tractor+implement) containing more than one DOF (the curvature of tractor), is more challenging to control, or to **design control laws** for, so the engineering challenge is set to especially to the manufacturers of guidance systems. In case of one DOF (like front wheel steering + path following) simple control laws may be used, like PID, but in case of two or three DOF (like presented in this paper), the design challenge is greater. It is not only number of DOF, but also **dynamics parameters** (lag, time constant, max. rate) of each motion. Closed-loop control of **unstable system** (a tractor following a path) with several degrees of freedom needs to be designed carefully, in order not to be unstable – in all conditions, based only on the information received not earlier than the farmer plugs the system together.

The ISOBUS system **integrator** is a farmer. He/she is not an engineer. The system of three ISOBUS guidance system components should be *plug-and-play*, without any service people needed to set up a three-brand-system. The number of combinations will be so large that it is not possible for any manufacturer to test all the combinations, so the challenge of standard development process and system architecture must *admit and address* this fact. In case ISOBUS Implement Steering is adapted by several manufacturers, it is no more possible to develop GUIDANCE controller devices machine-specific (one subprogram per each machine on the market), but there will be more demands for adaptive software. The best practice for the industry would be to design the standard in a way that programming & testing effort at both sides (implement manufacturer + guidance manufacturer) is minimized and the result would be easily reached plug-and-play system. Just the same as in every other ISOBUS functionality.

In case the actuators are controlled by using TIM (e.g. tractor remote control commands for hydraulic valves), this subsystem may require the tuning of control loops in the implement depending on tractor flows, a means to tune the gains of control loops or similar parameters should be available for a farmer. However, this problem is separate from GUIDANCE, at first the farmer should tune TIM, and then the GUIDANCE system may adapt to the closed control system performance.

For possibilities to use this interface for third-party add-on kits, see discussion in Appendix H.

This interface should be a new **functionality** in AEF terminology, to show support for the interface in both GUIDANCE and IMPLEMENT devices. Functionality should be tested in AEF conformance test, just like all the other functionalities. The proposed name for the interface is *Implement Guidance* and some abbreviation for that is needed.

The standards should not force manufacturers to any specific algorithms or mathematics when it comes to the implementation of GUIDANCE. The standard should leave doors open for various approaches, like to adapt current systems for the standard one with reasonable effort and on the other hand allow mathematically oriented adaptive GUIDANCE systems. We see the levels of GUIDANCE algorithms (where each manufacturer may select) are:

1. **Very simple**: simple PID controller, parameters are not requested from the implement, the user has to enter the required parameters (e.g. the length of body); user may need to adjust controller gain by hand
2. **Simple**: e.g. simple PID controller, parameter are received from the implement by using request of Implement Guidance Parameter Pool. Tuning may be done semi-automatically by using e.g. gain scheduling.
3. **Advanced**: multidimensional control algorithm (e.g. LQR), that controls both the tractor and implement synchronously, based on some model representing the connected system.
4. **Very advanced**: both the tractor and the implement(s) are controlled synchronously. State estimation is used to estimate the true state (compensating the control delays) and model based control is used to reach the best possible accuracy in path tracking.

This proposal is designed to support all these levels equally. Therefore, this proposal is not forcing GUIDANCE manufacturers to select one of the levels, but leaves doors open for various implementations. For the IMPLEMENT manufacturer, the interface is always the same, independent of which level the GUIDANCE manufacturer has selected.

2. Abstract implement

2.1 Definition

A*bstract Implement* model is proposed, it is a **skeleton** that represents the virtual structure of an implement. The underlying idea of Abstract implement is to abstract the real kinematics of the implement, so that GUIDANCE may be designed against the Abstract implement, without any need to know the actual location of actuators. Therefore designing GUIDANCE becomes remarkably easier and reaching ISOBUS plug-and-play requirements becomes true. However, the abstraction must not be done at too abstract level, as then the information would be lost – this proposal tries to be the best compromise between these two objectives.

Abstract implement is a skeleton that represents the necessary structure of the real implement. See the flow of abstraction in Figure 3. Any implement can be presented by using three points:

- CP – Connection Point (blue)
- TRP – True Rotation Point (ground point, green)
- VJP – Virtual Joint Point (yellow)

