

Navigation System for Modular Agricultural Machines using Optimal Control Methods and Industrial Standard Network

Juha Backman

Navigation System for Modular Agricultural Machines using Optimal Control Methods and Industrial Standard Network

Juha Backman

Doctoral dissertation completed for the degree of Doctor of Science in Technology to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall AS1 of the school on 22th November 2013 at 12:00.

**Aalto University
School of Electrical Engineering
Department of Automation and Systems Technology
Autonomous Systems**

Supervising professor

Prof. Arto Visala

Thesis advisor

Dr. Timo Oksanen

Preliminary examiners

Prof. Josse de Baerdemaeker, KU Leuven, Belgium

Asst. Prof. Stavros Vougioukas, UC Davis, United States

Opponent

Prof. Juha Rönning, University of Oulu, Finland

Aalto University publication series

DOCTORAL DISSERTATIONS 169/2013

© Juha Backman

ISBN 978-952-60-5392-9

ISBN 978-952-60-5391-2 (pdf)

ISSN-L 1799-4934

ISSN 1799-4934 (printed)

ISSN 1799-4942 (pdf)

<http://urn.fi/URN:ISBN:978-952-60-5391-2>

<http://lib.tkk.fi/Diss/>

Unigrafia Oy

Helsinki 2013

Finland



Author

Juha Backman

Name of the doctoral dissertation

Navigation System for Modular Agricultural Machines using Optimal Control Methods and Industrial Standard Network

Publisher School of Electrical Engineering

Unit Department of Automation and Systems Technology

Series Aalto University publication series DOCTORAL DISSERTATIONS 169/2013

Field of research Autonomous Systems

Manuscript submitted 31 May 2013

Date of the defence 22 November 2013

Permission to publish granted (date) 4 October 2013

Language English

Monograph

Article dissertation (summary + original articles)

Abstract

In agriculture, there is a growing demand to increase the efficiency of machinery. This has led to increasingly larger machines and an increasing usage of automation. The challenge is that there are numerous different combinations of tractors and implements and they all have to work together. The ISO 11783 standard is a widely accepted standard for realising an information exchange between the tractors and implements.

The motivation for this thesis was to reduce the workload of the driver and also to improve the efficiency and accuracy of field work. This was achieved by removing the human being from behind steering wheel and letting the navigation system do the driving. Because of their size, implements are more commonly towed these days. The navigation system should also take into account the towed implement. In most commercial products, only the tractor is considered.

Two specific goals were set: 1) to study a combined navigation system that is able to drive at a speed of at least 12 km/h with less than 10 cm lateral error under real field conditions; 2) to discuss and present the ways in which a decentralised and generic combined navigation system can be realised using the ISO 11783 network.

The solution was to use Nonlinear Model Predictive Control (NMPC) in path tracking. With NMPC, a controller can be realised that controls the motion of both the tractor and the trailer in an optimal manner. In NMPC, the reference trajectory is usually coupled with time. However, in path tracking for agricultural navigation, it is more suitable not to couple the positions of the tractor and the implement with time. Hence, the NMPC had to be modified. Also, other difficulties that arise from the usage of NMPC had to be overcome: the NMPC needs an accurate estimate of the state, so an optimal state estimator that merges the local and global position measurement was developed. Semi-automatic tuning methods were developed for tuning and configuring the numerous parameters needed for combined navigation. The reference path should be feasible, so a feasible path planning algorithm was developed. Additionally, a collision avoidance method was developed and incorporated within the NMPC.

The results of the field tests show that the goals were met. The accuracy of the navigation system is within the given limits. The system is also able to complete the field work without human intervention, including making headland turnings and avoiding electricity poles. The required information on combined navigation has been illustrated based on the case studies. It is possible to build system based on the ISO 11783 standard and distribute sensors and actuators.

Keywords implement guidance, path tracking, nonlinear model predictive control, collision avoidance, software architecture, simulation, tuning, state estimation

ISBN (printed) 978-952-60-5392-9

ISBN (pdf) 978-952-60-5391-2

ISSN-L 1799-4934

ISSN (printed) 1799-4934

ISSN (pdf) 1799-4942

Location of publisher Espoo

Location of printing Helsinki

Year 2013

Pages 227

urn <http://urn.fi/URN:ISBN:978-952-60-5391-2>

Tekijä

Juha Backman

Väitöskirjan nimi

Navigointijärjestelmä modulaarisille maataloustyökoneille käyttäen optimisäädön menetelmiä ja teollisuuden standardiväyliä

Julkaisija Sähkötekniikan korkeakoulu**Yksikkö** Automaatio- ja systeemitekniikan laitos**Sarja** Aalto University publication series DOCTORAL DISSERTATIONS 169/2013**Tutkimusala** Autonomiset järjestelmät**Käsikirjoituksen pvm** 31.05.2013**Väitöspäivä** 22.11.2013**Julkaisuluvan myöntämispäivä** 04.10.2013**Kieli** Englanti **Monografia** **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)****Tiivistelmä**

Maataloudessa on edelleen kasvava tarve tehostaa työkoneiden käyttöä. Kehitys on johtanut yhä suurempiin työkoneisiin ja enenevässä määrin automaation käyttöön. Haasteena on, että sekä traktorien että työkoneiden valmistajia on suuri määrä. Koneiden yhteensopivuutta ratkaisemaan on kehitetty ja yleisesti hyväksytty ISO 11783 standardi, joka määrittelee tiedonsiirron traktorin ja työkoneen välillä.

Työn motivaationa oli vähentää kuljettajan työmäärää sekä tehostaa ja tarkentaa peltotyötä. Tämä saavutettiin korvaamalla ihmisen ohjaus automaattisella navigointijärjestelmällä. Koska työkoneiden koko kasvaa, ne ovat yhä useammin hinattavia. Nykyiset kaupalliset navigointijärjestelmät eivät ota hinattavia työkoneita huomioon.

Vaatimusten saavuttamiseksi asetettiin kaksi tavoitetta: 1) tutkia ja kehittää navigointijärjestelmää, joka kykenee alle 10 cm työkoneen sivuttaisvirheeseen vähintään 12 km/h työnopeudella todellisissa pelto-olosuhteissa. 2) Pohtia ja esittää menetelmiä, joilla hajautettu ja yleinen yhdistetty navigointijärjestelmä voidaan toteuttaa käyttäen standardin ISO 11783 määrittelemää tiedonsiirtoa.

Ratkaisu oli käyttää epälineaarista malliprediktiivistä säädintä (NMPC) polun seurantaan. NMPC:tä käyttäen on mahdollista tehdä säädin, joka ohjaa sekä traktoria että työkoneita optimaalisella tavalla. NMPC:ssä referenssi on kuitenkin aikaperusteinen trajektorin ja polun seurannassa referenssi ei ole sidottu aikaan. Muutoksia säätimeen oli tehtävä. Myös muita ongelmia, jotka aiheutuivat NMPC:n käytöstä, ratkaistiin: NMPC tarvitsee systeemistä tarkan tilaestimaatin, joten luotiin optimitalaestimaattori yhdistämään paikalliset ja globaalit paikanmittaukset. Järjestelmässä on suuri määrä asetettavia parametreja, joten puoliautomaattinen viritysjärjestelmä kehitettiin. Referenssipolun tulisi olla toteutettavissa oleva, joten rajoitukset huomioiva polunsuunnittelualgoritmi kehitettiin. Myös NMPC:hen yhdistetty esteiden väistö kehitettiin.

Peltoestetit osoittivat että kaikki tavoitteet saavutettiin. Navigointijärjestelmän tarkkuus oli sallituissa rajoissa ja järjestelmä kykeni toteuttamaan kokonaisen peltotyön ilman ihmisen puuttumista ohjaukseen, mukaan lukien sähkötolppien väistön. Esimerkkitapauksiin perustuen tarvittavan siirrettävän informaation määrä yhdistetyssä navigoinnissa havainnollistettiin. Johtopäätöksenä voidaan todeta, että on mahdollista toteuttaa yhdistetty navigointijärjestelmä perustuen ISO 11783 standardiin ja hajauttaa mittaukset ja toimilaitteet väylälle.

Avainsanat työkoneen navigointi, polun seuranta, epälineaarinen malliprediktiivinen säädin, esteen väistö, ohjelmistoarkkitehtuuri, simulointi, virittäminen, tilan estimointi

ISBN (painettu) 978-952-60-5392-9**ISBN (pdf)** 978-952-60-5391-2**ISSN-L** 1799-4934**ISSN (painettu)** 1799-4934**ISSN (pdf)** 1799-4942**Julkaisupaikka** Espoo**Painopaikka** Helsinki**Vuosi** 2013**Sivumäärä** 227**urn** <http://urn.fi/URN:ISBN:978-952-60-5391-2>

Preface

First of all, I would like to thank to my wife, Anne, who has been beside me through all of this. Things haven't always gone as planned, but together we have overcome all the difficulties. I also want to thank to my son, Rasmus, who has brought a great deal of joy and happiness to us and who has also helped me get my mind out of the work.

To begin from the beginning, I have always been interested in how things work and in designing new things. It all began when I was just a boy. Legos were very interesting toys and, thanks to my parents, I had a large number of them. I think that my basic engineering skills came from those days playing with Legos. When I was a little older, my father gave me old broken electric devices so that I could disassemble and examine them. Shortly after that, I got a summer job at my father's electrical installation company.

I know already when I was young that my future job would be related to electricity or automation. But a doctoral thesis wasn't my original plan. Fortunately, I have always had excellent teachers. In particular, my mathematics and physics teachers were very inspiring and motivating. I have always set my goal a little farther than I could reach and that has taken me forward.

At the university, I ended up taking part in the field robot project, which aimed to participate in the international field robot event competition. Thanks to all those who were on the team and especially the instructors, we won that competition. With the help of this project, I got a master's thesis position with the Autonomous Systems research group. One thing led to another and I finally ended up as a postgraduate student in the same research group.

I would like to warmly thank everyone who has helped me during my postgraduate studies. Especially, I would like to thank to my instructor, Dr Timo Oksanen, for his invaluable guidance throughout my studies and also Senior Research Technician Raimo Linkolehto for his assistance in field tests and with test equipment. Without their help, this thesis wouldn't have been possible. I would like to also thank all the other members of our research group — Sami Alaiso, Maria Hakonen, Heikki Hyyti, Jouko

Kalmari, Teemu Kemppainen, Jakke Kulovesi, Antti Kunnas, Ville Matikainen, Mikko Miettinen, Mikko Vihlman and Matti Öhman — for providing an interesting and stimulating work environment and for all the encouraging and enlightening discussions.

Of course, my postgraduate studies and this research project wouldn't have been possible without my supervisor and the professor of the research group, Professor Arto Visala. I would like to thank to him for giving me an opportunity to pursue my postgraduate studies and also for his support during my studies.

I also appreciate the comments and suggestions for improvements made by the pre-examiners of this thesis, Professor Josse De Baerdemaeker and Assistant Professor Stavros Vougioukas.

Finally, this research was conducted partly in cooperation with the Energy and Life Cycle Cost Efficient Machines (EFFIMA) research programme, which is managed by the Finnish Metals and Engineering Competence Cluster (FIMECC), and funded primarily by the Finnish Funding Agency for Technology and Innovation (TEKES) and various research institutes and companies. Part of the research project was also funded by the Graduate School in Electronics, Telecommunications and Automation (GETA). GETA is a post-graduate programme offered jointly by five universities: Aalto University School of Electrical Engineering, Tampere University of Technology and the Universities of Oulu, Turku and Jyväskylä. Tekniikan edistämissäätiö (TES) has also given me a grant for this thesis. Their support is also gratefully acknowledged.

Once more, thanks to all who have supported me throughout my life.

Espoo, November 2013

Juha Backman

List of publications

- PUB I Backman J, Oksanen T, Visala A. (2012). Navigation System for Agricultural Machines: Nonlinear Model Predictive Path Tracking. *Computers and Electronics in Agriculture*. 82, 32-43
- PUB II Backman J, Oksanen T, Visala A. (2010). Nonlinear model predictive trajectory control in tractor-trailer system for parallel guidance in agricultural field operations. *In Proc. Agricontrol 2010, IFAC International Conference*. Kyoto, Japan, December 6-8.
- PUB III Backman J, Oksanen T, Visala A. (2009). Parallel guidance system for tractor-trailer system with active joint. *In Proc. Joint International Agricultural Conference (JIAC 2009)*. Wageningen, the Netherlands, July 6-7.
- PUB IV Backman J, Oksanen T, Visala A. (2013). Applicability of the ISO 11783 network in a distributed combined guidance system for agricultural machines. *Biosystems Engineering*. 114, 306-317
- PUB V Backman J, Kaivosoja J, Oksanen T, Visala A. (2010). Simulation Environment for Testing Guidance Algorithms with Realistic GPS Noise Model. *In Proc. Agricontrol 2010, IFAC International Conference*. Kyoto, Japan, December 6-8.
- PUB VI Backman J, Oksanen T, Visala A. (2011). Semi-automatic Tuning of NMPC Based Trajectory Control System in Agricultural Machine. *In Proc. 18th IFAC World Congress*. Milano, Italy, August 28-September 2.

- PUB VII Backman J, Oksanen T, Visala A. (2012). Path Generation Method with Steering Rate Constraint. *In Proc. 11th International Conference on Precision Agriculture*. Indianapolis, Indiana USA, July 15-18.
- PUB VIII Backman J, Oksanen T, Visala A. (2013). Collision Avoidance Method with Nonlinear Model Predictive Trajectory Control. *In Proc. Agricontrol 2013, IFAC International Conference*. Helsinki, Finland, August 28-29.
- PUB IX Backman J, Oksanen T, Visala A. (2013). Front Wheel Skidding Compensation System in Snow Ploughing. *In Proc. Agricontrol 2013, IFAC International Conference*. Helsinki, Finland, August 28-29.

Contributions of the author

PUB I: “Navigation System for Agricultural Machines: Nonlinear Model Predictive Path Tracking”

The article was written and its contents were solely based on the work by the author under the guidance of Dr Timo Oksanen and Professor Arto Visala.

PUB II: “Nonlinear model predictive trajectory control in tractor-trailer system for parallel guidance in agricultural field operations”.

The conference paper was written and its contents were solely based on the work by the author under the guidance of Dr Timo Oksanen and Professor Arto Visala.

PUB III: “Parallel guidance system for tractor-trailer system with active joint”.

The conference paper was written and its guidance part was solely the work of the author under the guidance of Dr Timo Oksanen and Professor Arto Visala. The results of this paper are based on the work done by the author while completing his Master’s thesis.

PUB IV: “Applicability of the ISO 11783 network in a distributed combined guidance system for agricultural machines”

The author had the main responsibility in writing this article. The author has also developed the software architecture presented in the article.

Dr Timo Oksanen has expertise in the ISO 11783 standard and he and the author came up with the required changes for the standard and also the proposed implement steering quantity.

PUB V: “Simulation Environment for Testing Guidance Algorithms with Realistic GPS Noise Model”

The publication was jointly written by the author and Mr Jere Kaivosoja; however, the author had the main responsibility. The author developed the simulation environment that was described in the publication.

Jere Kaivosoja has expertise in positioning systems. He developed the GPS noise model that was used in the simulator.

PUB VI: “Semi-automatic Tuning of NMPC Based Trajectory Control System in Agricultural Machine”

The conference paper was written and its contents were solely based on the work by the author under the guidance of Dr Timo Oksanen and Professor Arto Visala.

PUB VII: “Path Generation Method with Steering Rate Constraint”

The conference paper was written and its contents were solely based on the work by the author under the guidance of Dr. Timo Oksanen and Prof. Arto Visala.

PUB VIII: “Collision Avoidance Method with Nonlinear Model Predictive Trajectory Control”

The conference paper was written and its contents were solely based on the work by the author under the guidance of Dr Timo Oksanen and Professor Arto Visala.

PUB IX: “Front Wheel Skidding Compensation System in Snow Ploughing”

The conference paper was written and its contents were solely based on the work by the author under the guidance of Dr Timo Oksanen and Professor Arto Visala.