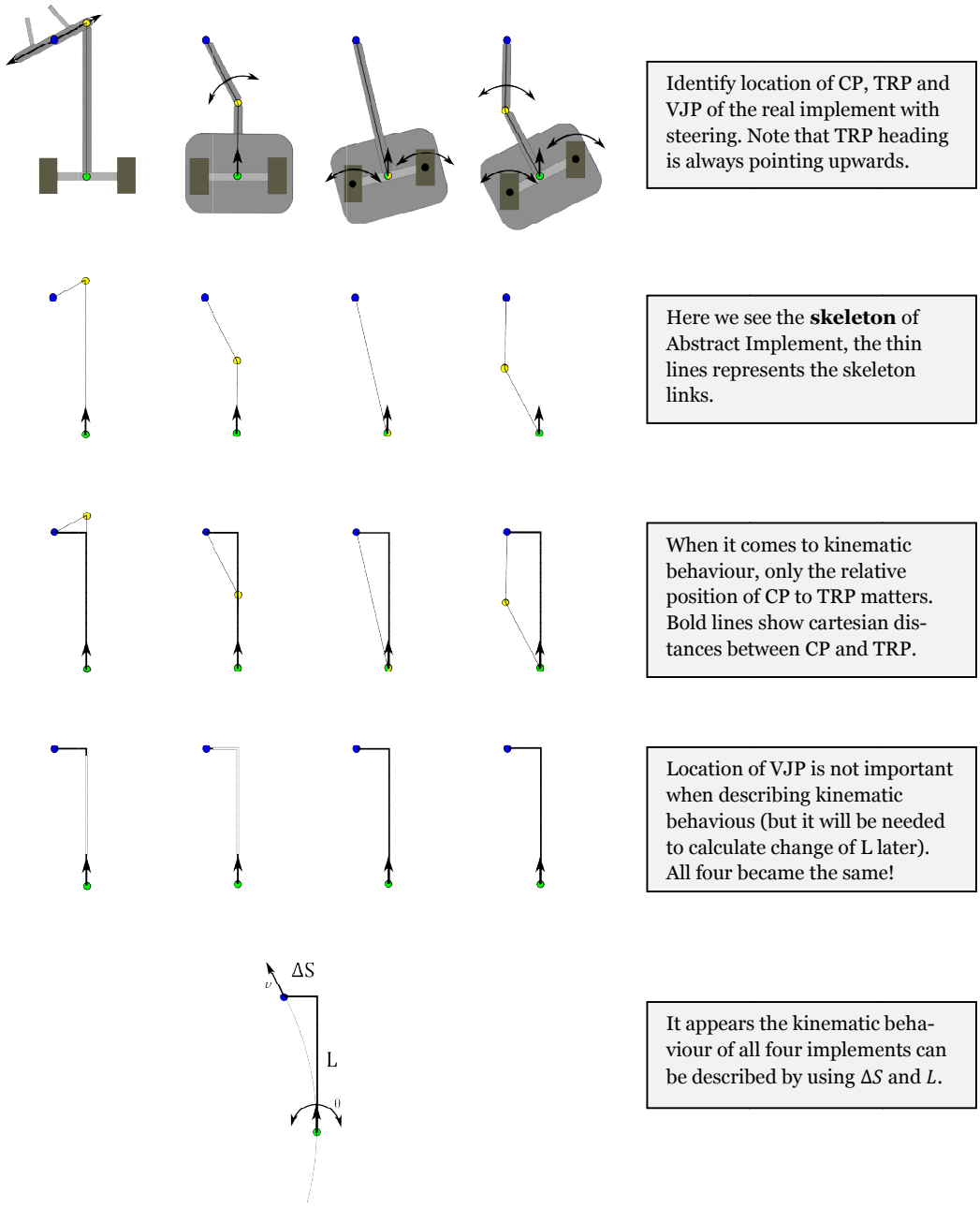


Figure 3. Abstraction step by step; it appears that in this case all four implements have the same kinematic behaviour; in the end only relative position of TRP and CP matters. Note that implement wheels are directed upwards in the situations. Note also that in the third example VJP=TRP, the points are coindident.

The beauty of this proposal is that the same kinematic model (differential equations) can be used for all implements, just the parameters change. The kinematic model can be derived in combined form (tractor + abstract implement together, the results is four states); or in separated form where the tractor has its own model and abstract implement another model. The kinematic equations for the combined model are presented in Appendix E (pure kinematic equations) and Appendix F (kinematic equations + actuator dynamics).

2.2 Mapping real implements to abstract

The **real implement** moves in 2D world in three (3) Degrees of Freedom (DOF): x,y and heading. However, an implement is always connected to the tractor (by using CP) so the number of controllable Degrees of Freedom is only two (2). Therefore, in rigid body implement, maximum two (2) Degrees of Freedom are available for control, even if the implement contained more actuators. Later, these two are known as Essential DOF and they are: **side offset** (ΔS) and **heading offset** ($\Delta\theta$). By using these two, ***any*** *real implement* can be modeled by using *Abstract Implement* (the skeleton).

Figure 4 defines some typical real implements. The proposed interface supports all these constructions and the standard will provide equations required for IMPLEMENT controller. The proposal supports for instance: hitch mounted implements with side shift (type A), trailed implements with drawbar steering (type C), trailed implements with steered wheels (type D), potato harvester (type E), and implements with drawbar steering with tool in drawbar (type F).

Note: the Abstract Implement is not limited to these examples; these are just selected samples that represent the variability of the real implements.

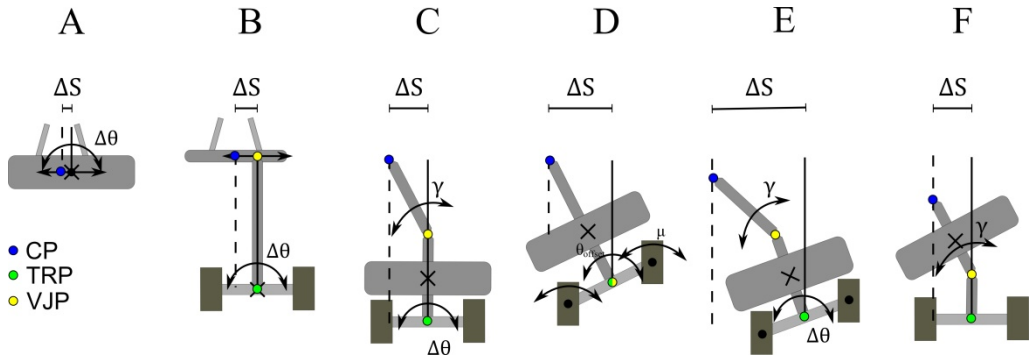


Figure 4. Typical implement types. The diagrams present how the real implements are mapped to the skeleton of the abstract implement.