Nomenclature

Symbols

C	number of measurements associated with the cluster
D	nominal obstacle avoiding distance
$J(x, \dot{u} t_k)$	cost function of the state
K	filtering weight in the heading estimation
L_{est}	estimated lateral distance from the marking furrow
L_{meas}	measured lateral distance from the marking furrow
M	prediction horizon size
$O_{\{n\}}$	centre point of the turning circle · at the iteration n
P_i	position in the path
$Pitch$	inclination angle around the y-axis of the vehicle
Q	weighting matrix of the states
$R_{\dot{u}}$	weighting matrix of the control derivatives
R_u	weighting matrix of the controls
$Roll$	inclination angle around the x-axis of the vehicle
T	sampling period or control cycle time
V	covariance matrix of the measurement noises
W	covariance matrix of the process noises
X	position corresponding to the state $x_{\{x_R, y_R\}}$
Yaw	rotation around the ground z-axis (heading)
\widetilde{Yaw}	heading measured by the gyro with unknown zero position
Yaw_{bias}	difference between north and integrated gyro
$Yaw_{compass}$	heading measured by the RTK-GPS
a	wheelbase of the tractor
b	distance to the attachment point from the rear axle
c	length of the drawbar
d	distance to the seed coulters from the drawbar
d_-	perpendicular distance to the line segment $P_{i-1} - P_i$
d_+	perpendicular distance to the line segment $P_i - P_{i+1}$
$f(x, u)$	model of the system

$f_{est}(\hat{x}, u)$	estimation model for the system
$h(\hat{x})$	measurement function
k_{center}	curvature at the centre of the path in headland turning
k_{end}	curvature at the path end in headland turning
k_{limit}	curvature limit
k_{start}	curvature at the path start in headland turning
k_v	dynamic coefficient of the speed dynamics
k_α	dynamic coefficient of the steering dynamics
k_γ	dynamic coefficient of the joint dynamics
l_x	laser scanner's axial distance from the centre of the vehicle
l_y	laser scanner's cross-axial distance from the centre of the vehicle
p_x	plough's axial distance from the centre of the vehicle
p_y	plough's cross-axial distance from the centre of the vehicle
r_n	distance to the path tangent from the turning circle n
r_u	reference trajectory for the controls
r_x	reference trajectory for state
t_-	position of the closest point along the line segment $P_{i-1} - P_i$
t_+	position of the closest point along the line segment $P_i - P_{i+1}$
t_k	time instant k
u	control vector
$u^*(t_{k+1} t_k)$	optimal control values for the time instant t_{k+1} calculated at the time t_k
u_l	limited control value
\hat{v}	estimated speed
v_d	desired speed
v_t	realised vehicle speed
x	state vector
$x(t_{k+j} t_k)$	predicted state for the future time t_{k+j} at the time t_k
\hat{x}	estimated state vector
(x_E, y_E)	centre position of the implement
(x_L, y_L)	position of the laser scanner
(x_P, y_P)	position of the marking plough
(x_R, y_R)	centre position of the rear axle
$(x_{cluster}, y_{cluster})$	position of the cluster centre
(x_{meas}, y_{meas})	measured position of the obstacle
\hat{y}	estimated measurement

α_d	desired steering angle
α_{slip}	slip angle of the front wheels
α_t	realised steering angle
$\hat{\alpha}$	estimated steering angle
β	angle between the tractor and the trailer
γ_d	desired joint angle
γ_t	realised angle of the controlled joint
δ	slipping factor of the front wheels
$\Delta x_{\{x_R, y_R\}}$	tractor's distance from the reference trajectory
θ	heading angle of the tractor
μ	coefficient which varies according to the angle between the vehicle and the obstacle
Ψ	boolean value whether or not to use the cost from the obstacle
ω_{Yaw}	projected measurement of the rotation rate around the z-axis of the ground
ω_{earth}	measured earth rotation rate
ω_{gyro}	raw rotation rate measured by the gyro

Operations

$\ \cdot\ _A^2$	weighted vector norm ($\ a\ _A^2 = a^T A a$)
$\sigma(\cdot)$	standard deviation of the measurement ·
$\tau(\cdot)$	delay (sampling periods) of the measurement ·
$\cdot_{\{a,b\}}$	components a and b of the vector ·
\cdot_{max}	maximum value of the variable/vector ·
\cdot_{min}	minimum value of the variable/vector ·

Abbreviations

ARX	autoregressive with exogenous terms
BFGS	Broyden–Fletcher–Goldfarb–Shanno
CAN	Controller Area Network
DOP	Dilution of Precision
ECU	Electronic Control Unit
EFFIMA	Energy and Life Cycle Cost Efficient Machines
FIMECC	Finnish Metals and Engineering Competence Cluster
FIR	Finite Impulse Response
EKF	Extended Kalman Filter
FOG	Fibre Optic Gyro
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HQP	Huge Quadratic Programming
IMU	Inertial Measurement Unit
INS	Inertial navigation system
LQR	Linear Quadratic Regulator
LRL	Left-Right-Left
LSL	Left-Straight-Left
LSR	Left-Straight-Right
MIMO	Multiple-Input Multiple-Output
MPC	Model Predictive Control
MSE	Mean Squared Error
MTT	Agrifood Research Finland
NMPC	Nonlinear Model Predictive Control
NP	Non-deterministic Polynomial-time
PGN	Parameter Group Number
PTO	Power Take Off
QP	Quadratic Program
RLR	Right-Left-Right
RSR	Right-Straight-Right
RSL	Right-Straight-Left
RTK-GPS	Real-Time Kinematic GPS
SQP	Sequential Quadratic Programming
TEKES	Finnish Funding Agency for Technology and Innovation
VRP	Vehicle Routing Problem
VRS-GPS	Virtual Reference Station RTK-GPS

Definitions

active implement control

Implement guidance scheme where the implement has its own steering mechanics.

actuating system

Includes the devices and methods that realise the actual steering function.

combined navigation

Navigation system that takes the implement position into account.

collision avoidance

System that prevents the vehicle from colliding with any obstacle.

decentralised navigation

The measurement instruments and the actuators are not directly wired to the guidance controller and are not necessarily provided by the navigation system supplier.

generic navigation system

The implement's kinematics is not limited to any specific case.

guidance system

Includes the devices and methods that calculate the steering commands based on the information obtained from the positioning system.

navigation system

The concept that includes all the devices and methods that are needed to control the position of the vehicle.

passive implement control

Implement guidance scheme where the implement does not have a steering mechanics of its own.

path planning

Generates reference path for the guidance system.

path tracking

Guidance method that does not determine the time instance when the vehicle should be at a certain point on the reference path.

positioning system

Includes the devices and methods that are needed to determine the position and orientation of the tractor or the implement.

task planning

System or method that decides when certain agricultural operations are carried out.

trajectory tracking

Guidance method where the path positions and time instants are coupled.

Contents

PREFACE.....	VII
LIST OF PUBLICATIONS.....	IX
CONTRIBUTIONS OF THE AUTHOR.....	XI
NOMENCLATURE	XIII
SYMBOLS	XIII
OPERATIONS.....	XV
ABBREVIATIONS.....	XVI
DEFINITIONS	XVII
CONTENTS	XIX
1. INTRODUCTION	1
1.1 BACKGROUND AND MOTIVATION.....	1
1.2 THE SCOPE AND THE OBJECTIVES OF THE THESIS	3
1.3 SUMMARY OF THE PUBLICATIONS.....	5
1.4 CONTRIBUTIONS OF THE AUTHOR	7
1.5 STRUCTURE OF THE THESIS.....	8
2. AGRICULTURAL NAVIGATION SYSTEMS	9
2.1 HISTORY OF AUTONOMOUS VEHICLES	9
2.2 CONTROL STRATEGIES FOR AUTONOMOUS VEHICLES.....	11
2.3 MODEL PREDICTIVE CONTROL AND GUIDANCE	12
2.4 GUIDANCE WITH A TRACTOR-TRAILER SYSTEM.....	15
2.5 TUNING OF THE NONLINEAR MODEL PREDICTIVE CONTROLLER.....	16
2.6 PATH PLANNING METHODS	17
2.7 COLLISION AVOIDANCE METHODS.....	19
2.8 COMMERCIAL GUIDANCE SYSTEMS	20
2.9 THE ISO 11783 SERIES OF STANDARDS	21
3. MODULAR AGRICULTURAL MACHINES.....	23
3.1 TRACTOR-TRAILER SYSTEM.....	23
3.2 POSITIONING AND GUIDANCE DEVICES.....	25
3.3 REAL WORLD DISTURBANCES	27
3.4 MODEL OF THE TRACTOR-TRAILER SYSTEM.....	29
4. METHODS.....	33
4.1 NONLINEAR MODEL PREDICTIVE PATH TRACKING	33
4.1.1 <i>From trajectory tracking to path tracking</i>	36
4.1.2 <i>Collision avoidance</i>	39
4.2 PATH PLANNING	44
4.2.1 <i>Spiral connection method</i>	44
4.2.2 <i>LRL and RLR turnings</i>	45
4.2.3 <i>LSL and RSR turnings</i>	47
4.2.4 <i>LSR and RSL turning</i>	49
4.2.5 <i>Path Smoothing</i>	50
4.2.6 <i>Simplified path planning</i>	52
4.3 STATE ESTIMATION	53

4.3.1	<i>Extended Kalman Filter</i>	53
4.3.2	<i>Heading measurement in estimation</i>	55
4.3.3	<i>Local measurement using a laser scanner</i>	57
4.4	THE STRUCTURE OF THE NAVIGATION SYSTEM	59
4.4.1	<i>Components of the navigation system</i>	59
4.4.2	<i>Software architecture</i>	61
4.4.3	<i>Scheduling</i>	65
4.5	SIMULATOR	66
4.5.1	<i>Simulation model</i>	67
4.5.2	<i>GPS noise model</i>	68
4.6	PARAMETER ESTIMATION	71
4.6.1	<i>Parameter types</i>	71
4.6.2	<i>Estimation process</i>	73
5.	RESULTS	77
5.1	PARAMETERS OF THE SYSTEM	77
5.1.1	<i>Estimated parameters in simulator</i>	78
5.1.2	<i>Parameters of the test configuration</i>	79
5.1.3	<i>Parameters used in the tests</i>	79
5.2	PATH GENERATION	81
5.2.1	<i>Headland types</i>	81
5.2.2	<i>Complete field</i>	82
5.3	PATH TRACKING	83
5.3.1	<i>Comparison to traditional path tracking</i>	83
5.3.2	<i>Estimate correction with laser scanner measurements</i>	85
5.3.3	<i>Collision avoidance</i>	87
5.3.4	<i>Prediction horizon length in the optimization</i>	89
6.	DISCUSSION	91
6.1	PARAMETER ESTIMATION	91
6.2	PATH GENERATION	92
6.3	PATH TRACKING	93
6.4	COLLISION AVOIDANCE	95
6.5	NAVIGATION SYSTEM ARCHITECTURE	95
6.6	PROPOSED CHANGES TO THE ISO 11783 STANDARD	96
7.	CONCLUSIONS	99
	REFERENCES	101
	APPENDIX: PUBLICATIONS	107

1. Introduction

This first chapter gives a brief introduction to the background of the thesis and the history related to the research. The scope and the objectives of the thesis are defined and also the author's contribution is highlighted. Finally, the structure of the thesis is presented.

1.1 Background and motivation

Agriculture has a long history behind it, but it is still undergoing many changes. At the beginning, cultivation was manually done by hand. Over the years, different kinds of tools were developed and people began to use animals for different tasks. The mechanisation of agriculture took place in early 20th century when horses and oxen were replaced by tractors. More and more implements have been invented all the time, and the size of machinery has also grown since the early days of mechanisation. However, the basic concept of a tractor and an implement has remained the same since the very first concepts. Typical modern-day agricultural operations include, for example, ploughing, harrowing (Figure 1.1), sowing and harvesting. One person can perform the whole operation alone using just a tractor and an implement.



Figure 1.1. Harrowing with a modern-day tractor and implement (AGCO Advanced Technology Solutions, 2012).

Lately, more and more electronics have been added to tractors and also to implements. The reason is evident. There is a growing demand to increase the efficiency of the machinery and improve the way in which resources are used. Just increasing the size of the machinery is no longer sufficient. Instead, machines need to be used in a more intelligent way. The challenge is that there are different manufacturers for tractors and implements. There are numerous different combinations of tractors and implements and they all have to work together.

The workload of the tractor's driver has increased as the machinery has evolved. Because of the increased workload, the driver's attention decreases during the working day. When his or her attention wanes, the accuracy and the efficiency of the work drops and, most of all, the safety of the work deteriorates. In order to reduce the workload, some of the tasks have been automated. One of those is the task of driving the tractor.

This thesis has been completed at the Department of Automation and Systems Technology, where the author participated in the Autonomous Systems research group. The research group has been studying automation in agricultural machines for ten years. The objectives of automation cover mostly tractor-implement systems, using standard networking technology to realise a control system. The ISO 11783 standard has become a true industrial standard during these ten years and it is currently the workhorse when talking about tractor-implement system control and automation. The study of combined navigation began in 2006-2007, when the first experiments to detect local positioning from field drawn features were done; at first, this was done by using machine vision technology and image processing, and later by using a laser distance scanner. After the first promising experiments, a complete system for guiding both the tractor and the implement was built in 2008.

Currently, a national research project, 'Agromassi', is focusing on the issue of automation. The objective of the project is to study the use assisting and adaptive features for tractor-implement systems. Another objective is to further develop a tractor-trailer guidance system. The participants in the project include MTT Agrifood Research Finland, the University of Helsinki and 11 Finnish companies (Arctic Machine, Elho, Junkkari, Kemira, Parker-Vansco, Potila, Suonentieto, Valio, Valtra, Vieskan Metalli and Wapice). The project is part of the Energy and Life Cycle Cost Efficient Machines (EFFIMA) research programme, which is managed by the Finnish Metals and Engineering Competence Cluster (FIMECC) and funded by the Finnish Funding Agency for Technology and Innovation (TEKES) together with participating research institutes and companies.

1.2 The scope and the objectives of the thesis

In this thesis, the *navigation system* is considered to be the concept that includes all the devices and methods that are needed to control the position of the vehicle. In other words, the navigation system consists of a *positioning system*, *guidance system* and *actuating system*. It may also include *path planning* and *collision avoidance* systems. Above the navigation system is *task planning*. In task planning, the farmer decides when and how certain operations will be carried out or a highly sophisticated farm management system proposes these operations. Figure 1.2 provides an illustration of the conceptual structure of the navigation system using these and several other definitions.

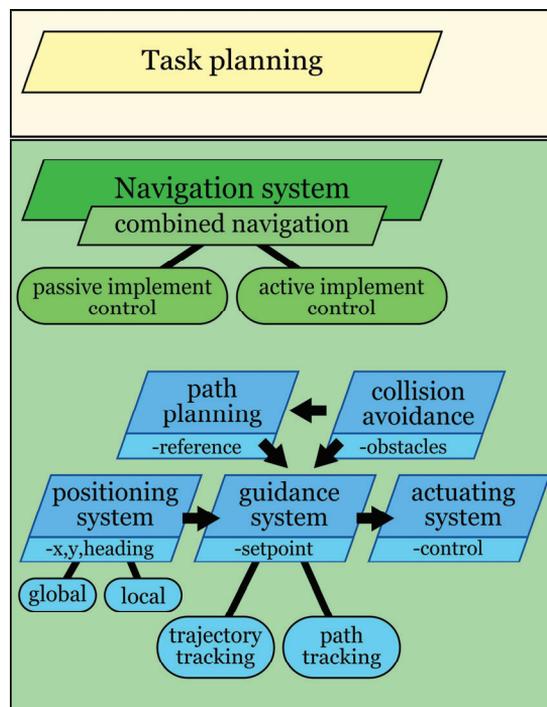


Figure 1.2. The conceptual structure of the navigation system. Task planning is used to determine the order of operations for the agricultural fields.

The main function of the path planning is to generate a reference path for the guidance system. In case path planning is not automatic, the driver has to decide the direction and the order of the driving lines. The positioning system includes the devices and methods that are needed to estimate the position and orientation of the tractor or the implement. The guidance system includes the devices and methods that calculate the steering commands based on the information obtained from the positioning system and from path planning. Finally, the actuating system includes the devices and methods that realise the actual steering function, e.g. that turn the

steering wheels of the tractor based on the setpoint obtained from the guidance system. The collision avoidance system can be included to insure that the vehicle does not collide with anything.

The objective of the navigation system in agricultural operations is to control the trajectory of the vehicle, to keep it within a constant distance of the adjacent driving line. Or, in agricultural terms, it is to lay the swaths side by side (like in Figure 1.1). Most navigation systems concentrate on keeping the tractor at a constant distance from the adjacent driving line, even if the objective is to consider the functional point of the implement rather than the tractor's position. In this thesis, the navigation system that takes the implement position into account is referred to as a *combined navigation* system.

In order to be useful, the navigation system must be able to drive as fast as a human driver and be at least as accurate as a human being. Most of the autonomous systems drive at much slower speeds than traditional agricultural working machines or achieve less accuracy when operated at higher speeds. In this thesis, the objective is to build a combined navigation system that is able to navigate at a speed of at least 12 km/h with less than 10 cm lateral error. This speed is considered sufficient for most agricultural operations that require a high degree of precision. A real-time solution with a processor comparable to a desktop computer is also required in order to experimentally evaluate the system under real field conditions.

The objective of this thesis is also to discuss and present the ways in which a *decentralised* and *generic* combined navigation system can be realised using the ISO 11783 network and to discuss how the ISO 11783 network should be improved to support this system. Within this context, a generic navigation system is defined in such a way that the implement's kinematics is not limited, for example, to a trailer with steering wheels; instead, other kinematics are also considered. Also within this context, the decentralised navigation system is defined in such a way that the measurement instruments and the actuators are not directly wired to the guidance controller and are not necessarily provided by the navigation system supplier. This study does not consider how to decentralise the guidance controller or algorithm through standard interfaces for a multi-vendor, tractor-implement system.

The guidance system can be based on either *trajectory tracking* or *path tracking*. With trajectory tracking, the path positions and time instants are coupled; in other words, the vehicle needs to be in a certain place at a certain time. In turn, path tracking does not determine the time instance when the vehicle should be at certain place. In field robotics the reference trajectories of these methods are also defined as *time based trajectory* (in

trajectory tracking) and *space based trajectory* (in path tracking). This thesis focuses solely the path tracking problem.

Furthermore, combined guidance can be based on either a *passive implement control* or an *active implement control*. With passive implement control, the implement does not possess its own steering mechanics. Instead, a tractor's steering capabilities are used to control the position of the implement. With active implement control, the implement has its own steering mechanics. The implement's steering mechanics can be realised by, for example, using a laterally moving hitching point, with articulated drawbar or steering wheels in the implement. The focus of this thesis is to investigate active implement control methods.

To summarise, the objectives of this thesis are as follows:

- to study a combined navigation system that is able to drive at a speed of at least 12 km/h with less than 10 cm lateral error under real field conditions and that is based on path tracking and active implement control methods;
- to discuss and present the ways in which a decentralised and generic combined navigation system can be realised using the ISO 11783 network.

1.3 Summary of the publications

As discussed in Section 1.1, the navigation research was started by studying the localisation problems pertaining to tractors and implements. After concluding that the low-cost solutions do not achieve a reasonable level of precision, the research focus moved on to that of path planning and the path tracking.

The first approach, which aimed to realise the combined path tracking of a tractor and trailer, involved the Model Predictive Control (MPC) method. The first approach is explained in PUB III, where a tractor-trailer system for navigation was presented for the first time. A Matlab-integrated Model Predictive Control Toolbox provided by Mathworks was used for real-time control. It was quickly noticed that the linear approximation of the kinematics is not sufficient and there were also problems with keeping the cycle time constant. However, the localisation problem was partially solved in that research article and the tractor-trailer system was equipped with sensors and actuators.

After the first experiments, the Nonlinear Model Predictive Control (NMPC) method was tried. The second approach is explained in PUB I and in PUB II. It was concluded that the NMPC method is a feasible method for realising path tracking. However, it requires much more expertise to tune the parameters of the NMPC method compared to traditional path tracking

methods. For that reason, the already existing simulator was further developed. The improved simulator is explained in PUB V. In addition to the simulator, semiautomatic offline parameter estimation methods were also studied. The semiautomatic parameter estimation is explained in PUB VI.

Slipping is a significant source of interference in the field. As such, it was already considered in PUB I and in PUB II. However, the solution used in these publications differs from the solutions presented in earlier literature. The problem was discussed again in PUB IX. This time, the slipping was modelled in the same way that it is presented in the earlier literature. However, the application was also different from the original; because of external forces that pushed the tractor sideways, the assumptions made for the original slipping model were not valid. In PUB IX, the final form of the tractor pose estimation was also explained.

The navigation system needs more than just a path tracking method in order to be operational. The path needs to be generated somehow. The problem is called ‘coverage path planning’. The size of the problem is worthy of a thesis in its own right, so the scope of this thesis is limited only to a basic path planning approach. It was decided that the field is operated in the same way as when the driver does it without automation; that is to say, the current path was created from an adjacent driving line and the longest edge of the field determined the main orientation of the driving lines. However, there was still a problem with feasible headline turnings. The simplified form of path planning with a feasible headland turning algorithm is explained in PUB VII.

The navigation system could be operational with just feasible path planning and path tracking methods together with a positioning system. However, the driver still needs to handle special situations, such as driving around obstacles. If the obstacles are not on the map or if they are moving, a system capable of detecting and avoiding them is needed. The obstacle avoidance system used in connection with the NMPC-based path tracking method is explained in PUB VIII.

Finally, when all of the pieces of the puzzle are put together, it can be concluded what it is required to build a commercially distributed combined navigation system for a tractor and an implement with a true multiple-input multiple-output (MIMO) controller. The system architecture viewpoint (both the physical and the software perspectives) is discussed in PUB IV.

1.4 Contributions of the author

It is believed that the main contributions of this thesis in order of the importance are as follows:

1. The normal way of using the NMPC method is to define the time-based trajectory that the process model should follow. In this thesis, the NMPC method is used for the first time as a path tracking method for a modular vehicle. With the path tracking method, the desired trajectories of the model's states are not coupled with the time.
2. Both commercial and research-based agricultural navigation systems are usually designed to support tractor-only navigation. The implement is usually omitted; this is especially the case when the implement is trailed. Even if the trailed implement is considered, the solution usually is not based on a MIMO approach or is not suitable for field conditions due to slipping. In this thesis, a true MIMO controller for both a tractor and trailed implement is designed and the uncertainties and interferences are taken into account by using a state estimator and different types of measurements.
3. The proposed method for realising the path tracking method is not case dependant. The underlying vehicle model can easily be replaced without making major modifications to the controller. Furthermore, the proposed method is suitable for both active implement control and for passive implement control by using different vehicle models or control limits.
4. Navigation systems are usually closed single-vendor systems. The proposed design for this navigation system is both modular and open. It supports the decentralisation of actuators and sensors and different types of implement with different kinematics and different manufacturers through the use of a standardised interface, the ISO 11783 network. Required modifications to the standard are also proposed.
5. There are several ways to realise obstacle avoidance within the NMPC-based guidance system. Usually, additional constraints or cost functions are used. In this thesis, the obstacle avoidance method is augmented to the NMPC method in such a way that it does not increase the algorithmic complexity of the original optimisation problem.

6. The proposed NMPC-based path tracking navigation system works better if the path is feasible. In order to generate a feasible path in the headlands, a headland turning path is generated using a steering rate constraint extension in the Dubins' Curves method. It is believed that the trajectory is close to an optimal path under the constraints given by the vehicle.
7. The parameters of the NMPC method and the associated state estimator are difficult to tune. There is no general method for tuning them; usually, rules of thumb are used. In this thesis, a semi-automatic method for offline parameter estimation is proposed.
8. Most of the development, testing and tuning was done using a hardware-in-loop simulator environment. The importance of an hardware-in-loop simulator when developing navigation systems is discussed.

1.5 Structure of the thesis

The rest of this thesis is organised in the following way: the second chapter introduces the agricultural navigation systems, provides a literature survey and discusses state-of-the-art methods. The third chapter introduces the application where the experimental navigation system is applied. In the fourth chapter, the proposed methods for the navigation system are presented. The results of the experiments are presented in the fifth chapter. Finally, the discussion and the conclusions are presented in sixth and seventh chapters.

2. Agricultural Navigation Systems

This chapter concentrates on the existing literature for navigation systems. First, a general history of autonomous vehicles and the evolution of control strategies for navigation systems are provided. After that, more thorough reviews of guidance with Nonlinear Model Predictive Control and guidance with tractor-trailer systems are presented. Tuning is an important part of the NMPC method, so tuning methods for the NMPC method are reviewed separately. After that, path planning algorithms are reviewed. Collision avoidance can be part of a path tracking or path planning system, so a review of collision avoidance methods is presented separately. Finally, current commercial navigation systems are briefly reviewed and the agricultural communication standard, ISO 11783, is briefly discussed.

2.1 History of autonomous vehicles

Scholars have been doing research on autonomous vehicles for several decades now. There are many famous and long-term projects in which autonomous vehicles have been, or are being, developed. One of the first truly autonomous cars was the VaMoRs. Already in 1987, it was able to drive on the basis of machine vision without human intervention for more than a hundred kilometres on the German Autobahn (Dickmanns, 2007). The Carnegie Mellon University Navigation Laboratory has built driverless car platforms (NavLab) for the development of navigation algorithms (Omead, 1990). Probably the most famous competitions between autonomous vehicles are the Darpa Grand Challenge and the Darpa Urban Challenge (Darpa, 2007). The vehicles that have taken part in those competitions are quite advanced, having many sensors for detecting the road and other elements of the environment, such as obstacles. In addition, many computers are also required to process all that information and to navigate the car.

Automatic steering systems for agricultural machines have also been under development for almost a century now (Heraud and Lange, 2009). However, automatic steering systems in practical applications only became possible after the development of the Global Positioning System (GPS) in

the mid-1990s (Heraud and Lange, 2009). The difference between agricultural and road applications is that something can be assumed about the conditions for agricultural operations. The field usually has quite an even surface and moving obstacles are not usually present, except other working machines. Furthermore, the boundaries of the field plots are mapped and the environment is structured in this sense. The main limiting factor is that the manufacturing price of navigation systems has to be much lower than for the cars in the Darpa competitions in order to be reasonable for real-life commercial applications. The accuracy requirements are, however, quite high: usually less than a 10 cm relative error between adjacent driving lines.

There are different approaches to building an autonomous agricultural vehicle. One is to build it completely from scratch. Another approach is to take a commercial product and modify it. The third approach is to use a commercial product as it is and to add a navigation system as an accessory to it. One project in which the agricultural robot was built completely from scratch was the Modulaire platform (Rintanen et al., 1996). The Modulaire platform was a tracked, off-road vehicle. It had real-time kinematic GPS (RTK-GPS) and a fibre-optic gyro for navigation purposes. Another similar robot was the Weedy (Ruckelshausen et al., 2006), a four-wheel, steered robot for mechanical weed control. It did not, however, use GPS to measure its position. Instead, it had a colour camera to track plants and a gyroscope for headland turnings.

HortiBot represents the second approach, in which a radio-controlled slope mower was transformed into a tool carrier robot (Jørgensen et al., 2006). Also, Blackmore et al. (2004) developed an autonomous tractor from a small garden tractor. The irony was that it required two persons to operate it: one to give it instructions and another for safety reasons. Nagasaka et al. (2009) have developed an autonomous rice transplanter.

The third approach was to equip a standard tractor for autonomous operations. Lenain et al. (2005, 2006) concentrated more on path tracking and position estimation, but they used a standard tractor as a test platform. Strentz et al. (2002), for their part, focused on semi-autonomous tractors for spraying applications. Werner et al. (2012, 2013) also used a standard tractor with an implement as a test platform for a navigation system.

Many other similar projects exist for each category listed above. For example, researches Keicher and Seufert (2000), Reid et al. (2000) and Torii (2000) have all reviewed some of the other projects. Commercial products for automatic guidance or automatic steering are also available. Those are reviewed shortly later on, in Section 2.8.

2.2 Control strategies for autonomous vehicles

With autonomous vehicles, the software architecture and control strategies have also improved tremendously over the years. The basic and earliest strategy for controlling a robot was sense-model-plan-act framework. First, the robot senses the environment and stores the information into some kind of database. Then, the robot's movements are planned according to this database. Finally, the first step of the plan is executed and the robot moves. The benefit of this approach is that the robot's behaviour is predictable and the researcher can know beforehand what should happen. The drawback is that this may require a great deal of computer power. Situations may change during the planning phase and something catastrophic may occur (Murphy, 2000).

Nowadays, a more popular strategy is reactive control. Scientists have studied the behaviour of animals and extracted some primitive laws that animals follow. These laws, or behaviours, are then adapted to fit the robots. The difference between this strategy and the first control strategy is that the planning phase is skipped. The sensing elements launch some actions directly as a result of certain sensing input. The benefit of this approach is that it is computationally light and easy to develop. The drawback, however, is that the robot's movements are not always predictable. Another drawback is that, without any higher-order planning, the robot may end up in situations that it cannot handle (Murphy, 2000).

The solution to these problems is to use both strategies. The control task can be divided into layers. The reactive part is used whenever possible, whereas the planning part is launched depending on the situation. This hybrid control strategy exploits the best features of both strategies (Murphy, 2000). In order to make a system that is both reactive and capable of planning, some kind of hierarchical architecture is required. The Autonomous Robot Architecture (AuRA) was the first navigation system to use this kind of hybrid architecture (Arkin and Balch, 1997).

For agricultural robots, Blackmore et al. (2002) have proposed an object-oriented architecture with the message passing through a common bus. Tasks are divided into subprograms called agents, which can be replaced or modified to fit a certain application. The hardware is abstracted into a Hardware Abstraction Layer (HAL) agent interface between the software and the devices that it will be operated on. This agent also includes the critical control loops. Such control loops are, for example, steering and speed controllers. These controllers utilise the inverse kinematic model to calculate the proper control values. This HAL agent can, in a way, be considered the reactive part and everything above it belongs to the planning part.

Although the above-mentioned system has the desired structure, the control of the vehicle can be more precise. One way to utilise the kinematic model more effectively is to use the Model Predictive Control (Maciejowski, 2002). The MPC method predicts the future according to the model of the system and tries to minimise some given criterion, while also taking into account the model's restrictions. The criterion is usually a sum of the squared errors of the desired and actual output values. Because the system model of the vehicle is usually nonlinear, a Nonlinear Model Predictive Control is required. The ways in which NMPC method is used in guidance systems are reviewed in the next section.

Another restriction of the above-mentioned system architecture proposed for an autonomous tractor is the requirement for a proprietary bus. The messages in the bus are not compatible with different manufacturers and are not harmonised with the ISO 11783 standard. The ISO 11783 standard is a widely accepted way to share information between the tractor and the different implements or manufacturer equipment in a common bus. The ISO 11783 standard is discussed in detail later on, in Section 2.9.

Darr et al. (2005) have utilized CAN-bus with messages similar to ISO 11783 standard in navigation system development. The focus on their research was to develop low-cost solution for tractor only navigation.

Ehrl and Auernhammer (2007) have proposed a Steer-by-Wire approach via the ISO 11783 network and discussed the requirements and applicability of the bus. While their results were promising, they also noted that further investigation is needed with, for example, pre-defined bus load scenarios. Also, they left the message content open and did not investigate the combined tractor-implement navigation system.

2.3 Model Predictive Control and guidance

In a recent survey, different existing guidance methods were extensively compared (Snider, 2009). The guidance methods were classified into three groups: the geometric approach, the kinematic control laws and the optimal control. The survey did not find any of them to be practical for every situation. Instead, they all have some characteristic advantages.

The most commonly used and simple guidance method is based on the geometric approach. The geometric relationship between the path and the vehicle has been exploited in these control laws. Often, a look-ahead distance is used to measure the error ahead of the vehicle. Such geometric guidance algorithms include, for example, Pure Pursuit (Amidi, 1990) and Vector pursuit (Wit, 2004).

More advanced guidance methods utilises the vehicle's kinematic model. The kinematic model is transformed into a chained form, and basic control

theory methods are used (Morin, 2008). The drawback of this approach is more complex implementation and not so intuitive tuning.

The most advanced guidance methods are based on the vehicle dynamics model and utilise optimal control methods. The most recent research topics make use of Model Predictive Control and its extension, Nonlinear Model Predictive Control, for the guidance methods. Those methods were also seen as the next logical evolutionary step in the survey conducted by Snider. By using these kinds of methods, the most accurate tracking is possible. However, the model must be perfect and the controller properly tuned for it to work.

Nonlinear Model Predictive Control is normally used in industrial plants and in process control to optimise the operating points of the controlled process. It is easier to implement NMPC method in these environments due to longer time constants. If the time constants of the system are smaller, as in vehicle trajectory control, the controller must run with a higher control cycle. This demands a high computing capacity for real-time control. However, in the last ten years or so, the developments in the computational capacity of desktop processors and also the developments in numerical methods for solving the optimisation problems have made it possible to use the NMPC method in real-time solutions for mobile robots. There is little research on how to use the MPC method for path or trajectory tracking purposes and the few available studies mostly deal with the computational requirements.

One way to lighten the computational requirements is to combine the MPC method and another control law that actually does the steering of the vehicle. The purpose of the MPC method is to improve the performance of the complementing control law. Kim et al. (2001) used the MPC method to predict wheel-ground slippage and by this way avoid the loss of wheel-ground contact. The method was proven to work in experiments with a three-wheeled vehicle on an inclined surface. Another study in which the MPC method was used to improve the performance of another control technique was carried out by Lenain et al. (2005). In their study, the MPC method was used for real-time control of the steering angle of the tractor. The desired steering angle was calculated using a nonlinear control law for the followed path. The MPC method was used to reduce the 'delay phenomenon' of the actual steering system.

In terms of developing automotive safety systems, Keviczky et al. (2006) used the NMPC method to stabilise the vehicle along a desired path while reducing the effect of wind gusts. The commercial NPSOL software package was used in the tests. The vehicle speed varied between 5 m/s and 17 m/s. They found that when the driving speed was increased, the corresponding

control and prediction horizons must be increased dramatically to achieve a stable performance. This increase then leads to problems with the computational complexity. To reduce the computational complexity, a suboptimal MPC controller based on successive online linearisation of the nonlinear vehicle model (LTV MPC) was later presented by Falcone et al. (2007). The resulting optimisation problem was recast as a quadratic program (QP) and solved using the MPC Toolbox for Matlab, which is available from The MathWorks, Inc. They found that the LTV MPC was able to stabilise the vehicle, even at high speeds, although the control horizon was reduced to one.

Kühne et al. (2005a) presented nonlinear MPC and linear MPC methods for solving the trajectory tracking problem on a nonholonomic wheeled mobile robot. They studied the computational effort required to solve the optimisation problems and compared the performance of both controllers. They found that, at the time of their research, the Nonlinear Model Predictive Controller was computationally too demanding to be solved in real-time with a prediction horizon larger than five and also that the linear MPC method performed well with a lower computational effort. But they also noted that the linear model is only valid near the reference trajectory. Furthermore, Kühne et al. (2005b) proposed an alternative way to formulate a cost function. They calculated the modified cost function using polar coordinates, and the weight of the state cost increased along with the prediction horizon. With this modified cost function, the computational effort was reduced and a better steady state performance was achieved.

Vougioukas et al. (2007) used a NMPC method to control the steering angle and the speed of their vehicle. The criterion was the difference between the desired trajectory and the predicted trajectory. Although the experiments were done completely in a simulator, they achieved good results and were able to show the advantage of this approach.

Gregor Klančar and Igor Škrjanc (2007) proposed a tracking-error, model-based, predictive control law for tackling the trajectory tracking problem for a nonholonomic wheeled mobile robot. The prerequisite was that the reference path be a smooth, twice-differentiable function of time. The control law was based on a linearised error dynamics model obtained around the reference trajectory. The resulting control law was analytically derived, and therefore, computationally light.

It is well known that the MPC controller is not guaranteed to be stable due to the finite prediction horizon. Dongbing Gu and Huosheng Hu (2006) developed a method to calculate the terminal state region and a corresponding controller as a way of ensuring the stability of the controller. However, the proposed controller needed a feasible initial solution. Also,

they noted that the computational efficiency was a problem and worthy of further investigation.

2.4 Guidance with a tractor-trailer system

All of the approaches mentioned in the previous section are intended for tractor-only navigation. In the existing literature, there are, however, numerous proposals for control laws for tractor-trailer systems. Most of the studies concern reverse motion with a trailer, because forward motion is considered naturally exponentially stable. According to Cariou et al. (2010) and Siew et al. (2009), these control laws are not well-adapted for an agricultural context due to delays and nonlinearities in the actuators and sliding conditions of the various soils. Cariou et al. (2010) studied headland manoeuvres with a trailer. Model Predictive Control was used to anticipate speed variations and reject significant overshoots in longitudinal motion in the same manner as in the study by Lenain et al. (2005). However, the trailer was again ignored in forward motion, whereas with backward motion the objective was to maintain a constant joint angle between the tractor and trailer.

Siew et al. (2009) modelled the behaviour of a tractor-implement-trailer (a tractor with two trailers; the first trailer was controlled by steering its wheels) system with sliding conditions. The controller for the trailer was not constructed. However, the impractical assumption of slippage-free motion was shown in this study. Also, Karkee and Steward (2010) studied the characteristics of a tractor and a single-axle, towed implement system. They derived three different models for a tractor-trailer system: a kinematic model, a dynamic model and a high-fidelity model. In the last model, the tire relaxation length was included in the dynamic model. The experiments showed that the kinematic model described the behaviour sufficiently well when the driving speed was less than 4.5 m/s and the input frequency less than 1 rad/s. A high-fidelity model was needed when the driving speed was increased. They also studied the closed-loop behaviour with the Linear Quadratic Regulator (LQR). The controller was used to stabilise the tractor and implement heading errors as well as the tractor lateral error. The controller was unstable at a 4.5 m/s forward velocity when the kinematic model was used. With the high-fidelity model, the controller was stable for a range of 0.5–10.0 m/s forward velocities. However, it was noticed that the linearisation of the model does not hold up when the steering or heading angle is above 10 degrees.

In another study, Werner et al. (2012) derived kinematic and dynamic models for the tractor and the towed implement. They also designed output feedback LQR controllers for both models. The control performance was

tested using a simulation, and the characteristics of open-loop systems were further analysed using eigenvalues. Again, it was noticed that the behaviour of both systems are quite similar at velocities of up to 4.5 m/sec. Werner et al. (2013) also developed a custom-made, actively steered implement with a multitude of actuators. The position of the implement could be controlled via actively steered wheels, actively steered coulters or an actively steered drawbar or with a side shift-frame, which was connected to a hitch point. A standard tractor, together with the implement and an output feedback LQR controller designed for the kinematic model, was used in the experiments. On a level field and with straight path, the tracking error was below 2 cm for both the tractor and the implement, and the standard deviation of the error was below 2 cm for the implement and below 2.5 cm for the tractor. However, Werner et al. noted that either a feedforward control or integrational error states are required for curved paths and if there are slopes in the field. The LQR control has also been used in other tractor-trailer experiments (e.g. Bevy, 2001; Bell, 1999).

There are even commercial solutions for tractor-trailer navigation. These solutions will be reviewed in Section 2.8.