CP – Connection Point: towing point; speed and direction are free.
TRP – True Rotation Point: ground contact; only forward motion (respect to implement frame) and rotate.
VJP – Virtual Joint Position; describes implement mechanical structure
ΔS is lateral movement of CP respect to TRP (In Figure 3 the horizontal distance)
$\Delta\theta$ is rotation of implement tool with respect to vertical axis (pointing upwards)

2.3 Conversion math

Abstract Implement *does not lose information* in case the number of mechanical degrees of freedom (e.g. cylinders) is maximum two (2).

In case the implement contains *more* Degrees of Freedom (3 or more), the implement manufacturer has to decide how to reduce the number of DOF to maximum two, by introducing additional constraints in the implement con-

troller. For instance, if the drawbar of towed implement contained three (3) rotational joints controlled by three hydraulic cylinders, the implement manufacturer would have to find a way to control three joints in order to realize side offset (ΔS) and heading offset ($\Delta\theta$), by reducing the number of DOF from 3 to 2. However, commercial implements with more than two actuators for implement steering are not found on the market, so it has not been possible to show sample equations how this could be done.

Figure 4 presents typical implement types. The Abstract implement is not limited for these, but in order to help IMPLEMENT manufacturers, below the required conversions on the implement side are listed. These conversion formulas are used between the joint coordinates of real actuators and variable of the Abstract Implement:

A) The ΔS is directly the lateral movement of linear actuator that moves the implement sideways
   $\Delta\theta$ is directly the angular movement of rotary actuator (optional)
B) The ΔS is directly the lateral movement of linear actuator that moves VJP sideways
   $\Delta\theta$ is directly the angular movement of rotary actuator (optional)
C) The angle of controlled joint (the real rotational joint = VJP)
   $\gamma = \sin^{-1}\left(\frac{\Delta S}{|CP-VJP|}\right)$
   $\Delta\theta$ is directly the angular movement of rotary actuator (optional)
D) The angle of implement steering wheels:
   $\mu = \sin^{-1}\left(\frac{\Delta S}{|CP-VJP|}\right)$
   $\Delta\theta$ is directly the angular movement of rotary actuator (optional)
E) See Appendix C.
F) The angle of controlled joint (the real rotational joint = VJP)
   $\gamma = \sin^{-1}\left(\frac{\Delta S}{|CP-VJP|}\right)$
   $\Delta\theta$ is directly the angular movement of rotary actuator (optional)

If you are an IMPLEMENT manufacturer, and you don't find your implement type in the samples, see Appendix A for instructions how to derive the necessary equations yourself.

3. Requirements

3.1 For IMPLEMENT

The first word for the engineer of IMPLEMENT controller: Your task is to *abstract* your machine to the general framework or to the skeleton of Abstract implement; you don't have to think about guidance problem, or Guidance Objectives.

The implement manufacturer may decide the mechanical structure of the implement steering without constraints to any specific kinematic structure. There is no requirement how the actuating system is realized. It can be real-

ized by using an electric or hydraulic actuator using either tractor's or implement's valves or by other means.

However, minimum requirements are:

- ECU with CAN-BUS interface (PLC, Microcontroller etc.) + minimum ISO 11783 stack
- Actuator for 1 degree of freedom: ΔS or $\Delta\theta$
- or actuators for 2 degrees of freedom to control both ΔS and $\Delta\theta$
- Position sensor(s) for corresponding actuator(s)
- Conversion between mechanical joint angles / actuator positions to essential DOF's
- PID (or similar) controller for controlling joint angles or positions

See the sketched guideline in Appendix A.

3.2 For GUIDANCE

The first word for the engineer of GUIDANCE controller: forget the *real* implement kinematics or mechanical structure or real actuators; change your mind to *Abstract Implement* and make your math only against that. It was up to the IMPLEMENT manufacturer to do the conversion between Abstract implement and the real implement, in the implement controller connected to ISOBUS. It might be possible to convert Abstract Implement variables back to the actuator positions of an implement, but in long term this is not helping you to reach ISOBUS plug-and-play.

GUIDANCE controller needs to support various Guidance Objectives that depend on operation. The guidance control law depends on Guidance Objectives and the proper one is selected based on operation. In each Guidance Objective case, the implement is described by Abstract implement, so for instance the prediction of movement is just the same independent of Guidance Objective. In this proposal the Guidance Objective is *not hard-coded* into the implement controller.

In each Guidance Objective very simple, simple, advanced or very advanced algorithms (see above for definitions) may be used. This interface does not hard-code anything, so it is open to very simple guidance control laws, like feedforward-curvature in "Follow Trails of Tractor" Guidance Objective (if you like that approach, see Appendix D for help). On the other hand, the interface does not lose any information (in case of <3 actuators), so more advanced control algorithms utilizing the kinematic model can be used without any conversion.

The interface is selected in a way that kinematic model of trailer type implement is straight-forward to design. You may combine kinematic models of the tractor and the Abstract Implement, by taking into account the dynamics of actuators and design a parameterized control law. Control law gains can be automatically tuned e.g. by using gain scheduling based on Implement Guidance Parameter Pool (IGPP).

On the other hand, you can still use simple control laws (like PID), but utilize the kinematic model of Abstract Implement to **predict** the movement.

A sample of kinematic equations for Abstract Implement is presented in Appendix E. The extended version with actuation dynamics is presented in Appendix F. These are presented as a proof that the kinematic equations exist.

4. The extension of ISO 11783

4.1 Messaging for control

A new PGN is requested for Implement Guidance. It can be tailored for PDU1 or PDU2.

B.26.3 Implement guidance message

Transmission repetition rate:	100 ms
Data length:	8 byte
Data page:	0
PDU format:	??
PDU specific:	DA
Default priority:	3
Parameter group number:	??
Byte 1:	Subcommand, see Table 1.
Byte 2-8:	Table 1.