2.5 Tuning of the Nonlinear Model Predictive Controller

As stated before, Nonlinear Model Predictive Controllers are widely used in industry plants and in process control, where they are used to optimise the operation point of the controlled process. Wojsznis et al. (2003) have proposed a practical approach for tuning an MPC controller for process control. The proposed approach is based on four steps: (i) designing the process model; (ii) establishing controller generation parameters; (iii) testing the controller response and behaviour in simulations; and, (iv) making online adjustments. The process is modelled in a Black-box manner using FIR and ARX models. The controller generation parameters are the prediction and control horizons and penalty matrices on the controlled output error and on the control moves. According to Wojsznis et al., horizon parameters are not suitable for tuning or setup. The control and prediction horizons are selected to be long enough when taking the process time constants into account. Also, the ratio between the weights is more important than the absolute values. For that reason, the penalty matrix on the controlled output error is assumed to be a scalar one. Wojsznis et al. have derived an experimental formula for calculating the penalty matrix on control moves based on the process model. In that way, the process model is the most important parameter to be tuned or estimated.

The problem of implementing an NMPC trajectory control in a real system can be solved by having a representative model of a system and a proper

state estimate. The Kalman Filter is a natural choice for the state estimator. Tuning the Kalman Filter involves designing the model of the estimated system and establishing the covariance matrices for the measurement noise (usually the symbols R or V) and system internal noise (usually Q or W). For chemical process control, Odelson et al. (2006) have proposed a method for estimating the covariance matrices automatically using the autocovariance least-squares technique. Especially in relation to GPS and INS integration, Mohamed et al. (1999), Congwei et al. (2003) and Weidong et al. (2007) have all studied covariance matching methods to estimate the Q and R matrices using innovation terms. The drawback of these methods is that they work only for a linear system model.

Jetto et al. (1999a) proposed an algorithm for updating noise covariance matrices using a nonlinear Extended Kalman Filter. In the proposed algorithm, the Q and R matrices are assumed to be diagonal matrices. The components of the diagonal matrices are updated based on the innovation terms, as in the linear case. Algorithms based on Fuzzy logic also exist for calculating weighting values for covariance matrices (e.g. Jetto et al., 1999b; Sasiadek et al., 2000).

The stability of the NMPC controller depends on the correct model of the controlled system, as well as on the correctness and stability of the state estimate.

2.6 Path planning methods

Another important part of the navigation system besides path tracking is path planning. There are at least three different cases for path planning in agricultural operations: the direction and the order of the driving lines are not limited (for example, sowing); the direction of the driving lines is given but the order is free (for example, silage harvesting); both the direction and the order of the driving lines are at least partially fixed (for example, ploughing). The selection of the path planning method is highly dependent on the agricultural operation to be realized. Algorithms for solving the driving lines for general operations have also been proposed. Oksanen and Visala (2009) have proposed two different methods for solving the direction of the driving lines: a split-and-merge approach and a predictive recursive online approach. With the first approach, the field is split into subfields, which are simple to drive. After the splitting, the best driving direction for each subfield is established. The second algorithm is an incremental algorithm that takes into account the machines and field current state and establishes the next nearly optimal swath for driving. Bochtis and Sørensen (2009) have proposed a method to tailor the driving order selection to fit a commonly known vehicle routing problem (VRP).

The method presumes that the optimal direction of the driving lines is predefined. Moreover, Bochtis and Sørensen (2010) have proposed a similar approach for multiple vehicles.

The direction and order of the driving lines generally determine the path that the path tracking system should follow. However, the path is not yet feasible. The transitions between driving lines, i.e. headland turnings, are still missing. Dubins (1957) has shown that, if the car has limited curvature and only forward motion is allowed, then the minimum path between two arbitrary positions can be found among from a set of six different turning types: LRL, RLR, LSL, RSR, LSR and RSL, where 'L' denotes a left-turning segment with maximum curvature, 'R' denotes a right-turning segment and 'S' denotes a straight segment. Furthermore, Reeds and Shepp (1990) have shown that if backward motion is also allowed, then the minimum path can be found from among a set of 68 different turning types consisting of, at most, four arcs with maximum curvature and one straight line segment. However, at the junction point between the different segments in the path, the curvature is discontinuous, or else steps appear.

To prevent discontinuities in the Dubins' or Reed-Shepp Curves, different solutions have been proposed. Parlange and Indiveri (2010) have proposed a method for calculating a smooth path with bounded curvature and a curvature derivative. However, the method is applicable only when there is a straight line segment between two arcs. Fraichard and Scheuer (2004) have proposed a method for extending Reeds and Shepps' turning types to paths with continuous and upper-bounded curvature and an upper-bounded curvature derivative. However, in certain cases the method does not produce paths that have an optimal length. In these cases, the curvature derivative is allowed to be smaller than the maximum curvature. The method is also designed to connect only configurations with a null curvature in initial and terminal positions.

There are also various proposals based on numerical optimisation (e.g. Fernandes et al., 1991; Oksanen and Visala, 2004; Vougioukas et al. 2006; Tu 2013). The problem with numerical optimisation is the computational complexity. The algorithms are heavy and there is no guaranteed solution at the given time window.

The path may also become infeasible in another way. If the coverage path planning method generates new driving lines by copying the adjacent driving line with some offset, the curvature of the inner curves may become too tight to drive. Or, if the coverage path planning method generates piecewise linear driving lines, there will be discontinuities between the linear segments.

Brezak and Petrovic (2011) have proposed a path-smoothing method for smoothing a path consisting of straight line segments by using clothoids. Also, Fleury et al. (1995) have introduced a method to smoothen a path described as broken lines with circles and connecting clothoids. Yang and Sukkariah (2010) have proposed an analytical method for path smoothing using cubic Bézier curves. Again, the original path consists of straight line segments between waypoints.

2.7 Collision avoidance methods

As stated previously, different navigation methods have been widely studied in the past years. The focuses of the studies have usually been on either path tracking or path planning methods. With these methods, a tractor is able to perform complete agricultural operations on a field. However, a driver or operator is still needed to monitor the system and to ensure that the tractor does not collide with anything.

Generally, the collision avoidance methodology is a widely studied research area in robotics. The methods can be divided roughly into two categories, although some of the methods can be used in both situations. The first category is collision-free path planning algorithms. The second one is more reactive real-time obstacle avoidance methods.

Within the context of agriculture, the field is usually quite static: the electricity poles, wells, bugholes and large rocks are more or less stationary. If all obstacles are known beforehand, then the route can be designed beforehand using a suitable coverage path planning method. One such method is, for example, previously mentioned by Oksanen and Visala (2009). However, there might be a situation where there is a known obstacle in the field but the position has not been mapped yet with the required degree of accuracy. For example, this may be the case if there is a moving object (animal, human or another machine) or the original map was imperfect. In such situations, there has to be some device that can recognise these obstacles and a method that can be used to recalculate the route or simply stop the navigation before the collision.

Widely used methods from the second category include, for example, the Potential Field method (Tilove, 1990), the Vector Field Histogram method (Borenstein, 1991) and the Velocity Objects method (Fiorini and Shiller, 1998).

The Potential Field method creates an artificial repulsive force field around the obstacle and an artificial attractive force around the target. The direction of the movement is chosen based on the artificial potential field.

With the Vector Field Histogram method, a set of candidate directions are created and the best one closest to the target is chosen. The candidate

directions are created according to the probability of obstacle density in every direction. If the density is below a certain threshold, then movement in that direction is allowed.

The Velocity Objects method uses candidate velocities rather than just directions. The new velocity is selected in the velocity space, where the obstacle's velocities are also added. Furthermore, the kinematic and dynamic constraints can be taken into account in the set of candidate velocities in the velocity space.

There are even a few studies on collision avoidance methods, especially in agricultural field operations; for example, there are studies by Noguchi et al. (2004) and Vougioukas (2007, 2009, 2012).

Noguchi et al. (2004) developed a concept for a master-slave system for farm operations. It employs two different algorithms for the slave tractor or robot: a GOTO algorithm and a Follow algorithm. With the GOTO algorithm, the tractor moves to a predefined location, whereas with the Follow algorithm it follows the master, which is operated by a person. The slave monitors the master position constantly through a radio link. The risk index is calculated based on the current position of the master and slave. The risk index indicates a potential risk of collision. Two actions are used to prevent collisions if the risk is too high: speed reduction and pathway correction. Noguchi et al. conducted simulations to prove the functionality of the system.

Vougioukas (2007, 2009, 2012) used the Nonlinear Model Predictive Control method to control the position of the vehicle. Moreover, they included collision avoidance in the controller by using the additional cost from distance sensor readings and also from other vehicles. The controller was able to follow a predefined path and also avoid collisions. The functionality of the method was again proven via simulations.

2.8 Commercial guidance systems

Commercial automatic steering applications for agricultural machinery only became possible after the development of the GPS system. The first commercial GPS-based guidance system was sold in Australia by Beeline in 1997. Short after that, IntegriNav and Trimble Navigation also produced their own GPS-based automatic steering systems (Heraud and Lange, 2009).

Nowadays, there are a number of manufactures for automatic steering systems. For example, Farm Journal Inc. (2011) lists 29 different commercial products with different capabilities. These commercial products follow predefined paths or adjacent driving lines and, therefore,

do not include any planning or reactive capabilities (John Deere, 2012; AGCO, 2012).

There are even commercial solutions for tractor-trailer navigation, for example John Deere's iGuide and iSteer (Deere, 2009a, 2009b). The difference between these systems is that the iGuide system is based on passive implement control, whereas the iSteer system is based on active implement control. However, it seems that both of these systems are add-ons for basic tractor-only guidance systems. With the iGuide system, the roll angle of the machine is monitored and slipping due to the slope is compensated by setting an offset from the path in the tractor's navigation system. The offset value is directly proportional to the roll angle. With the iSteer system, both the tractor and the trailer are kept on the desired path. The trailer has its own positioning and steering systems. In this manner, the tractor-trailer system has two separate navigation systems that have a common user interface. It seems that both of these systems are intended to be used mainly on straight driving lines.

Other manufactures have similar products; they include the AutoFarm AFTracker, Sunco Acura Trak, Orthman Tracker IV and Trimble TrueGuide. However, it seems that all of these products use either two separate controllers (the tractor and implement have their own) or an implement error is used directly as an offset value for tractor guidance.

2.9 The ISO 11783 series of standards

The ISO 11783 series of standards, 'Tractors and machinery for agriculture and forestry - Serial control and communications data network', was developed to support the exchange of information between different manufacturers' mobile agricultural machinery products. The need for a communication standard is evident; in a typical agricultural machine configuration, a tractor is connected to one or more implements that are manufactured by a different company than the one that manufactured the tractor.

The ISO 11783 standard is partially based on the SAE J1939 standard (SAE J1939:1994), which was developed for use in truck and bus applications. Both standards are based on the CAN 2.0B specification (CAN Specification Version 2.0:1991). The purpose of the ISO 11783 standard is to 'specify the method and format of transfer of data between sensor, actuators, control elements, information storage and display units whether mounted or part of the tractor, or any implements'. The market name for systems and devices that are proven to be compatible with the standard is ISOBUS.

Nowadays, the standard contains 13 parts. Part 1 is a general introduction to the standard series and it includes definitions and abbreviations. Parts 2-5 and part 12 specify the lower-level protocols, or protocol stack. The other parts specify the higher-level protocols for various applications in the network. Part 6 specifies the virtual terminal and the protocol for the corresponding client. Part 7 specifies implement messages, basically those used for tractor-implement communication. Part 8 covers the drivetrain. Part 9 specifies a tractor as a device in the network and a Tractor ECU (TECU). Part 10 specifies the task controller and the protocol for the corresponding client as well as the data file format for the tasks. Part 11 is nowadays an online dictionary for the presentation layer, which is mainly used for part 10. Part 13 specifies the file server and the protocol for using it.

Guidance-related material can be found in part 7, which introduces 'remote control' messages for the tractor and some implementation issues pertaining to remote control messages. With the remote control message, an implement may command the curvature of the tractor to the desired value; it is up to the tractor's internal control system to operate the tractor's steering hydraulic cylinder in a manner that realises the setpoint. The other message gives feedback to an implement about the estimated or measured curvature. Similar messages are also given for the speed command and measurement.

However, it is not mandatory that a tractor manufacturer implement the remote control messages. Part 9 specifies tractor classes, from 1 to 3; only in Class 3 are the remote control messages required, but guidance remote control is still an option (ISO 11783-9:2002).

Even if the remote control for a tractor's curvature is available in a standard format, it will only cover one crucial link in the combined guidance system. The missing links are between the guidance system and the implement and the state information interchange between the tractor and the implement. Hence, when it comes to the combined guidance system, the ISO 11783 is considered more of a limiting factor than a supporting feature. The standard needs to be improved to support a combined guidance system.

3. Modular Agricultural Machines

In this thesis, the proposed navigation system and the methods related to it are intended to be generic, i.e. the methods are not restricted to only a certain vehicle configuration. However, in order to be credible, the methods are tested on an experimental navigation system with real-scale agricultural machinery.

This chapter describes the vehicle configurations and positioning devices that are used in the tests. Also, the real world disturbances that interfere with them are discussed. Finally, a mathematical model of the system is derived.

3.1 Tractor-trailer system

A tractor with a single implement is the most common vehicle configuration in agricultural field operations, even if some machines are self-propelled. The agricultural implement can be either a front- or rear-mounted implement or it can be towed. Because the kinematic model for the mounted implement does not differ from the tractor alone, a towed implement was selected for the purposes of this thesis. Two different vehicle configurations were used in this thesis; the second configuration is an improved version with a new tractor generation.

For vehicle configuration A, the tractor was a Valtra T190, equipped with an ISO 11783 compatible Tractor ECU that fulfils Class 3 requirements with guidance option. Thus, the remote control messages in part 7 of the ISO 11783 standard could be used to steer the tractor. The steering actuator was the same one that the manufacturer uses in its commercial tractor guidance system and the calibration of the steering servo system was carried out by the manufacturer. The implement was a Junkkari Maestro 3000 combined seed drill equipped with an ISO 11783 compatible implement controller. Combined seed drills are used to sow both seeds and local fertilizer. The implement was a trailer-type seed drill; the supporting wheels were located in the rear and the coulters just in front of them. For implement steering purposes, the drawbar of the combined seed drill was modified by adding an articulated joint to the end of the seed drill. The articulated joint gave an

additional degree of freedom for guiding the vehicle. The prototype modification was done together with MTT Agrifood Research Finland and the manufacturer of the combined seed drill. Figure 3.1 shows vehicle configuration A.



Figure 3.1. Vehicle configuration A: Valtra T190 and Junkkari Maestro 3000.

For vehicle configuration B, the tractor was a Valtra T132, which was also equipped with an ISO 11783 compatible Tractor ECU and a Class 3 guidance option. In this case, the continuous variable transmission (CVT) of the tractor made it possible to control the speed through the ISO 11783 network by using remote control messages. The implement was the same as in configuration A. Hence, the combined guidance system had three actuators under its control: the steering angle of the front wheels of the tractor, the angle of the articulated drawbar of the combined seed drill and the speed of the tractor. Figure 3.2 shows vehicle configuration B.



Figure 3.2. Vehicle configuration B: Valtra T132 and Junkkari Maestro 3000

In both configurations, the implement was connected to the rear hitch of the tractor with a packer. The trailer drawbar was attached to the packer, which moves the free joint from the tractor draw hook backwards. The packer is used to compact the soil that is left between the tractor tracks in order to provide an even surface for the seed coulters and enable a uniform

growth of crops. The packer also assists with manoeuvring in the headlands since it allows for a larger joint angle compared to the draw hook connection. The free joint angle was measured with a potentiometer-type measuring device (Figure 3.3, right). The joint in the seed drill end of the drawbar was hydraulically controlled and the angle of the controlled joint angle was measured using a non-contact inductive position sensor (GILL Sensors) (Figure 3.3, left). The nonlinearity of the non-contact inductive position sensor was compensated for. The sensors were selected based on angle of movement, amount of backlash in the joint and based on previous experience of selected sensors. The hydraulic cylinders were connected to the tractor auxiliary valves and the position sensor information was transmitted to the ISO 11783 network.



Figure 3.3. The articulated joint at the end of the drawbar (left), and the measurement of the freely moving joint (right).

3.2 Positioning and guidance devices

Two types of positioning devices were used in this thesis: global and local measurements. The tractor position was measured using a GPS based receiver and the orientation using an Inertial Measurement Unit (IMU). The trailer and the obstacle positions were measured with 2D laser scanners.

In this thesis, two different global positioning measurement systems were used. In measurement system I, a Trimble 5700 GPS was used as an RTK-GPS receiver together with a virtual reference station (VRS-GPS). The orientation was measured with an Xsens MT9-B IMU, which was also used to improve the course measurement of the RTK-GPS receiver by means of pose estimation.

In measurement system II, the GPS was the same as in measurement system I. The attitude (roll and pitch angles) was measured using an Inertial-Link 3DM-GX2 IMU. The heading measurement of the RTK-GPS receiver was improved with filtering method together with a Fibre Optic Gyro (FOG), the KVH DSP-3000. The devices were packed in a single box and connected together, and the information was merged using an embedded controller. The combined information was sent to the ISO 11783

network using standard NMEA 2000 messages. Figure 3.4 shows measurement system II, whereas the mounting of the measurement box on the roof of the tractor's cabin can be seen in Figure 3.2.



Figure 3.4. In measurement system II, the position and orientation of the tractor was measured with a Real Time Kinematic GPS, a Fibre Optic Gyro and an Inertial Measurement Unit, which were packed together.

The guidance system relied not only upon the GPS positioning and heading measurements; additionally, a local positioning device was used to measure the distance to the previous swath on the implement side. In the seed drill, a sensor measured the relative distance to the adjacent swath based on a mark drawn on the soil surface. The sensor was based on a SICK LMS221 laser scanner and a small plough that makes an identifiable furrow in the topsoil next to the swath (Figure 3.5). The laser scanner was mounted in right front corner of the seed drill at a height of one metre above the soil surface and the plough was mounted on the right side of the following harrow.



Figure 3.5. A laser scanner measured the profile of the ground (left). A furrow was produced at the edge of the sown area by a small plough, which was mounted on the following harrow of the seed drill (right).

Also, the obstacles were measured using a SICK LMS 221 laser scanner. The scanner was mounted on the front of the tractor and it scanned the

front area horizontally. Figure 3.6 shows the obstacle measurement system, which can also be seen in Figure 3.2.



Figure 3.6. A laser scanner was used to detect the obstacles in front of the tractor.

In the test configuration, the guidance system controller was based on a standard desktop computer (Intel DG45FC motherboard, Intel Core 2 Duo E8600 processor, 2GB memory and Kvaser LeafLight HS CAN interface). The guidance system controller was connected to an ISO 11783 network and WLAN. The ISO 11783 network was the main interface for the guidance system controller, while WLAN interface was used to connect the laser scanners and a GUI to the guidance system.

3.3 Real world disturbances

Ideally, all of the state variables should be measurable and the kinematic equations would describe the behaviour of the system perfectly. Unfortunately, this is only possible in simulations, whereas in the real world all kinds of disturbances affect the measurements.