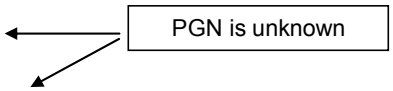


Table 1. Subcommands

Subcommand	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1 = Guidance-to-Implement RT control	implement number	ΔS setpoint		$\Delta\theta$ setpoint		Reserved	Command state (bits, like B.26.1 byte 3)
2 = Implement-to-Guidance RT estimated	implement number	ΔS estimated		$\Delta\theta$ estimated		Reserved	Readiness state (bits, like B.26.2 byte 3)
3 = Implement-to-Guidance local sensor measurement	implement number	See Appendix N.					
4 = request IGPP	implement number						
5 = reply to request IGPP	implement number	IGPP transmitted using ISO TP, the pool contains fixed number of parameters in order, foreseen length around 50 bytes					

Table 2. Gains and offsets

Variable	Gain	Offset	Range	Special
ΔS	1mm/bit	-32.768m	-32.768 - 32.766m	FFFFh=N/A
$\Delta\theta$	0.01deg/bit	-90deg	-90 - 90 deg	FFFFh=N/A
implement number	1=the first implement, 2=the second implement connected to the first one etc.			254=not set

Note: The rotation of the implement tool can be packed in less than 16 bits but 8 bits is too rough; so it would be 10-12 bits; so up to 4 bits can be released for another purpose.

Note: *Implement number* is defining the order in chain. The correct number for the implement is shall be settable by using user interface of the implement, typically by Virtual Terminal.

Messages for the online guidance sensors are drafted in Appendix G.

4.2 Protocol for the parameters

In ISO 11783, two protocols are available to communicate parameters. In Task Controller (ISO 11783-10), the parameters are exchanged between TC and TC Client by using the process data procotol. The protocol is tailored to the application between these two parties and therefore it is hard to reuse for other purposes due to embedded design.

The other option that ISO 11783-6 provides is to use the so called object pool, to pack (serialize) the objects into a byte array. However, that messaging is also coupled with Virtual Terminal application, and therefore hard to reuse for other purposes.

Therefore, to follow the same pattern as ISO 11783 series have been designed, is to set up a *dedicated object pool* for implement guidance. The pool may be transmitted either by using similar messaging pattern as used in ISO 11783-6 or alternatively take the packet model from NMEA2000 standard (which is defined for navigation applications, by the way).

We define the dedicated object pool as **Implement Guidance Parameter Pool** and the abbreviation **IGPP** will be used. The pool contains both geometry parameters of skeleton (the location of CP, VJP etc...) and dynamic system parameters (like lag and delay).

4.3 Implement Guidance Parameter Pool (IGPP)

We see two options to define IGPP. One option is to follow the pattern of NMEA2000 where the parameters are a defined list that contains a linear presentation of required parameters. For the parameters not used in a particular implement, a means to define Not-available should be included, e.g. 0xFFFF like defined in ISO 11783. The other option is to form an extensible tree of parameter objects that refer to the parent e.g. which would define the base for the coordinate system. This would follow the concept of ISO 11783-6 Virtual terminal object pool.

First of all, the Abstract Implement contains three coordinate systems (CS): The base coordinate system is TRP, and then comes VJP and the third one is Tool Rotation Point (TORP). In some cases the body of implement rotates with the tool (Implement Type E), and this is defined by using parameter VJP_ROT.

Relation of CS is that VJP is always in TRP, CP is always in VJP and GCP is always in TORP. However, the TORP may be located either in TRP (implement types A-E) or in VJP (implement type F) – the appropriate CS is defined with TORP_ORIGIN parameter.

The dynamics of implement steering actuators is approximated by using 1st order dynamics + delay; or in other words lag (time constant of 1st order) + delay. In control engineering terms the transfer function is form:

$$G(s) = \frac{1}{LAG \cdot s + 1} e^{-DELAY \cdot s}$$

The list type of IGPP contains the following parameters:

Name	Abbreviation	Unit
VJP location X in TRP VJP location Y in TRP	VJP_X VJP_Y	m
Is VJP rotating with heading offset	VJP_ROT	yes/no
CP location X in VJP CS CP location Y in VJP CS	CP_X CP_Y	m
Tool Rotation Point X in VJP/TRP Tool Rotation Point Y in VJP/TRP	TORP_X TORP_Y	m
Is TORP in VJP coordinate system (..or TRP)	TORP_ORIGIN	yes/no
GCP location X in TORP GCP location Y in TORP	GCP_X GCP_Y	m
Lag of ΔS (1st order dynamics)	LAG_DS	s
Lag of $\Delta\theta$ (1st order dynamics)	LAG_DT	s
Delay of ΔS	DELAY_DS	s
Delay of $\Delta\theta$	DELAY_DT	s
Control space constraint point 1 ΔS Control space constraint point 1 $\Delta\theta$	CSC1_DS CSC1_DT	m deg
Control space constraint point 2 ΔS Control space constraint point 2 $\Delta\theta$	CSC2_DS CSC2_DT	m deg
Control space constraint point 3 ΔS Control space constraint point 3 $\Delta\theta$	CSC3_DS CSC3_DT	m deg
Control space constraint point 4 ΔS Control space constraint point 4 $\Delta\theta$	CSC4_DS CSC4_DT	m deg

The other option, to use tree type pool would contain the same parameters, in hierarchical presentation. Selection which pattern to follow is so much related to the general requirements of architecture design of ISO 11783 series that we leave the choice open for the options.

The maximum control space needs to be defined so that GUIDANCE may take that into account in control laws to give valid commands that are can be reached with the actuators available in the implement. The control space is necessary in simple control laws, like PID controller, to avoid integrator windup and in advanced control methods it is used as a constraint. In some implement types the control space is *rectangular* (Implement Types A-C) and some other form of parallelogram (Implement Type E), so IGPP needs to support both of these. Therefore, up to four points are defined (CSC1-CSC4) to define the corner points; these can be used to define rectangular or parallelogram control space.

5. Summary

The idea of the proposal is to model the Essential Degrees of Freedom of the implement steering capabilities. The parameters are the same regardless of the mechanical structure of the implement – there is no need for the separate parameters of linear and joint actuators, the implement ECU does the conversion in both directions without losing information.

The *beauty* of this proposal is that it doesn't require fancy controllers either on IMPLEMENT side or on the GUIDANCE system. Both simple and advanced solutions are supported. ΔS is the way to make the implement mechanical steering system abstract, or to generalize implement to the similar format which we call the skeleton of Abstract implement. The abstract implement tries to be as simple as possible, but still support all kinds of configurations.

Abstract Implement interface supports not only trailer type implements but also *hitch mounted* implements with the side shift actuator and/or rotation of Implement Tool.

GUIDANCE manufacturer does not need to have full kinematic model of the implement mechanics. GUIDANCE manufacturer may fix the system based on the abstracted implement, with those dimensions (CP, TRP, VJP) and Essential DOF's ($\Delta S, \Delta\theta$). GUIDANCE may use the fixed model of Implement in the control system and adapt the parameters based on dynamic parameters received from the Implement during the first plug-in – for instance by using gain scheduling pattern.

As there are only two signals (the essential DOF's) to be exchanged in real time control between GUIDANCE and IMPLEMENT (any kind of), it is possible to fit these signals in a single CAN frame (Chapter 4), which is increasing the bus load only by 1% (any kind of implements).

Last but not least, in this proposal the true requirement of **ISOBUS plug-and-play** requirement is considered. By using this separation between the tractor, implement and guidance; it is possible both in theory and in practice that all three partners can be engineered against the standard, without a need that engineer of the guidance controller has to program mathematics for each implement separately and visit every farm in the world to do this.

Appendix A:
User guideline for IMPLEMENT manufacturer

(this is a sketch of an guideline)

This guideline explains how to proceed in order to program support for ISOBUS IMPLEMENT GUIDANCE interface. The assumption is that you have already existing mechanical implement that has capability to do steering action – in case you design a completely new implement mechanics, you need to apply the same rules.

Option 1 (professional)

Step 1: Identify the steering actuators. How many actuators, where they are located. Identify location of TRP (typically between wheels)
Step 2: Draw your implement in x-y coordinate system so that TRP is located in origin and wheels are pointing upwards; along direction of y-axis.
Step 3: Draw VJP and CP. Draw actuators and joints.
Step 4: ΔS is defined as a distance from y-axis to CP horizontally.
Step 5: $\Delta\theta$ is defined as an angle between Implement Tool and y-axis.
Step 6: Derive equations that connect actuators/joints and $(\Delta S, \Delta\theta)$.

Option 2 (easy)

Or... If you are lazy and your implement is one of the common types (See Figure 4 for Type A-F), identify which is your type and then proceed to Chapter 2.3. to collect your formulas. No need to derive any math. The math for common implement types will be provided in the standard.

To understand better either option, see Appendix B.

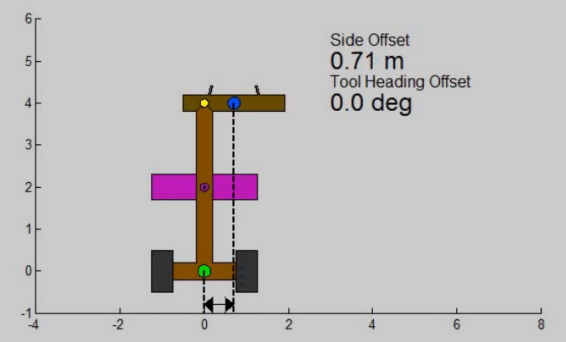
Appendix B: Case Implement Type B-D

See the animations for correct interpretation how to derive equations. The animations also illustrate how to derive math for any implement: position TRP to origin, keep wheels always upwards position (along y-axis) and variate all your actuators to let CP (blue dot) and Implement Tool heading to change. The result is the Equations that connect your actuators and $(\Delta S, \Delta\theta)$.

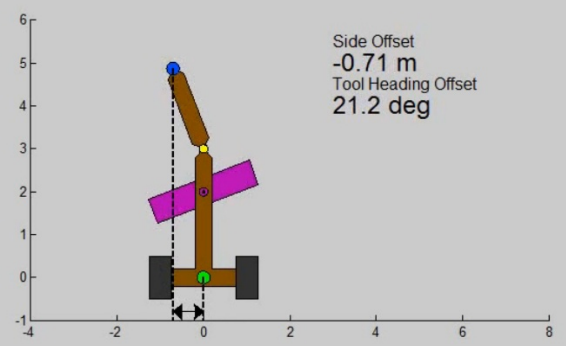
In all videos there are three parts:

1. part of video: $\Delta S = \sim,\ \Delta\theta = 0$ (only side offset)
2. part of video: $\Delta S = 0,\ \Delta\theta = \sim$ (only tool heading offset)
3. part of video: $\Delta S = \sim,\ \Delta\theta = \sim$ (both offsets)