The position of the tractor and the trailer as well as the realised steering angle and the speed can be directly measured. However, these measures include disturbances that are not pure Gaussian white noise. Even the most accurate non-military GPS receivers with a correction signal have a roaming error of a few centimetres. This error cannot be filtered out without using any external local measurements (Oksanen et al. 2005).

Common sources of errors in GPS positioning are ionospheric and tropospheric effects, errors in satellite orbits and clocks, multipath effects, receiver noise and clock errors, and calculation errors (Kaplan and Hegarty 2005). Also, the geometry and the number GPS satellites cause GPS errors. These errors can be attributed to various factors. The total error at the time instant is a sum of all of these errors. Different real-time correction methods can reduce these errors, but they can hardly eliminate all of the error sources, thus complicating the structure of the GPS noise.

A moving tractor in a changing environment makes it difficult to establish the GPS noise definition. This is because a static performance by the receivers might not be indicative of a dynamic performance when determining the GPS positioning quality (Stombaugh et al., 2002). Pirti (2008) found that tree canopy on one side of the field increased the standard deviation by approximately 40% for both the baselines and height differences. Min et al. (2008) conducted dynamic GPS tests in citrus orchards. They found that receivers performed differently under various test and orchard conditions. Also, the type of receiver and the mounting height significantly affected the accuracies.

There are only a few GPS error generation studies dealing with GPS output messages. Rankin (1994) constructed a simulator that models the error statistics for various receivers. The simulator included a model for GPS satellite orbits, which were used to create dilution of precision (DOP) values that translated pseudo range errors into XYZ errors; the simulator then outputted these errors. Oksanen et al. (2005) presented a noise model for low-cost GPS positioning. For guidance algorithm testing and development, there is a need for a more realistic and controllable noise model.

A kinematic model of the vehicle can be constructed by assuming that the ground is ideal and that no slipping will affect the system. This is not a true assumption in the real world. Oftentimes the ground is not flat and homogenous. A tractor-trailer system does not follow the kinematic route, and especially in the curves, the difference can be remarkable. Also, the slip is remarkable in field conditions compared to those for a vehicle driving on hard terrain. In this thesis, the difference is compensated for via the slipping factor. This factor cannot be directly measured or tuned beforehand, and for that reason, it must be estimated continuously.

Lenain et al. (2006) have studied slip modelling for bicycle kinematic models, whereas Karkee and Steward (2011) have studied slip modelling and model parameter estimation for dynamic models. Modelling the slip is extremely challenging in the case of agricultural fields since the properties of the field vary to a significant degree. The soil and terrain properties, including the slope, soil type and soil moisture, are all key factors affecting the slip. Also, the varying parameters in a tractor-trailer system affect the slip; this includes weight change, tire pressure, weight distribution, the amount of additional counterweight installed on the tractor and the up/down state of the implement. If it is assumed that the parameters do not change, then it would be possible to estimate all of the required parameters for the dynamic model with a reasonable degree of accuracy from multiple sets of field trajectories offline (Karkee and Steward, 2011). However, in

In the existing literature (e.g. Wong, 2008, pp. 30-39), tyre slipping is usually modelled by introducing a slip angle, which is added to the steer angle: $\alpha = \alpha_t + \alpha_{slip}$, where α is the effective steering angle. Then, the slip angle, α_{slip} , would constantly change, especially for curves. If there are no external forces pulling or pushing the tractor sideways, it can be assumed that the slipping is caused by inertia and that the slipping angle is approximately relative to the steering angle. By introducing the slipping factor, the slip angle can be calculated relatively well based on the steer angle: $\alpha_{slip} = (\delta - 1)\alpha_t$, if the steering angle is not equal to zero.

Because both the state estimation and the controller work in discrete time, Equation 3.1 is discretised using Euler's approximation:

$$\begin{bmatrix} x_R(t_{k+1}) \\ y_R(t_{k+1}) \\ \theta(t_{k+1}) \\ \delta(t_{k+1}) \end{bmatrix} = \begin{bmatrix} x_R(t_k) + v_t(t_k) \cos \theta(t_k) T \\ y_R(t_k) + v_t(t_k) \sin \theta(t_k) T \\ \theta(t_k) + v_t(t_k) \frac{\tan \delta(t_k) \alpha_t(t_k)}{a} T \\ \delta(t_k) \end{bmatrix}, \quad (3.2)$$

where the sampling period is constant: $t_{k+1} - t_k = T$. A $T = 100\text{ms}$ sampling period was used for the purposes of this study.

Because the tractor actuators are not infinitely fast, the dynamics of those are modelled. Similar to Karkee and Steward (2010) and Werner et al. (2012), a simple first-order dynamics was used. It is assumed that the actuators will eventually be able to realise the desired control values and that no steady state errors will be present. The realised control values or the states where the actuators are currently located are modelled according to the following equation:

$$\begin{bmatrix} v_t(t_{k+1}) \\ \alpha_t(t_{k+1}) \end{bmatrix} = \begin{bmatrix} k_v v_t(t_k) + (1 - k_v) v_d(t_k) \\ k_\alpha \alpha_t(t_k) + (1 - k_\alpha) \alpha_d(t_k) \end{bmatrix}, \quad (3.3)$$

where k_v and k_α are the dynamic coefficient parameters and v_d and α_d are the desired control values, which are fed to the actuators, the desired speed and steering angle, respectively.

If it is assumed that the trailer does not slide sideways, then the kinematic behaviour of the trailer can be modelled using only the angle between the trailer and the tractor. The derivation is explained in PUB III, and the final form of the differential equation for the freely moving joint angle is as follows:

$$\dot{\beta} = \frac{-av_t \sin(\beta + \gamma_t) + v_t(d + c \cos \beta + b \cos(\beta + \gamma_t)) \tan \alpha_t - ad\dot{\gamma}_t}{a(d + c \cos \gamma_t)}, \quad (3.4)$$

where β is the angle between the tractor and the trailer, γ_t is the realised angle of the controlled joint, d is the distance to the seed coulters from the

drawbar, c is the length of the drawbar and b is the distance to the attachment point from the rear axle. The differential equation is again discretised using Euler's approximation:

$$\beta(t_{k+1}) = \beta(t_k) + \dot{\beta}(t_k)T. \quad (3.5)$$

The realised control value is modelled using a first-order dynamic model:

$$\gamma_t(t_{k+1}) = k_\gamma \gamma_t(t_k) + (1 - k_\gamma) \gamma_d(t_k), \quad (3.6)$$

where k_γ is the filter coefficient and γ_d is the desired joint angle.

There are also auxiliary states for the optimisation and estimation process. The centre position of the implement is needed for the cost function in the NMPC method. The implement position is calculated as follows:

$$\begin{bmatrix} x_E \\ y_E \end{bmatrix} = \begin{bmatrix} x_R - b \cos \theta - c \cos(\beta - \theta) - d \cos(\beta + \gamma_{actual} - \theta) \\ y_R - b \sin \theta + c \sin(\beta - \theta) + d \sin(\beta + \gamma_{actual} - \theta) \end{bmatrix}, \quad (3.7)$$

where (x_E, y_E) is the centre position of the implement and d is the distance from the drawbar.

The location of the laser scanner and the marking plough are needed to locally correct the position of the vehicle with the laser scanner measurement. The position of the laser scanner is calculated as follows:

$$\begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} x_E + l_x \cos(\beta + \gamma_{actual} - \theta) - l_y \sin(\beta + \gamma_{actual} - \theta) \\ y_E - l_x \sin(\beta + \gamma_{actual} - \theta) - l_y \cos(\beta + \gamma_{actual} - \theta) \end{bmatrix}, \quad (3.8)$$

where (x_L, y_L) is the position of the laser scanner, l_x is the axial and l_y is the cross-axial distance from the centre position of the implement. The position of the marking plough is calculated as follows:

$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = \begin{bmatrix} x_E + p_x \cos(\beta + \gamma_{actual} - \theta) - p_y \sin(\beta + \gamma_{actual} - \theta) \\ y_E - p_x \sin(\beta + \gamma_{actual} - \theta) - p_y \cos(\beta + \gamma_{actual} - \theta) \end{bmatrix}, \quad (3.9)$$

where (x_P, y_P) is the position of the marking plough, p_x is the axial and p_y is the cross-axial distance from the centre position of the implement.

The resulting state vector for the overall kinematic model is as follows:

$$x = [x_R \ y_R \ \theta \ \delta \ v_t \ \alpha_t \ \beta \ \gamma_t \ \dot{\gamma}_t \ x_E \ y_E]^T, \quad (3.10)$$

and the overall resulting control vector is as follows:

$$u = [v_d \ \alpha_d \ \gamma_d]^T. \quad (3.11)$$

Subsequently, the system model is given as follows:

$$x(t_{k+1}) = f(x(t_k), u(t_k)), \quad (3.12)$$

where function f includes Equations 3.2, 3.3 and 3.5-3.7.

Because the derivative of the joint angle is needed in Equation 3.5, the optimised control value is actually \dot{u} , and u is obtained via integration. This also makes it possible to limit the derivatives of the control values without numerical derivations.

4. Methods

In this chapter, the methods used for the distributed navigation system of an agricultural machine combination are described. First, the Nonlinear Model Predictive Path Tracking method and the extension to the obstacle avoidance are described. The path tracking method needs a feasible path to follow and an accurate state estimate of the current state of the system. The methods for realising them are described next. Finally, the complete physical and software structures of the navigation system, which consists of the previously mentioned parts, are presented. The advanced navigation system, which consists of many parts and a number of parameters, is challenging to develop and tune. Therefore, a simulator and parameter estimation methods are necessary tools when developing the navigation system. Those tools are presented last, before the results are explained.

4.1 Nonlinear Model Predictive Path Tracking

The basic idea of the NMPC method is to predict the future states of the system and to minimise the given cost function. The future states of the system are predicted using a mathematical model of the controlled system (Equation 3.12) by applying the control values to the system model in an open-loop manner. The cost function is a weighted quadratic sum of the state and the control values. The general form of the cost function at time t_k is as follows:

$$\begin{aligned} J(x, \dot{u} | t_k) &= \sum_{j=1}^M \|x(t_{k+j}|t_k) - r_x(t_{k+j})\|_Q^2 \\ &+ \sum_{j=1}^M \|u(t_{k+j}|t_k) - r_u(t_{k+j})\|_{R_u}^2 + \sum_{j=1}^M \|\dot{u}(t_{k+j}|t_k)\|_{R_{\dot{u}}}^2, \end{aligned} \quad (4.1)$$

where M is the prediction horizon size, $x(t_{k+j}|t_k)$ is the predicted state for future time t_{k+j} at time t_k , r_x is the reference trajectory for the predicted state and r_u is the reference trajectory for the controls. In the cost function, Q , R_u and $R_{\dot{u}}$ are symmetric, positive, semi-definite weighting matrices. The

optimisation problem can then be formulated to find the sequence of controls, such that

$$\dot{u}^*(t_k \cdots t_{k+M} | t_k) = \underset{\dot{u}}{\operatorname{argmin}} J(x, \dot{u} | t_k), \quad (4.2)$$

where $\dot{u}^*(t_k \cdots t_{k+M} | t_k)$ is the sequence of the optimal control values at time t_k . Only the first control values are used for the actual control and the optimisation is repeated with the new state estimates at time t_{k+1} . The constraints for the optimisation problem are the system model and the constraints for the states and control values as follows:

$$\begin{aligned} x(t_{k+j+1} | t_k) &= f(x(t_{k+j} | t_k), u(t_{k+j} | t_k)) \\ u(t_{k+j+1} | t_k) &= u(t_{k+j} | t_k) + \dot{u}(t_{k+j} | t_k)T \\ x_{min} &\leq x(t) \leq x_{max}, \forall t \in (t_k, t_{k+M}) \\ u_{min} &\leq u(t) \leq u_{max}, \forall t \in (t_k, t_{k+M}) \\ \dot{u}_{min} &\leq \dot{u}(t) \leq \dot{u}_{max}, \forall t \in (t_k, t_{k+M}), \end{aligned} \quad (4.3)$$

where x_{min} and x_{max} are the minimum and maximum values of the states, u_{min} and u_{max} are the minimum and maximum values of the control values, and \dot{u}_{min} and \dot{u}_{max} are the maximal decreases and the maximal increases of the control values.

The NMPC method defines the nonlinear dynamic optimisation problem presented in Equations 4.1, 4.2 and 4.3 but not how it is solved. Betts (2001) described some practical numerical methods that are proven to work in real applications. Those methods are described next.

There are two types of numerical iterative methods available: indirect and direct methods. With the indirect method, a root of the necessary optimisation condition is searched, i.e. $F'(z) = 0$, where $F(z)$ is the minimised function. With the direct method, a sequence of points, z_1, z_2, \dots, z^* , is constructed such that the objective function is minimised and typically $F(z_1) > F(z_2) > \dots > F(z^*)$. There are advantages in both strategies, but the direct method is more popular because it is numerically more robust and easier to initialise.

The minimised function, F , is not the same as the state cost, J , because of the constraints. The classical approach to solving constrained optimisation problem is to use the Lagrangian function:

$$L(z, \lambda) = J(z) - \lambda^T c(z) = J(z) - \sum_{i=1}^m \lambda_i c_i(z), \quad (4.4)$$

where λ is the Lagrange multipliers, $c(z)$ is the equality constraints and z includes the optimised variables ($u(t_k \cdots t_{k+M})$). The optimality requires that the derivatives of L with respect to both z and λ be zero. Moreover, the

roots of the derivatives of L are the only necessary conditions for the optimality, but they do not distinguish between minimum, maximum or stationary points. For that reason, a sufficient condition is that the Hessian matrix of the Lagrangian function also needs to be positive.

The minimum value of L can be found using Newton's method, i.e. a Taylor series expansion about z and λ of the gradients of L with respect to z and λ are constructed and set to zero:

$$\begin{aligned} 0 &= g - G^T \lambda + H_L(\bar{z} - z) - G^T(\bar{\lambda} - \lambda) \\ 0 &= -c - G(\bar{z} - z), \end{aligned} \quad (4.5)$$

where g is the gradient vector of J with respect to z , G is the Jacobian matrix of c with respect to z , H_L is the Hessian matrix of the Lagrangian function and \bar{z} and $\bar{\lambda}$ are the new optimal variable candidates. After simplification, the Karush-Kuhn-Tucker (KKT) system can be formulated as follows:

$$\begin{bmatrix} H_L & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} -p \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} g \\ c \end{bmatrix}, \quad (4.6)$$

where p is the new search direction: $\bar{z} = z + p$. Any reasonable numerical method can be used to solve p and $\bar{\lambda}$ from the KKT system. However, in practice it is important that the method is efficient and numerically stable.

If the minimised objective function, J , is quadratic and the equality constraints, c , are linear, then Newton's method will converge in one step. An important special form of the optimisation problem is Quadratic Programming (QP), where the objective is quadratic (as in Equation 4.1) and the equality and inequality constraints are linear:

$$\begin{aligned} Az &= a \\ Bz &\geq b. \end{aligned} \quad (4.7)$$

When using the Lagrangian function, only the equality constraints are present. However, in Equation 4.7 (and in Equation 4.3) there are also inequality constraints. The solution is to use an Active Set strategy. In the Active Set strategy, the full optimisation step is not used after solving the KKT system. Instead, the variables are searched for from the search direction ($\bar{z} = z + \zeta p$), such that a maximum step (ζ) is taken with respect to the inactive inequality constraints. After this step, either one of the inactive inequality constraints will be an active equality constraint or else a full step ($\zeta = 1$) is taken. If a full step is taken, then the sign of the Lagrangian multipliers are checked, and if the sign of some multiplier is incorrect, then the corresponding equality constraint is taken away from the active set and it will once again become an inequality constraint. If

there are no changes to the active set after taking the full step, then the algorithm is ended.

In the QP, the constraints must be linear. However, there are also nonlinear constraints in Equation 4.3. A Sequential Quadratic Programming (SQP) algorithm is among the most widely used algorithms for solving the nonlinear optimisation problem. With the SQP algorithm, the fundamental idea is to solve QP sub-problems sequentially, where in each SQP iteration the objective function is approximated quadratically and the constraints are approximated linearly.

With the NMPC method, re-initialising the optimisation problem using the optimal solution proposed in the previous step yields a good approximation of the optimal solution with only a minor number of iterations. However, the matrices of the linear system quickly become quite large, especially if the system model has several states and the optimisation horizon is long. Then, the sparsity of the matrices can be exploited and more efficient methods than direct matrix factorisation should be used in the QP algorithm. Also, because the QP algorithm is used to approximate the nonlinearly constrained optimisation problem, different kinds of merit functions can be used to modify the step length of the line search.

4.1.1 From trajectory tracking to path tracking

The most natural and easy way to use the NMPC method is to implement a trajectory tracking controller. In trajectory tracking, a desired trajectory is explicitly used for the cost function (Equation 4.1). In this thesis, however, the goal was to study path tracking methods. The first part of the cost function (state penalty) is illustrated in Figure 4.1. The difference between the control points and the desired path is penalised. Therefore, the trajectories r_x and r_u are not constant and fixed to the time. The $r_x(t_{k+j})$ is calculated at each SQP iteration such that it minimises the distance of $x(t_{k+j}|t_k)$ from the path. The $r_u(t_{k+j})$ is calculated such that it corresponds to the position of the $r_x(t_{k+j})$ along the path when the tracking error is zero. The idea of how to transform a trajectory tracking controller to a path tracking controller was originally suggested by Hauser and Hindmard (1995). The idea has been utilized later with different controllers for example in Encarnacao and Pascoal (2001). However, it has not been applied before to NMPC.

The reference trajectories are calculated separately for both the tractor and the trailer. In the following equations, the trajectory is calculated for the tractor. The same equations hold true for the trailer, but the state components ($x_{\{x_R, y_R\}}$) are changed to correspond to the trailer's state ($x_{\{x_E, y_E\}}$).

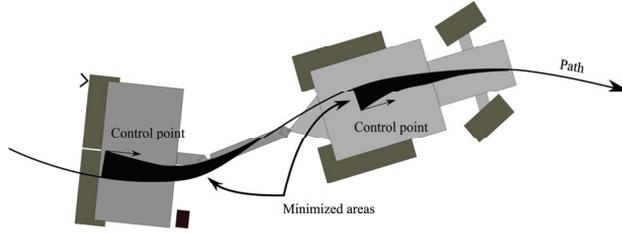


Figure 4.1. An illustration of the state penalty in the NMPC cost function.

The target path is modelled as a polyline. The distance between consecutive points along the path is considered to be constant. Also, the orientation of the tractor along the path and the desired velocity and steering angles (i.e. the steady state controls) are incorporated into the path points. First, the path point (P_i) that is closest to the current state, $x_{\{x_R, y_R\}}(t_{k+j}|t_k)$, is searched for:

$$i = \underset{i}{\operatorname{argmin}} \|x_{\{x_R, y_R\}}(t_{k+j}|t_k) - P_i\|^2. \quad (4.8)$$

During the searching procedure, it is assumed that the local minimum, which is found closest to the minimum of the previous time step, is also the global minimum.

Then, the distances of the state from the line $P_{i-1} - P_i$ (Case 1 in Figure 4.2) and from the line $P_i - P_{i+1}$ (Case 2 in Figure 4.2) are calculated. In addition, the corresponding closest points and the derivatives of the distances with respect to the state are also calculated.

To clarify the equations, the following shorthand notations are used: In Case 2 point X corresponds to the state, $x_{\{x_R, y_R\}}(t_{k+j}|t_k)$, A corresponds to P_i and B corresponds to P_{i+1} . The notation AX is the line segment from A to X , BX is the line segment from B to X and AB is the line segment from A to B . Finally, the subscript, $\cdot_{\{x\}}$ or $\cdot_{\{y\}}$, represents the x or y component of the corresponding line segment or point. With these shorthand notations, the distance (d_+) of X from AB is as follows:

$$d_+ = \frac{X_{\{x\}} \cdot (AB)_{\{y\}} - B_{\{x\}} \cdot (AX)_{\{y\}} + A_{\{x\}} \cdot (BX)_{\{y\}}}{\sqrt{\|AB\|^2}}. \quad (4.9)$$

Equally, the position of the closest point (r_x) along line segment AB is as follows:

$$t_+ = \frac{(AX)_{\{x\}} \cdot (AB)_{\{x\}} + (AX)_{\{y\}} \cdot (AB)_{\{y\}}}{\|AB\|^2}, \quad (4.10)$$

where the value $t_+ = 0$ means that $r_x = A$, and the value $t_+ = 1$ means that $r_x = B$.

The derivatives of the distance, d_+ , with respect to the state are as follows:

$$\frac{\delta d_+}{\delta X_x} = \frac{(AB)_y}{2\sqrt{\|AB\|^2}} \quad (4.11)$$

and

$$\frac{\delta d_+}{\delta X_y} = \frac{-(AB)_x}{2\sqrt{\|AB\|^2}} \quad (4.12)$$

To calculate the distances (d_- and t_-) from the line, $P_{i-1} - P_i$ (Case 1), the shorthand notations are changed such that B corresponds to P_{i-1} . Because the order of path points changes, the sign of the distance d_- and the corresponding derivatives also change and must be compensated for in the equations.

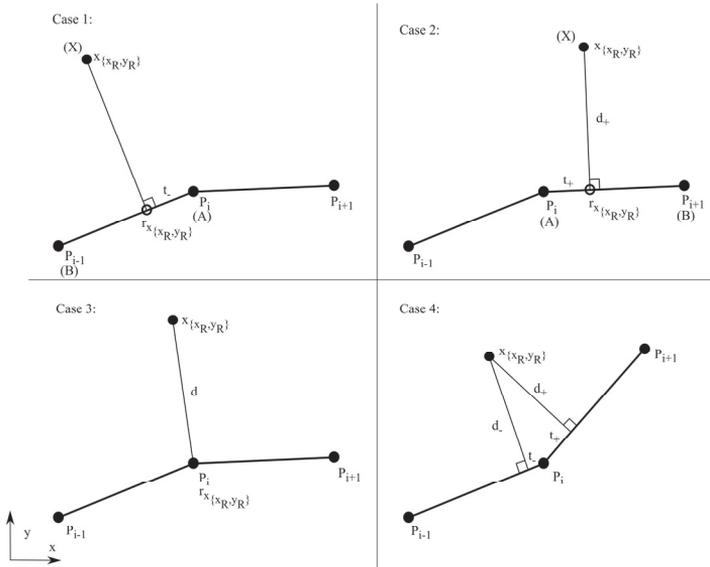


Figure 4.2. Four different cases used to calculate the trajectory point corresponding to the current state: in Case 1, position X is closest to the line $P_{i-1} - P_i$; in Case 2, position X is closest to the line $P_i - P_{i+1}$; in Case 3, position X is closest to path position P_i ; and in Case 4, the inner curve.

If the distances between the path points are equal, then four different cases will be used to calculate the closest position from the path (Figure 4.2):

Case 1: $(t_- \in (0,1) \wedge t_+ < 0)$

Position X is closest to the line segment $P_{i-1} - P_i$. d_- and the corresponding derivatives can be directly used in the cost function;

Case 2: $(t_+ \in (0,1) \wedge t_- < 0)$

Position X is closest to the line segment $P_i - P_{i+1}$. d_+ and the corresponding derivatives can be directly used in the cost function;

Case 3: $(t_- < 0 \wedge t_+ < 0)$

Position X is closest to path point P_i . Path point P_i is used as the trajectory point r_x ;

Case 4: $(t_+ \in (0,1) \wedge t_- \in (0,1))$

The inner curve. The weighted average of d_+ and d_- is used; this is done by using t_+ and t_- as weighting factors.

In Case 4, the weighting makes the path smoother, though the weighted distance to the path is not equal to the actual shortest distance.

In Cases 1, 2 and 4, there is no need to explicitly calculate the trajectory point, r_x , since that part of the cost function can be written as follows:

$$\left\| x_{\{x_R, y_R\}}(t_{k+j}|t_k) - r_{x_{\{x_R, y_R\}}}(t_{k+j}) \right\|_{Q_{\{x_R, y_R\}}}^2 = d^2 Q_{\{x_R, y_R\}}, \quad (4.13)$$

where $Q_{\{x_R, y_R\}}$ is the weighting factor of the corresponding states. And the partial derivatives of the cost with respect to the state elements are as follows:

$$\frac{\delta J(t_k)}{\delta x_{\{x_R\}}(t_{k+j}|t_k)} = 2d Q_{\{x_R\}} \frac{\delta d}{\delta x_{\{x_R\}}(t_{k+j}|t_k)} \quad (4.14)$$

and

$$\frac{\delta J(t_k)}{\delta x_{\{y_R\}}(t_{k+j}|t_k)} = 2d Q_{\{y_R\}} \frac{\delta d}{\delta x_{\{y_R\}}(t_{k+j}|t_k)}. \quad (4.15)$$

The partial derivatives on the right-hand side in Equations 4.14 and 4.15 are calculated in Equations 4.11 and 4.12 using the shorthand notations.

4.1.2 Collision avoidance

The collision avoidance problem can be divided into two different sub-problems: detecting the obstacle and bypassing the obstacle. In this section, the obstacle detection method is described first. Then, the modified path tracking algorithm is explained, including the collision avoidance problem. In Finnish fields, one of the most common obstacles is an electricity pole, and therefore the attention is on pole-type obstacles.

The obstacle detection method is based on a 2D laser scanner (SICK LMS221). The scanner is mounted on the front of the tractor and it scans the front area horizontally. The raw measurements consist of 181 distance

measurements with one degree of resolution. The raw measurements are at first transformed into Cartesian coordinates. The obstacles are detected in the transformed measurements using a clustering method. The whole clustering process is illustrated in Figure 4.3. The initial positions of the clusters are obtained from the set of known obstacles. If there is a known obstacle in the map that should be in sight of the scanner, a corresponding cluster is initialized. The same is done for every obstacle that is in sight of the scanner. In Figure 4.4, the blue cross represents the centre of the known obstacle and the initial cluster centre.

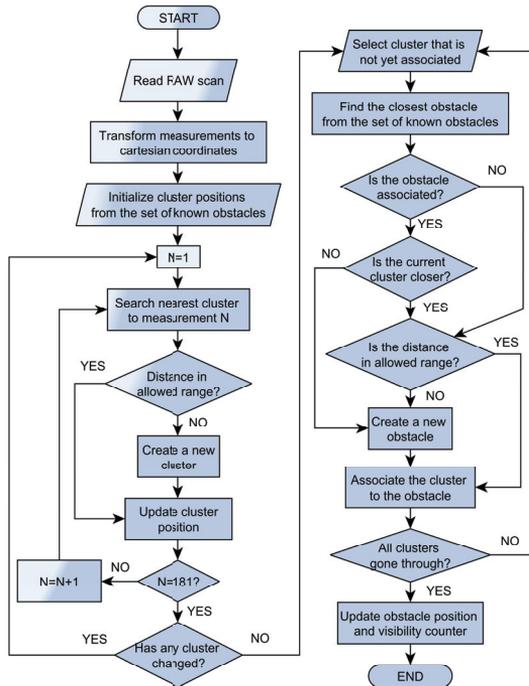


Figure 4.3. The clustering algorithm.

The clustering algorithm is based on a modified nearest neighbour algorithm. First, all of the measurements are passed through one by one and the closest existing cluster for every measurement is found. If the distance to the closest cluster is within the limits permitted, then the point is associated with the cluster and the cluster position is updated according to the following equation:

$$\begin{cases} x_{cluster} = \frac{C * x_{cluster} + x_{meas}}{C + 1} \\ y_{cluster} = \frac{C * y_{cluster} + y_{meas}}{C + 1} \end{cases} \quad (4.16)$$

where $(x_{cluster}, y_{cluster})$ is the old position of the cluster centre, (x_{meas}, y_{meas}) is the measurement and C is the number of measurements associated with the cluster before the current measurement. In Figure 4.4, four adjacent scanner measurements are clustered together and marked with a red circle.

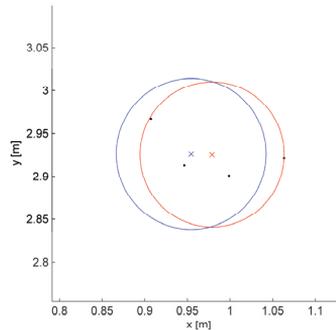


Figure 4.4. Obstacle detection based on laser scanner data (black dots). The red circle represents the clustered measurements and the blue circle is the previous position of the associated obstacle.

If the distance to the closest cluster is not within the limits allowed, then a new cluster is created.

After all the measurements are associated with some cluster, the number of associated cluster measurements (C) is reduced to one and the clustering is repeated. The process is continued until all of the measurements are associated with the same cluster, as in the previous iteration, or until the maximum number of iterations is reached.

After clustering the measurements, the clusters are matched back to the known obstacles. All obstacles that are within sight of the laser scanner are studied once again. The closest cluster to each existing obstacle is searched for and associated with that particular obstacle. The positions of the associated obstacles are updated using the same equation that was used for the cluster positions (Equation 4.16). If there are new clusters, i.e. the cluster was not close enough to any of the existing obstacles, a new obstacle candidate is created. In Figure 4.4, the cluster marked with a red cross and a circle is associated with a known obstacle marked with the colour blue.

An additional visibility counter is also updated. If the obstacle is within sight of the scanner and it can be seen, then the visibility counter is increased; otherwise, the counter is decreased. When the obstacle is seen a predetermined number of times, the recognition of that particular obstacle is considered to be reliable. In turn, if the obstacle is not seen, even though it should have been seen, then the recognition is considered unreliable and the obstacle is removed from the set of known obstacles.

There are different ways to include the object avoidance problem within the NMPC method. One way is to add additional constraints to the state values. Another way is to add an additional cost from the obstacles, or simply to modify the reference trajectory so that it goes past the obstacle.

In this thesis, the option of modifying the cost function was chosen. Neither the underlying path tracking cost function nor the reference trajectory has been changed, but the cost from state has been modified. This is because of the calculation capacity and the possibility that the obstacles could move.

The original cost of the tractor position can be calculated as follows:

$$J_{\{x_R, y_R\}}(t_{k+j}|t_k) = \left\| x_{\{x_R, y_R\}}(t_{k+j}|t_k) - r_{x_{\{x_R, y_R\}}}(t_{k+j}) \right\|_{Q_{\{x_R, y_R\}}}^2. \quad (4.17)$$

When the reference trajectory, $r_{x_{\{x_R, y_R\}}}$, is close to an obstacle, it cannot be followed without colliding with the obstacle. Therefore, it is irrelevant to keep the cost out of the reference trajectory. Instead, a cost that ensures that the vehicle will drive past the obstacle should be added. The area where the original cost is changed into the avoiding cost is illustrated in Figure 4.5.

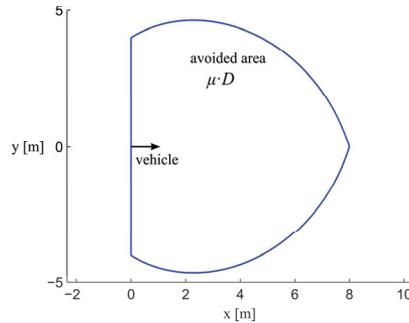


Figure 4.5. The avoided area where the obstacles are not allowed when the avoiding distance, D , is set at 8 metres.

As seen in Figure 4.5, the avoided area is not a circle around the vehicle. The obstacle can be closer to the side of the vehicle. The coefficient, which varies according to the angle between the vehicle and the obstacle, is as follows:

$$\mu = \begin{cases} \frac{\pi - |\theta - \text{atan2}(OV)|}{\pi} & , \text{ if } |\theta - \text{atan2}(OV)| < \frac{\pi}{2}, \\ 0 & \text{otherwise} \end{cases} \quad (4.18)$$

where θ is the current heading angle and $\text{atan2}(OV)$ is the direction to the obstacle. By using this coefficient and nominal avoiding distance, D , the distance from the obstacle to the edge of the avoided area can be calculated according to the following equation:

$$\varepsilon = (r + \mu \cdot D) - |OV|, \quad (4.19)$$

where $|OV|$ is the distance between the obstacle and the vehicle and r is the radius of the obstacle. The variables used in the equation are shown in more detail in Figure 4.6.

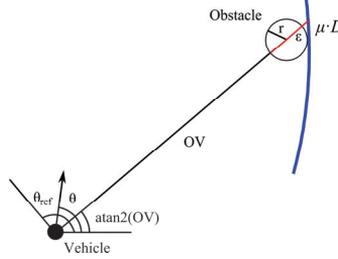


Figure 4.6. Calculation of the distance from the obstacle to the edge of the avoided area within the avoided area.

The calculated distance to the edge of the avoided area is used in the cost function when the obstacle is within the avoided area or the obstacle is closer to the avoided area than the vehicle is to the original reference trajectory.

Using these definitions, the cost of the tractor position can be changed to:

$$J_{\{x_R, y_R\}}(t_{k+j}|t_k) = \begin{cases} \|\varepsilon\|_Q^2 & , \text{ if } \Psi \\ \|\Delta x_{\{x_R, y_R\}}\|_Q^2 & \text{ otherwise} \end{cases}, \quad (4.20)$$

where $\Delta x_{\{x_R, y_R\}}$ is the original distance to the reference trajectory:

$$\Delta x_{\{x_R, y_R\}} = x_{\{x_R, y_R\}}(t_{k+j}|t_k) - r_{x_{\{x_R, y_R\}}}(t_{k+j}), \quad (4.21)$$

and Ψ is the boolean value whether the cost from the obstacle is used or not:

$$\Psi = \varepsilon > 0 \vee -\varepsilon < |\Delta x_{\{x_R, y_R\}}|. \quad (4.22)$$

Together with the cost of the position, the cost of the heading angle can also be changed to:

$$J_{\{\theta\}}(t_{k+j}|t_k) = \begin{cases} \|x_{\{\theta\}}(t_{k+j}|t_k) - \theta_{ref}\|_Q^2 & \text{ if } \Psi \\ \|x_{\{\theta\}}(t_{k+j}|t_k) - r_{x_{\{\theta\}}}(t_{k+j})\|_Q^2 & \text{ otherwise} \end{cases}, \quad (4.23)$$

where the new reference angle is calculated according to the following equation:

$$\theta_{ref} = \text{atan2}(OV) \pm \pi/2, \quad (4.24)$$

where the arithmetic operator sign is chosen to be the one giving a reference angle that is closest to the original heading angle.

In the above equations, the cost is calculated only based on one obstacle. If there are multiple obstacles within the avoided area, then the obstacle with the largest value for ε should be chosen. The same methods are also used for the cost of the trailer position.

4.2 Path planning

The feasibility of the target path is crucial for reducing the computational time of the NMPC method and for achieving a high degree of tracking accuracy. As a simplification, it is assumed that a field plot is convex. The convex area can be covered by drawing lines of equal width side by side and from one end to another end such that none of the lines will ever go outside the field. Such a field can easily be managed by driving to and fro parallel to the longest edge of the field. The remaining problem is to find feasible transitions between the different driving lines, i.e. the headland turnings, and to ensure that all of the driving lines are also feasible.

As discussed in Section 2.6, Dubins' Curves can be used to solve the problem of headland turnings. However, there are problems with the connection point between different sections of the turning lines; the curvature of the path is not continuous and therefore it is not feasible.

The fundamental idea is to use numerical lookup tables to quickly evaluate the path parameters, particularly the momentary centre points of the turning circles in the spirals used in the headland turnings. In this manner, slower evaluations of the Fresnel integrals (a common way to solve spiral points) can be avoided. Also, by using numerical lookup tables, the limit of the curvature derivative does not need to be constant and the resulting spiral does not necessarily need to be a Fresnel integral.

First, the idea of the connecting spirals is introduced in the next section. Then, Dubins' Curves are extended to support spirals between the arcs and lines. After that, the same idea is applied to smooth a predefined path, or in other words, to limit the curvature of the path. Finally, a completely simplified path planning algorithm for convex field plots is presented.

4.2.1 Spiral connection method

Dubins' Curves consist of six different turning types. These turnings consist of arcs with a maximum curvature and a straight line segment between the two arcs. At the junction point between different segments in the path, the curvature is discontinuous. To prevent these discontinuities, an extra connection segment is introduced between every original segment. The connection segments are constructed by driving the vehicle at a

constant speed and simultaneously turning the wheels from right to left at the maximum steering rate (Figure 4.7). The resulting trajectory represents transitions between any two momentary turning circles that the vehicle is capable of driving. The positions and headings of the vehicle and the centres of the turning circles are stored in a lookup table, from which the connection between two arbitrary turnings can be quickly be found. The input parameters for the spiral and lookup tables are presented in Algorithm 4.1.