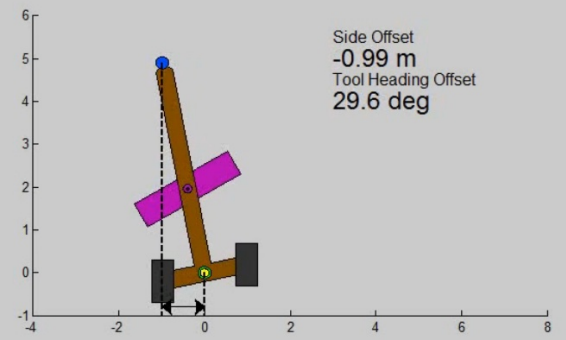
Type B: http://www.youtube.com/watch?v=mI8EqswRzZA



Type C: http://www.youtube.com/watch?v=4QyE8vQFivo



Type D: http://www.youtube.com/watch?v=ACNGph_Cd7E



Appendix C: Case Potato Harvester (Type E)

The required equation to be written to implement ECU is:

$$\alpha_{wheels} = -\Delta\theta$$
$$\alpha_{drawbar} = -\Delta\theta + \sin^{-1}\frac{\left(\Delta S - \Delta_{VJP}\cos\Delta\theta - \|TRP - VJP\|\sin\Delta\theta\right)}{\|VJP - CP\|}$$

and to the other way

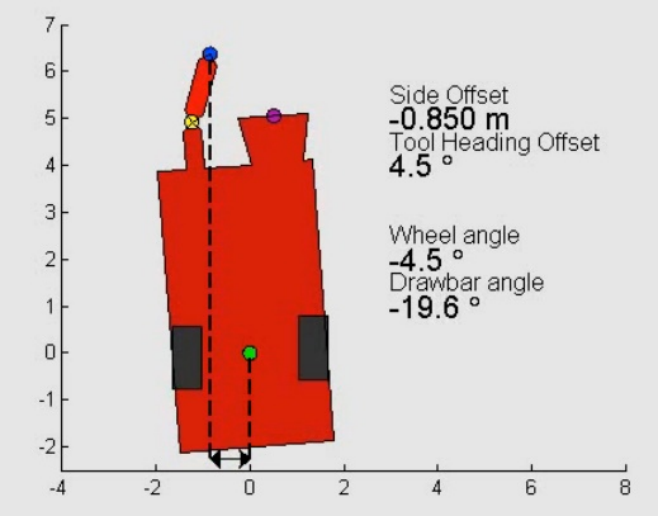
$$\Delta\theta = -\alpha_{wheels}$$
$$\Delta S = \|TRP - VJP\|\cos\Delta\theta + \|VJP - CP\|\cos(\alpha_{drawbar} + \Delta\theta) - \Delta_{VJP}\sin\Delta\theta$$

These can be calculated with an embedded processor every 100ms, without floating point unit, with required accuracy.

Geometry of Grimme SE 260 harvester was used to verify equations. So this is a real machine. Note that the drawbar joint is located Δ_{VJP}=0.85m offset from the center line, not in the center like in all previous examples we have seen in various proposals. The other numbers are |TRP-VJP|=5m, |VJP-CP|=1.5m.

See the animation: www.youtube.com/watch?v=_ZqUZmI9Ep8

1. part of video: $\Delta S = \sim,\ \Delta\theta = 0$ (only side offset)
2. part of video: $\Delta S = 0,\ \Delta\theta = \sim$ (only tool heading offset)
3. part of video: $\Delta S = \sim,\ \Delta\theta = \sim$ (both offsets)



So these equations are required to be programmed to ECU of Potato Harvester (and inverse functions which are more simple). Nothing more has to be done to the implement side. The “follow the trails of tractor” or other Guidance Objectives are programmed in GUIDANCE controller and implement manufacturer does not have to worry on those.

Appendix D: Guidance Objective – Follow Trails of Tractor in Feedforward way

In this Guidance Objective the trailer wheels must follow the trails of tractor wheels. Orientation does not matter. With feedforward manner, the simplest way to do this is to approximate the equations, by assuming hitch length=0.

In "Trailer follow the trails of tractor" case a simple feed-forward rule can be used to approximate movements:

$$\Delta S = \frac{1}{k_{trac}}\left(1 - \sqrt{1 - {k_{trac}}^2 L^2}\right)$$

The Conversion to the other direction is:

$$k_{trac} = \frac{2 \cdot \Delta S}{L^2 + \Delta S^2}$$

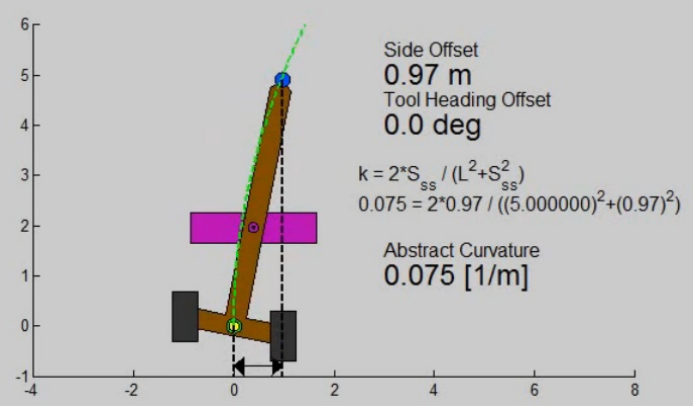
However, the equation above assumes the hitch length=0, so it is recommend to use a bit more effort in GUIDANCE controller to do it better. Let the general function for Curvature-Feedforward be

$$\Delta S = f_{curv}(k_{trac})$$

See the animation how the curvature projects over the implement:
http://www.youtube.com/watch?v=Emiwf5eR-TU

The green dashed line shows the curve that has curvature k_{trac} equal to the tractor. You can see that green dashed line goes through CP (blue dot), so the conversion from curvature to side offset (ΔS) works ok.

The animation shows **Abstract Curvature** of implement instead of Curvature, because using word "Curvature" misleads to mix with the Curvature of Tractor, which is *real curvature*; in case of implement the meaning is *purely abstract*.



Furthermore; the Guidance Objective: ”Follow Trails of Tractor” is presented combined simulation of the tractor and implement. Note: *These animations do not have any feedback control enabled.*

Animation number 1: Follow curve with Curvature-Feedforward:

Here feedforward is applied to follow curvature of tractor, formula $\Delta S = f_{curv}(k_{trac})$

http://www.youtube.com/watch?v=sWPJE4qkQUk

Animation number 2: Follow straight and make SideOffset-Feedforward:

Here the tractor drives straight, and ΔS is used to realize static offset for “follow trails” Objective.

http://www.youtube.com/watch?v=nPaRShsHWYs

Animation number 3: Follow curve with Curvature-Feedforward + adjust SideOffset

In this animation, the trailer side-offset is used to do curvature following and side-offset is adjusted simultaneously to +0.5m. The formula $\Delta S = f_{curv}(k_{trac}) + \Delta S_{wanted}$.

http://www.youtube.com/watch?v=OCihoqmWSYo

Conclusion: This Abstract Implement interface can be used with very easy equation to do the Guidance Objective ”Follow Trails of Tractor”, but thanks to its versatility, it can be used for **any** Guidance Objectives.

Appendix E: Kinematic model of Abstract Implement

The GUIDANCE manufacturer may freely select the approach to be used to steer the implement. In a simple approach, the system may be based on feed-forward law (e.g. Appendix D) together with feedback law.

More advanded control algorithms may need kinematic equations (differential equations) that describe the behaviour of the implement in 2D space. The kinematic equations can be derived in many ways, for instance the tractor and the abstract implement may be two different models; or a single combined model may describe both connected together.

Here is presented a combined model for a front wheel steered tractor. The symbols are:

(x_{trac}, y_{trac}): location of tractor rear axle in global coordinate system
ϕ: heading of tractor in global coordinate system
ψ: heading of TRP (of Abstract Implement) in global coordinate system
v: velocity of tractor
k: curvature of tractor
ΔS: side offset
$\Delta\theta$: tool heading offset
b: distance from tractor rear axle to connection point (CP), hitch length
c_{VJP}: binary constant in IGPP, defines whether VJP rotates around TRP with $\Delta\theta$
(example implement types: 0 for types A-D and F; 1 for type E)

The kinematic model is:

$$\begin{cases} \dot{x}_{trac} = v\ \cos\phi \\ \dot{y}_{trac} = v\ \sin\phi \\ \dot{\phi} = k\ v \\ \dot{\psi} = \dfrac{-b\ k\ v\cos(\phi-\psi) - v\sin(\phi-\psi) + c_{VJP}\dfrac{d}{dt}L_2\sin\Delta\theta + \dfrac{d}{dt}\Delta S}{(L_1+L_2)\cos\left(c_{VJP}\ \Delta\theta\right) - \Delta_{VJP}\sin\left(c_{VJP}\ \Delta\theta\right)} - c_{VJP}\dfrac{d}{dt}\Delta\theta \\ where\ L_1 = \|TRP - VJP\|\ and\ L_2 = \sqrt{\|VJP - CP\|^2 - \Delta S^2} \end{cases}$$

Note: This is not unique form, other equations may provide the same result.

Appendix F: Kinematic model and dynamics of actuation

To design GUIDANCE controller even more accurately than only kinematics, the dynamics of actuation need to be taken into account. Here is the kinematic model derived in Appendix E is supplemented with actuation dynamics.

The foreseen model to describe the dynamics of actuation for both side offset and heading offset is based on assumption of *first-order-lag plus time delay (FOLPD)*. This incorporates two constants: time constant of first order dynamics and the time delay, the unit of both is a *second (s)*.

A combined model for front wheel steered tractor. The symbols are:

(x_{trac}, y_{trac}): location of tractor rear axle in global coordinate system
ϕ: heading of tractor in global coordinate system
ψ: heading of TRP (of Abstract Implement) in global coordinate system
v: velocity of tractor
b: distance from tractor rear axle to connection point (CP), hitch length
c_{VJP}: binary constant in IGPP, defines whether VJP rotates around TRP with $\Delta\theta$
(example implement types: 0 for types A-D and F; 1 for type E)
k_{sp}: curvature of tractor (setpoint)
k_{ms}: curvature of tractor (actual)
ΔS_{sp}: side offset (setpoint)
ΔS_{ms}: side offset (actual)
$\Delta\theta_{sp}$: tool heading offset (setpoint)
$\Delta\theta_{ms}$: tool heading offset (actual)
τ_1: time constant of side offset dynamics
τ_2: time constant of tool heading offset dynamics
delay_1: transport delay of side offset dynamics
delay_2: transport delay of tool heading offset dynamics