Algorithm 4.1. CreateSpiral

Input:	α_{max} : maximum steering angle $\dot{\alpha}_{max}$: maximum steering rate wheelbase : the distance between the front and rear wheels dt : calculation resolution
Output:	$P(k)$: position (x,y) in a spiral corresponding to curvature k $\theta(k)$: heading corresponding to curvature k $O(k)$: centre (x,y) of turning circle corresponding to curvature k $r_1(k_1, k_2)$: distance from the turning circle $(k=k_1)$ to the path tangent $(k=k_2)$ $r_2(k_1, k_2)$: distance from the turning circle $(k=k_1)$ to the path tangent $(k=k_2)$

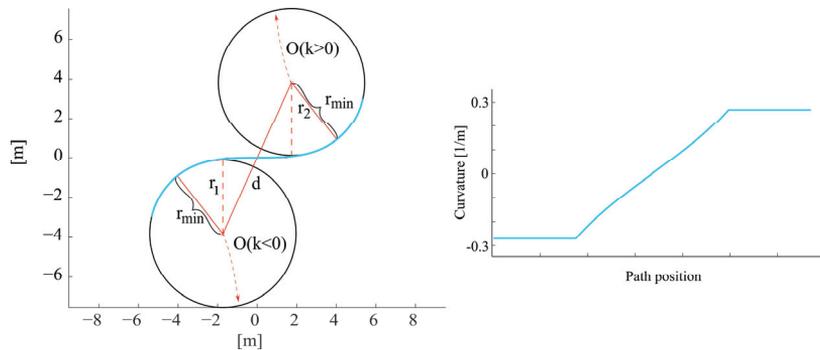


Figure 4.7. The vehicle is driven at a constant speed while simultaneously turning the wheels from the right to the left. The resulting trajectory is a spiral between two turning circles where the curvature is bounded by the properties of the steering mechanics.

In the following sections, the calculations for the different turning types are presented. LRL and RLR are the basic turning types for headland turnings. LRL turning is presented in the next section. The calculation for RLR turning is equal to that for LRL with the mirrored axis, and for that reason it is omitted. After that, LSL turning is discussed. LSL turning is used if the turning distance is rather long relative to the minimum turning radius. Finally, LSR turning is discussed. The final turning path is selected from among the set of feasible turnings such that the turning path has a minimum length.

4.2.2 LRL and RLR turnings

The calculation for LRL turning is presented via a pseudo code in Algorithm 4.2 and via a drawing in Figure 4.8. First, the starting and ending spirals are created by translating and rotating pre-calculated spirals such that the

starting position of $Spiral_l$ is equal to position PA and the end position of $Spiral_4$, is equal to position PB . Then, the centre of the middle turning circle is calculated by finding the section at which the two circles overlap; here, the centre points are equal to the centre points of the $Spiral_l$ ending turning circle and the $Spiral_4$ starting turning circle. The radii of the overlapping circles are calculated based on the spirals (symbol d in Figure 4.7), the curvatures of which start at k_{start} and end at k_{center} and equally from k_{center} to k_{end} . If such an overlapping section is found, then the corresponding spirals between the different turning circles can be created again by translating and rotating the pre-calculated spiral. Finally, the feasibility of the solution is checked. If the travelling angle within the starting or ending circle is greater than half of the circle, i.e. if the path has a loop, then the curvature of the starting or ending circle is decreased until a feasible solution is found or until the decreased curvature reaches the starting or ending curvature. Ultimately, the feasible turning path is evaluated based on the spirals that have been created.

Algorithm 4.2. GenerateLRLTurning

Input: PA : starting pose consist position (x, y) orientation (θ) and curvature (k)
 PB : ending pose consist position (x, y) orientation (θ) and curvature (k)

Output: $Path$: LRL turning path

$k_{start} = k_{end} = \max$ left curvature
 $k_{center} = \max$ right curvature
while ($k_{start} > PA.k$ & $k_{end} > PB.k$)
 $Spiral_l =$ copy $Spiral$ from curvature $PA.k$ to curvature k_{start}
 $Spiral_l =$ move $Spiral_l$ such that spiral start pose equals to PA

$Spiral_4 =$ copy $Spiral$ from curvature k_{end} to curvature $PB.k$
 $Spiral_4 =$ move $Spiral_4$ such that spiral end pose equals to PB

$O_{center} =$ find crossing of circles which centre points are $Spiral_l.O(k_{start})$ and $Spiral_4.O(k_{end})$ and radii $|O(k_{start}) - O(k_{center})|$ and $|O(k_{center}) - O(k_{end})|$.

if (O_{center} not found)
LRL is not feasible turning type
end if

$Spiral_2 =$ copy $Spiral$ from curvature k_{start} to curvature k_{center}
 $Spiral_2 =$ move $Spiral_2$ such that spiral starts from $Spiral_l.O(k_{start})$ and ends to O_{center}

$Spiral_3 =$ copy $Spiral$ from curvature k_{center} to curvature k_{end}
 $Spiral_3 =$ move $Spiral_3$ such that spiral starts from O_{center} and ends to $Spiral_4.O(k_{end})$

if (Angle of sector from $Spiral_l.P(k_{start})$ to $Spiral_2.P(k_{start})$ counterclockwise using $Spiral_l.O(k_{start})$ as center point $> \pi$)
continue with $k_{start} =$ decrease k_{start}
end if

if (Angle of sector from $Spiral_3.P(k_{end})$ to $Spiral_4.P(k_{end})$ counterclockwise using $Spiral_4.O(k_{end})$ as center point $> \pi$)
continue with $k_{end} =$ decrease k_{end}
end if

$Path =$ evaluate path from $Spiral_l$, $Spiral_2$, $Spiral_3$ and $Spiral_4$
end while

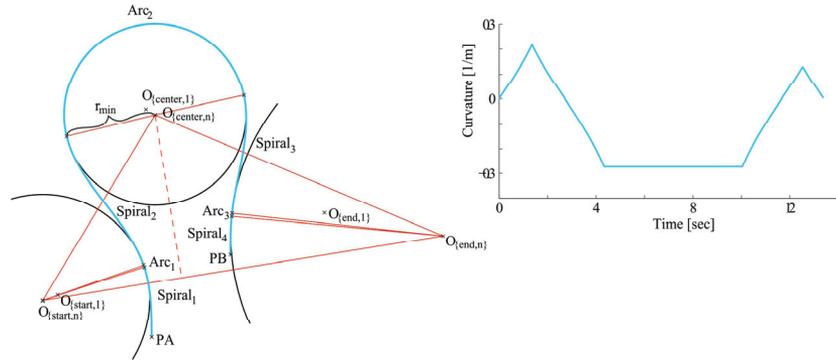


Figure 4.8. LRL turning from position PA to position PB using spirals. The drawing consists of seven different segments: three arcs and four connecting spirals.

4.2.3 LSL and RSR turnings

The calculation for LSL turning is presented via a pseudo code in Algorithm 4.3 and via a drawing in Figure 4.9. As with the LRL turning, first the starting and ending spirals need to be created by translating and rotating the pre-calculated spirals such that the starting position of $Spiral_i$ is equal to position PA and the end position of $Spiral_4$ is equal to position PB. Then, the orientation of the centre line is determined based on the starting and ending circles and the associated distances to the path tangent (r_1 and r_2 in Figure 4.7 and in Algorithm 4.1). If it is possible to connect the starting and ending circles, then corresponding spirals, $Spiral_2$ and $Spiral_3$, are created such that the starting and ending circles of the spirals are equal to the $Spiral_1$ and $Spiral_4$ ending and starting circles, respectively. Finally, the feasibility of the solution is checked using a two-part test. The first test is used to check if a line between $Spiral_2$ and $Spiral_3$ exists. If the spirals intersect, then the curvature of the centre line is increased and the spirals connected to this arc are searched for iteratively. If the first test is passed, the feasibility of the first and last arc is checked, just as in the LRL turning algorithm. If the path has a loop either in the first or last arc, then the corresponding curvature is decreased and the algorithm is repeated iteratively until either a starting or ending curvature is reached or until a feasible solution is found. Ultimately, the feasible turning path is evaluated based on the spirals that have been created.

Algorithm 4.3. GenerateLSLTurning

Input: PA : starting pose consist position (x, y) orientation (θ) and curvature (k)
 PB : ending pose consist position (x, y) orientation (θ) and curvature (k)
Output: $Path$: LSL turning path

$k_{start} = k_{end} = \max$ left curvature

while ($k_{start} > PA.k$ & $k_{end} > PB.k$)

$k_{mid} = 0$

$Spiral_1 =$ copy $Spiral$ from curvature $PA.k$ to curvature k_{start}

$Spiral_1 =$ move $Spiral_1$ such that spiral start pose equals to PA

$Spiral_4 =$ copy $Spiral$ from curvature k_{end} to curvature $PB.k$

$Spiral_4 =$ move $Spiral_4$ such that spiral end pose equals to PB

$d = |Spiral_1.O(k_{start}) - Spiral_4.O(k_{end})|$

while ($k_{mid} < k_{start}$ & $k_{mid} < k_{end}$)

if ($|r_1(k_{start}, k_{mid}) - r_2(k_{end}, k_{mid})| > d$)

LSL is not feasible turning type

end if

$\hat{\varphi} =$ angle of line between $Spiral_1.O(k_{start})$ and $Spiral_4.O(k_{end})$

$\varphi = \hat{\varphi} - \text{asin} \left(\frac{r_2(k_{end}, k_{mid}) - r_1(k_{start}, k_{mid})}{d} \right)$

$Spiral_2 =$ copy $Spiral$ from curvature k_{start} to curvature k_{mid}

$Spiral_2 =$ move $Spiral_2$ such that spiral starts from $Spiral_1.O(k_{start})$ and $\theta(k_{mid}) = \varphi$

$Spiral_3 =$ copy $Spiral$ from curvature k_{mid} to curvature k_{end}

$Spiral_3 =$ move $Spiral_3$ such that spiral ends to $Spiral_4.O(k_{end})$ and $\theta(k_{mid}) = \varphi$

if (Angle of line between $Spiral_2.P(k_{mid})$ and $Spiral_3.P(k_{mid}) \neq \varphi$)

continue with $k_{mid} =$ increase k_{mid}

end if

end while ($k_{mid} < k_{start}$ & $k_{mid} < k_{end}$)

if (Angle of sector from $Spiral_1.P(k_{start})$ to $Spiral_2.P(k_{start})$ counterclockwise using $Spiral_1.O(k_{start})$ as center point $> \pi$)

continue with $k_{start} =$ decrease k_{start}

end if

if (Angle of sector from $Spiral_3.P(k_{end})$ to $Spiral_4.P(k_{end})$ counterclockwise using $Spiral_4.O(k_{end})$ as center point $> \pi$)

continue with $k_{end} =$ decrease k_{end}

end if

$Path =$ evaluate path from $Spiral_1, Spiral_2, Spiral_3$ and $Spiral_4$

end while ($k_{start} > PA.k$ & $k_{end} > PB.k$)

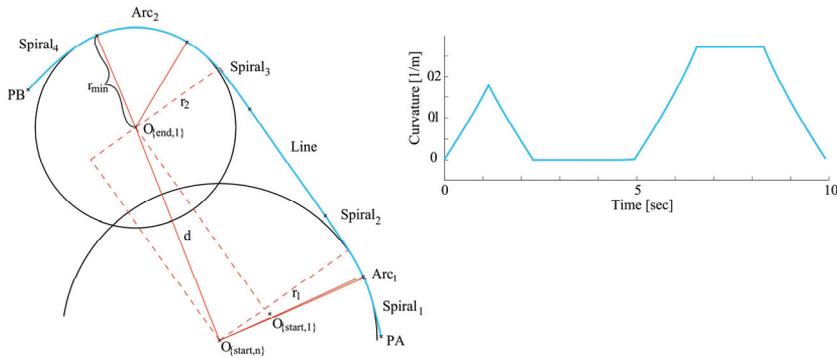


Figure 4.9. LSL turning from position PA to position PB using spirals. The drawing consists of seven different segments: two arcs, one line and four connecting spirals.

4.2.4 LSR and RSL turning

The calculation for LSR turning is presented via a pseudo code in Algorithm 4.4 and via a drawing in Figure 4.10. The algorithm is quite similar to the LSL algorithm, but the curvature in the middle of the path must go through zero, and for that reason the algorithm is somewhat simpler. Again, the starting and ending spirals are created first. After that, the centre spirals and the possible centre line are created. If the path has loops, the starting or ending circle is reduced until the path is feasible or until a solution is no longer possible. Finally, the resulting path is evaluated based on the spirals that have been created.

Algorithm 4.4. GenerateLSRTurning

Input: PA : starting pose consist position (x, y) orientation (θ) and curvature (k)
 PB : ending pose consist position (x, y) orientation (θ) and curvature (k)
Output: $Path$: LSR turning path

$k_{start} = \max$ left curvature
 $k_{end} = \max$ right curvature
 $k_{mid} = 0$
while ($k_{start} > PA.k$ & $k_{end} < PB.k$)
 $Spiral_1 =$ copy $Spiral$ from curvature $PA.k$ to curvature k_{start}
 $Spiral_1 =$ move $Spiral_1$ such that spiral start pose equals to PA

 $Spiral_4 =$ copy $Spiral$ from curvature k_{end} to curvature $PB.k$
 $Spiral_4 =$ move $Spiral_4$ such that spiral end pose equals to PB

 $d = |Spiral_1.O(k_{start}) - Spiral_4.O(k_{end})|$

if ($|O(k_{start}) - O(k_{end})| < d$)
LSR is not feasible turning type
end if

 $\hat{\varphi} =$ angle of line between $Spiral_1.O(k_{start})$ and $Spiral_4.O(k_{end})$
 $\varphi = \hat{\varphi} - \text{asin}\left(\frac{r_2(k_{end}, k_{mid}) + r_1(k_{start}, k_{mid})}{d}\right)$

 $Spiral_2 =$ copy $Spiral$ from curvature k_{start} to curvature k_{mid}
 $Spiral_2 =$ move $Spiral_2$ such that spiral starts form $Spiral_1.O(k_{start})$ and $\theta(k_{mid}) = \varphi$

 $Spiral_3 =$ copy $Spiral$ from curvature k_{mid} to curvature k_{end}
 $Spiral_3 =$ move $Spiral_3$ such that spiral ends to $Spiral_4.O(k_{start})$ and $\theta(k_{mid}) = \varphi$

if (Angle of sector from $Spiral_1.P(k_{start})$ to $Spiral_2.P(k_{start})$ counterclockwise using
 $Spiral_1.O(k_{start})$ as center point $> \pi$)
continue with $k_{start} =$ decrease k_{start}
end if
if (Angle of sector from $Spiral_3.P(k_{end})$ to $Spiral_4.P(k_{end})$ clockwise using
 $Spiral_4.O(k_{end})$ as center point $> \pi$)
continue with $k_{end} =$ increase k_{end}
end if

 $Path =$ evaluate path from $Spiral_1, Spiral_2, Spiral_3$ and $Spiral_4$
end while

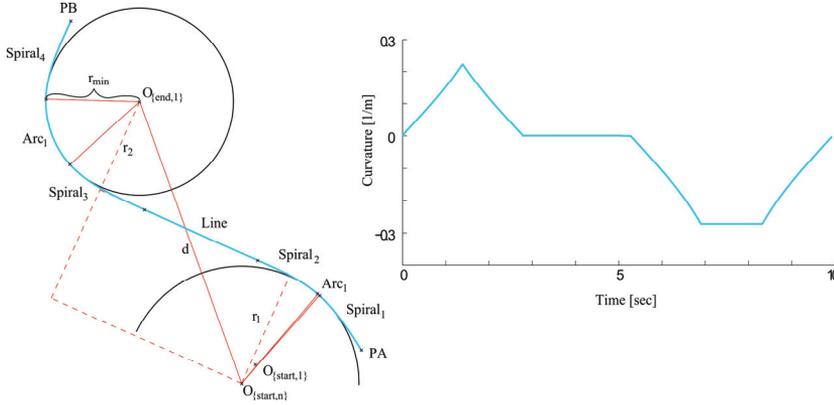


Figure 4.10. LSR turning from position PA to position PB using spirals. The drawing consists of seven different segments: two arcs, one line and four connecting spirals.

4.2.5 Path Smoothing

Another challenge in path generation arises when the previous driving line is used to create a new path, resulting in too sharp curves. Especially with inner curves, the turning circle of the driving line decreases and eventually it is impossible to follow.

Algorithm 4.5 replaces the path points in the inner curves such that the resulting curve is feasible. Before the algorithm can be applied, the entire path is first scanned and the starting and ending points of sharp curves are identified. The curvature limit is calculated based on the working width according to the following equation:

$$k_{limit} = \frac{\text{sign}(\text{follow_dist})}{a / \tan(\alpha_{max}) + |\text{follow_dist}|}, \quad (4.25)$$

where k_{limit} is the limited curvature, follow_dist is the distance to the adjacent driving line, a is the wheelbase and α_{max} is the maximum steering angle.

Algorithm 4.5 uses the starting and ending points of the too sharp curve and the limit values together with the original path as input. The output of the algorithm is the smoothed path, where the path points in the neighbourhood of the sharp curve are moved so that the curvature of the resulting path is less than the limit (Figure 4.11).

The algorithm first creates several starting and ending spirals for the circle of maximum curvature from path points starting from the given start and end points and moving further away from the limited curve. The spiral's starting and ending points have the same position, orientation and curvature as the original path's starting and ending points, respectively. The centre positions of the turning circles in the created spirals form two polylines. The point at which these polylines overlap is the centre of the desired turning circle. The starting and ending spirals extending to and

from this turning circle can be calculated by taking the weighted average of the starting and ending spirals before and after the crossing section. The resulting smoothed path is obtained by moving the original path points to the nearest position in the evaluated spiral-arc-spiral path.

Algorithm 4.5. ReplaceSharpCurve

Input: *start* : starting position on *Path*
end : ending position on *Path*
k_{limit} : maximum limited curvature
Path : original path

Output: *SmoothPath* : new path with limited curvature

for *i* := 1 **to** SEARCH_POINTS
Spiral_{start}[i] = copy *Spiral* from curvature *Path*[*start-i*].*k* to curvature *k_{limit}*
Spiral_{start}[i] = move *Spiral_{start}[i]* such that spiral start pose equals to *Path*[*start-i*]

O_{start}[i] = *Spiral_{start}[i]*.*O*(*k_{limit}*)
end for

for *i* := 1 **to** SEARCH_POINTS
Spiral_{end}[i] = copy *Spiral* from curvature *k_{limit}* to curvature *Path*[*end+i*].*k*
Spiral_{end}[i] = move *Spiral_{end}[i]* such that spiral end pose equals to *Path*[*end+i*]

O_{end}[i] = *Spiral_{end}[i]*.*O*(*k_{limit}*)
end for

[*start*, *end*] = Crossing positions of polylines *O_{start}* and *O_{end}*

Spiral₁ = Weighted average of *Spiral_{start}*[*floor*(*start*)] and *Spiral_{start}*[*ceil*(*start*)]
using weights *start-floor*(*start*) and *ceil*(*start*)-*start*

Spiral₂ = Weighted average of *Spiral_{end}*[*floor*(*end*)] and *Spiral_{end}*[*ceil*(*end*)]
using weights *end-floor*(*end*) and *ceil*(*end*)-*end*

SmoothCurve = evaluate path from *Spiral₁* and *Spiral₂*
SmoothPath = replace points from *Path*[*start*] to *Path*[*end*] using *SmoothCurve*

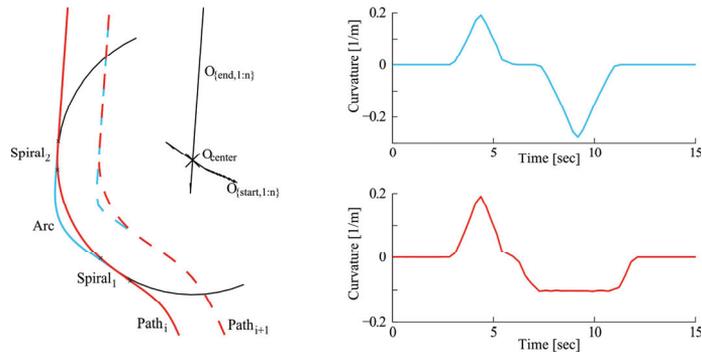


Figure 4.11. The ReplaceSharpCurve algorithm finds suitable spirals to connect the original (solid blue) path to the circular arch (black), the radius of which is equal to the minimum turning radius plus the working width. Because the gap between the stored path points is relatively large, the resulting limited curvature (red plot line) is quite smooth.

4.2.6 Simplified path planning

The basic idea of the path planning algorithm is to follow the previously driven driving lines or swaths with some offset that is a multiple of the working width. The simplified path planning algorithm for agricultural vehicles utilises the spiral algorithms described above.

The working order of the field is always the same. The field boundaries are first processed by driving around the field. After completing a predetermined number of cycles, the last driving line (or cycle) is decomposed into relatively straight segments based on an algorithm proposed by Oksanen (2007) and the longest segment is searched. The cycle is continuously tracked until the longest segment is reached. After that, the inner area is processed.

A state diagram of the simplified path planning algorithm is presented in Figure 4.12. The algorithm has three main states: *PathStart*, *PathNearEnd* and *PathChange*. Path planning is carried out when these states have been entered. The triggering points for these states are illustrated on the right side of Figure 4.12. When the headland area is processed and the path is close to an end, a new recorded path is connected to the currently followed path. When this new path is followed rather than the previous path, the new path is no longer recorded and the path is saved in the storage memory. The currently followed path is also smoothed out using Algorithm 4.5. The process is repeated until the turning pattern has ended.

Inner area processing has the same triggering points as headland area processing. However, when the currently followed path is a working path, headland turning is generated before the working path has ended. When the currently followed path changes to a headland path, the path closest to the end of the turning end point is searched and connected to the headland path. If there are crossing points on the connected path, they are removed, i.e. the connected path is shortened such that it does not cross any previously driven path. After the headland path changes to a working path, recording is restarted. The new path is saved only when the path to be followed is a working path. The process is repeated until the whole field has been processed.

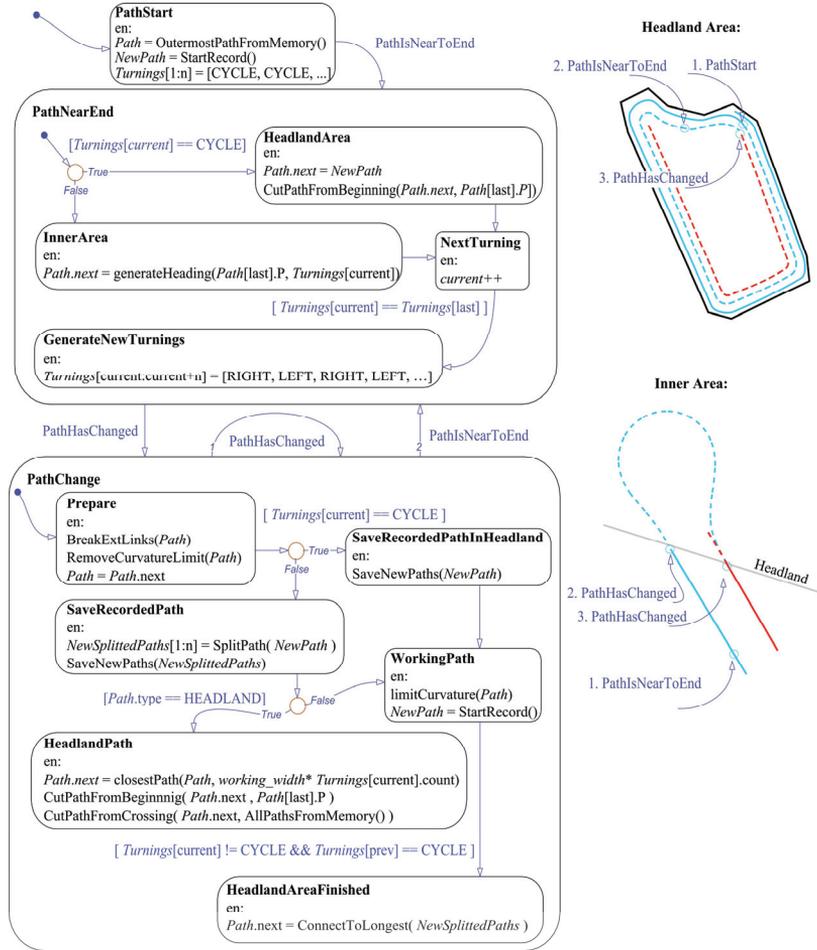


Figure 4.12. A state diagram of the simplified path planning algorithm. The right side of the diagram illustrates the triggering points of the state chart both in the headland area work and in the inner area work as well as the evolution of the path; the solid blue line is the original path, the dashed blue line is the next path connected to the original path and the red line is the last path connected at this stage.

4.3 State estimation

The state of the controlled system has to be estimated based on the measurements and the models described in the chapter 3.4. The obtained measurements are delayed at some specific amount of time. Also, the control outputs are delayed and the actuators include dynamics. The NMPC controller needs a best possible at least approximately optimal estimate of the state at time instant when the current control outputs affect the controlled system. Otherwise, the stability of the controller is uncertain.

4.3.1 Extended Kalman Filter

The Extended Kalman Filter (EKF) was used for the state estimation. The EKF implemented as a part of this thesis follows standard estimation methods. The general form of the estimated system model is as follows:

$$\begin{aligned}\hat{x}(t_{k+1}) &= f_{est}(\hat{x}(t_k), u(t_k)) + w(t_k) \\ \hat{y}(t_k) &= h(\hat{x}(t_k)) + v(t_k),\end{aligned}\tag{4.26}$$

where f_{est} is the estimation model for the system and h is the measurement function. The difference between it and the prediction model used in the NMPC model is that the estimation model includes noise terms, ($w(t_k)$ and $v(t_k)$), both in the state equation and in the measurement equation. The noise terms are supposed to be independent and white Gaussian noise:

$$\begin{aligned}p(w(t_k)) &\sim N(0, W(t_k)) \\ p(v(t_k)) &\sim N(0, V(t_k)),\end{aligned}\tag{4.27}$$

where $W(t_k)$ and $V(t_k)$ are the covariance matrices of the noises at time t_k .

Another difference is that the estimated state vector includes augmented delayed states in order to obtain the delayed measurements:

$$\hat{x}(t_k) = \begin{bmatrix} x(t_k) \\ x(t_{k-1}) \\ x(t_{k-2}) \\ \vdots \\ x(t_{k-n}) \end{bmatrix},\tag{4.28}$$

where n is the number of delayed states and x is the state vector of the system (Equation 3.10).

The estimation model predicts the new state of the system ($x(t_{k+1})$) using a model of the system (f in Equation 3.12) and moves the previously estimated states further within the augmented state vector:

$$f_{est}(\hat{x}(t_k), u(t_k)) = \begin{bmatrix} f(x(t_k), u(t_k)) \\ x(t_k) \\ x(t_{k-1}) \\ \vdots \\ x(t_{k-n+1}) \end{bmatrix},\tag{4.29}$$

The measurement model picks up certain elements from the augmented state vector such that the delay in the corresponding measurement is equal to the true delay measured within the system:

$$h(\hat{x}(t_k)) = \begin{bmatrix} x_R(t_{k-\tau(x_R)}) \\ y_R(t_{k-\tau(y_R)}) \\ \theta(t_{k-\tau(\theta)}) \\ v_t(t_{k-\tau(v_t)}) \\ \alpha_t(t_{k-\tau(\alpha_t)}) \\ \beta(t_{k-\tau(\beta)}) \\ \gamma_t(t_{k-\tau(\gamma_t)}) \\ x_E(t_{k-\tau(x_E)}) \\ y_E(t_{k-\tau(y_E)}) \end{bmatrix}, \quad (4.30)$$

where $\tau(\cdot)$ is the delay (sampling periods) in the corresponding measurement.

4.3.2 Heading measurement in estimation

The heading of a tractor is an important measurement for accurate path tracking, especially when slipping occurs and needs to be estimated. A Fibre Optic Gyro was used for the heading measurement. The raw measurement obtained by the FOG was first processed before it was used in the EKF.

The FOG that was used measures the rotation around the z-axis of the vehicle coordinate system. Because the vehicle moves on uneven surfaces, the measured rotation is not the same as the rotation around the z-axis of the ground coordinate system. The *Yaw* angle, or the *Course* that needs to be measured in general, is the rotation around the z-axis of the ground coordinate system. The accuracy of the gyro also necessitates that the earth's rotation must be compensated for.

The rate of the earth's rotation, which interferes with the measurement, is first calculated. The dot product between the earth's rotation axis and the z-axis of the vehicle coordinate system is calculated as follows:

$$Z_{earth} \cdot Z_{vehicle} = \begin{bmatrix} \cos(Lat) \\ 0 \\ \sin(Lon) \end{bmatrix} \cdot \begin{bmatrix} cR * cY * sP + sR * sY \\ cR * sY * sP - sR * cY \\ cP * cR \end{bmatrix}, \quad (4.31)$$

where

$$\begin{aligned} cP &= \cos(Pitch), & sP &= \sin(Pitch), \\ cR &= \cos(Roll), & sR &= \sin(Roll), \\ cY &= \cos(Yaw), & sY &= \sin(Yaw). \end{aligned} \quad (4.32)$$

The *Yaw* is the rotation around the z-axis of the ground coordinate system, with the positive direction being up. The *Roll* and the *Pitch* are the inclination angles around the x-axis and y-axis of the vehicle coordinate system, respectively. Now, the earth's rotation measured by the gyro can be calculated according to the following equation:

$$\omega_{earth} = -7.29212 * 10^{-9} * (Z_{earth} \cdot Z_{vehicle}). \quad (4.33)$$

The second step is to correct the measurement error due to the inclination. The rotation rate that the gyro measure is projected onto the ground after removing the measured rotation of the earth and the measured pitch rotation:

$$\omega_{yaw} = \frac{\omega_{gyro} - \omega_{earth} + \omega_{pitch} * sR}{cR * cP}, \quad (4.34)$$

where ω_{yaw} is the projected measurement of the rotation rate around the z-axis of the ground coordinate system, ω_{gyro} is the raw rotation rate measured by the gyro and ω_{pitch} is the *Pitch* angle rate.

The notation ω_{yaw} is now the rotation rate around the z-axis of the ground coordinate system; the *Yaw* angle can be calculated based on it:

$$Yaw = \widetilde{Yaw} + Yaw_{bias} = \int \omega_{yaw} + Y_{bias}, \quad (4.35)$$

where Yaw_{bias} corrects the zero heading to the north and \widetilde{Yaw} is the uncorrected *Yaw* angle, which is calculated by integrating ω_{yaw} .

The bias Yaw_{bias} is found by using the *Compass* measurement. Since no magnetometer is included, the compass angle is obtained from the RTK-GPS receiver in the form of a course. The GPS course measurement is not accurate for slow driving speeds. For that reason, there are three different methods for different driving speed ranges.

When the driving speed is above 0.5 m/s, the course measurement from the RTK-GPS receiver is considered to be sufficiently accurate. First, the momentary difference between the uncorrected *Yaw* angle and the measured *Compass* angle is calculated:

$$\Delta \widetilde{Yaw} = \text{mod}(\widetilde{Yaw}(t_{k-\tau(\mathcal{Y})}) - Yaw_{compass}, 2\pi), \quad (4.36)$$

where $\widetilde{Yaw}(t_{k-\tau(\mathcal{Y})})$ is the delayed *Yaw* angle if there is a difference between the gyro and RTK-GPS delays. After that, the bias is found by filtering the momentary difference values:

$$Yaw_{bias}(t_{k+1}) = \text{mod}(Yaw_{bias}(t_k) + K * \Delta Yaw_{bias}, 2\pi), \quad (4.37)$$

where K is the filter weight and ΔYaw_{bias} is calculated based on the following equation:

$$\Delta Yaw_{bias} = \text{mod}(\Delta \widetilde{Yaw} - Yaw_{bias}(t_k) + \pi, 2\pi) - \pi. \quad (4.38)$$

The weight K is modified for every measurement update step according to the following equation:

$$K = \max\left(\frac{1}{\frac{1}{K} + 1}, K_{min}\right), \quad (4.39)$$

where K_{min} is the minimum weight of the filter.

When the driving speed is below 0.5 m/s but above 0.05 m/s, there is no accurate absolute course measurement. In this case, only the filter weight is modified according to the following equation:

$$K = \min\left(\frac{1}{\frac{1}{K} - 1}, 0.5\right). \quad (4.40)$$

When the driving speed is below 0.05 m/s, the course is considered constant. Driving speeds below this are not used in the field, thus this case corresponds to a situation where the system is stopped. In this case, the bias is found by keeping the heading constant:

$$Yaw_{bias}(t_{k+1}) = \overline{Yaw}(t_k) - Yaw(t_k). \quad (4.41)$$

4.3.3 Local measurement using a laser scanner

One challenge in the estimation process is to merge the global position measurement produced by the GPS and local measurement produced by the laser scanner. In order to do that, the route of the seed drill is recorded in the storage memory of the navigation system. The estimated position of the marking plough (x_P, y_P) and the produced furrow can be calculated geometrically based on the estimated (x_E, y_E) position and orientation of the seed drill (Equation 3.9). The estimated distance from the furrow can be calculated based on the recorded furrow positions and the estimated laser scanner position (x_L, y_L) in the same manner as the trajectory points were calculated based on the path in the NMPC method (Section 4.1.1). However, in this case the direction of the distance vector is known (Figure 4.13).

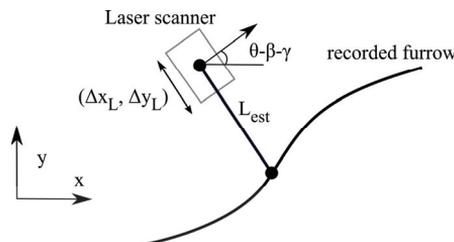


Figure 4.13. Correcting the estimated position of the laser scanner using the laser scanner measurement and recorded furrow position. Note that L_{est} is not necessarily perpendicular to the furrow (as it is for the trajectory generation).

The laser scanner measures the vertical profile of the ground. The ground level is estimated using this profile by fitting a first-degree polynomial into the profile by minimising the MSE error. Only the part of the profile that is close to the previously detected swath position is used for the ground-level estimation. The lateral distance to the adjacent swath is found by fitting the prototype of the furrow profile into the residuals of the first-degree polynomial fit. Figure 4.14 shows a drawing of the field profile and the fitted ground level together with the mark prototype.

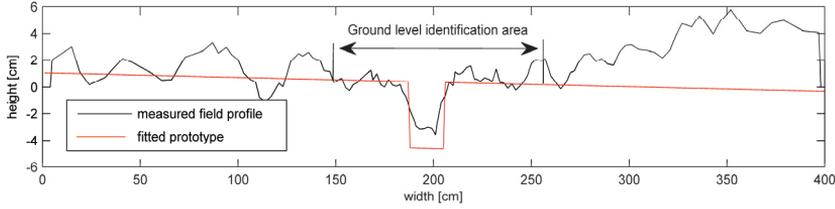


Figure 4.14. Laser scanner measurements and fitted mark prototype; note that the scale is different for the x-axis and y-axis.

The difference between the measured and estimated distances reveal the extent to which the current and past estimations differ laterally from one another at the angle perpendicular to the seed drill's current heading (Figure 4.13). Correcting both estimates would require recalculating all of the recorded estimates again for every estimation step, which would increase the number of calculated, estimated positions at every time step. Because this would eventually require too much computation time in real time, only the current estimate is corrected via the EKF. The correction equations are as follows:

$$\begin{aligned}\Delta x_L &= (L_{\text{meas}} - L_{\text{est}}) \cos(\theta - \beta - \gamma_t) \\ \Delta y_L &= (L_{\text{meas}} - L_{\text{est}}) \sin(\theta - \beta - \gamma_t),\end{aligned}\quad (4.42)$$

where L_{est} is the estimated lateral distance, calculated based on the recorded furrow positions, and L_{meas} is the measured lateral distance. The correction values are used directly in the EKF as innovation terms, hence:

$$\begin{aligned}x_{E,\text{meas}}(t_k) &= h(\hat{x}(t_k))_{\{x_E\}} + \Delta x_L \\ y_{E,\text{meas}}(t_k) &= h(\hat{x}(t_k))_{\{y_E\}} + \Delta y_L,\end{aligned}\quad (4.43)$$

where $x_{E,\text{meas}}$ is the x coordinate and $y_{E,\text{meas}}$ is the y coordinate for a pseudo measurement of the laser scanner.

In addition to the innovation terms, the reliability of the measurement is estimated based on the goodness of the fit in mark searching procedure and also based on the characteristics of the L_{meas} signal. If the signal is very noisy, i.e. if the furrow is not continuously found, the reliability is

decreased. The reliability is used to increase the value of the scanner's noise parameter in the EKF.

4.4 The structure of the navigation system

A very important part of the navigation system, besides the algorithms described in the previous chapters, is the implementation and the physical structure of the system. As discussed in Section 3.1, agricultural machines are quite modular; usually, they consist of a tractor and at least one implement. The implement can be a front- or rear-mounted implement or it can be towed. There are many possible alternatives for the kinematic structure of the machine combination. Therefore, a general combined navigation system cannot be designed for any particular configuration; instead, it must also be modular such that it can conform to the actual machinery.

4.4.1 Components of the navigation system

The physical structure of the navigation system consists of positioning devices, actuating devices and navigation devices. The positioning devices in the application presented here included a ground-speed sensor (radar) for speed measurement, an IMU for roll and pitch-angle measurements, a FOG for heading measurements, a RTK-GPS receiver for accurate position measurements and 2D laser scanners for local position measurements and obstacle detection. The actuating devices included a steering controller, a cruise controller and a hydraulic valve for implement control. The navigation devices were the position estimation and guidance controllers. All of these devices were connected to one another through the