Thus, the kinematic model is:

$$\begin{cases} \dot{x}_{trac} = v\ \cos\phi \\ \dot{y}_{trac} = v\ \sin\phi \\ \dot{\phi} = k\ v \\ \dot{\psi} = \dfrac{-b\ k\ v\cos(\phi-\psi) - v\sin(\phi-\psi) + c_{VJP}\dfrac{d}{dt}L_2\sin\Delta\theta_{ms} + \dfrac{d}{dt}\Delta S_{ms}}{(L_1+L_2)\cos\left(c_{VJP}\,\Delta\theta_{ms}\right) - \Delta_{VJP}\sin\left(c_{VJP}\,\Delta\theta_{ms}\right)} - c_{VJP}\dfrac{d}{dt}\Delta\theta_{ms} \\ where\ L_1 = \|TRP - VJP\|\ and\ L_2 = \sqrt{\|VJP - CP\|^2 - {\Delta S_{ms}}^2} \end{cases}$$

The inputs to the system are $\{\Delta S_{sp}, \Delta\theta_{sp}\}$ (the command values going on ISOBUS). The outputs are $\{\Delta S_{ms}, \Delta\theta_{ms}\}$ (the actual values going on ISOBUS).

$$\Delta S_{ms}(t) = g_1\big(\Delta S_{sp}(t), t\big) \rightarrow G_1(s) = \frac{1}{\tau_1 s + 1} e^{-\text{delay}_1 \cdot s}$$
$$\Delta \theta_{ms}(t) = g_2\big(\Delta \theta_{sp}(t), t\big) \rightarrow G_2(s) = \frac{1}{\tau_2 s + 1} e^{-\text{delay}_2 \cdot s}$$

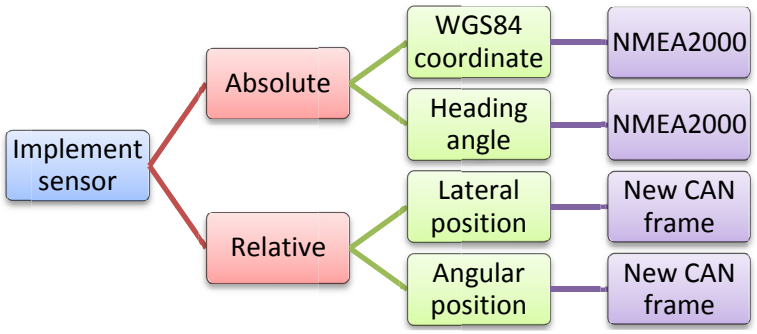
If this model is used for simulation purposes, the inputs are $\{k, v, \Delta S_{sp}, \Delta \theta_{sp}\}$ and the outputs are $\{x_{trac}, y_{trac}, \phi, \psi\}$. For the tractor dynamics, the standard does not provide any parameters, but the same FOLPD dynamics approximation could be used for $\{k, v\}$ too.

Appendix G: Sensor messages

Implement side sensors can be classified into absolute and relative.

Absolute sensors measure the position and/or orientation regarding global coordinate system (like position in GPS coordinates or heading in compass angle). The absolute compass angle can be produced either by using double GNSS receivers or other system, this standard does not say which technology to be used. For both, NMEA2000 standard provides CAN frame definitions and as it has been utilized in another parts of ISO 11783 standard series, it is natural to use messages also here.

Relative sensors measure position and/or orientation regarding landmarks in the field. Examples are: the row feeler of potato harvester measuring lateral distance to ridge, contactless sensor measuring the location of windrow in grass field or vision system measuring **both** lateral position and orientation in row crop field. Sending relative lateral position (m) and relative angular position (deg) in CAN frame, a new message needs to be defined.



Assuming GNSS controlled tractor-implement system; two NMEA2000 receivers would co-exist in the ISO 11783 network. In order to define which NMEA2000 device (control function) is located in the implement and which one in the tractor, some parameter information must be provided to GUIDANCE. The authors of this document propose that NAME of GNSS receiver (fixed to implement) shall be included in IGPP (Implement Guidance Parameter Pool) and additional supported NMEA2000 frames are listed as options. Plus the location in the implement frame.

Relative sensor information needs a new CAN frame definition. Proposed subcommand for message defined in Chapter 4 is added, see Table G.1 & Table G.2. Sensor may contain lateral position measurement or relative orientation measurement, or both. The location of sensor in the implement frame must be included in IGPP. Multiple sensors may exist in one implement.

Table G.1. Relative measurement message

Subcommand	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
3 = Implement-to-Guidance local sensor measurement	implement number | sensor number<<4	relative lateral position measured (FFFF=N/A)		relative heading measured (FFFF=N/A)		Reserved	Reserved

Table G.2. Gains and offsets

Variable	**Gain**	**Offset**	**Range**	**Special**
relative lateral position measured	1mm/bit	-32.768m	-32.768 - 32.766m	$FFFF_h$=N/A
relative heading measured	0.01deg/bit	-90deg	-90 - 90 deg	$FFFF_h$=N/A

Appendix H: Aftermarket products for Implements

It is assumed that most of (AEF) ISOBUS implements supporting this Implement Guidance functionality also contain some other ISOBUS functionalities like UT. The other assumption is that the steering capability has to be provided as an integrated part of ISO 11783 Working Set, not as a third-party add-on. However, as this proposal is rather independent of any other ISOBUS functionalities (like TC, AUX-*, UT), it is possible to use this interface also for aftermarket add-on products, but still AEF certification has to be done for ECU put in the implement and the same rules apply as for OEM. In case of aftermarket ECU's, the manufacturer should design the ECU in a way that several implement types are all preprogrammed and the service people installing the add-on can select from the parameters the correct type and set the correct parameters, by using some maintenance tool.

Aalto-ST 2/2015

ISBN 978-952-60-6165-8 (pdf)
ISSN-L 1799-4896
ISSN 1799-4896 (printed)
ISSN 1799-490X (pdf)

Aalto University
School of Electrical Engineering
Department of Electrical Engineering and Automation
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS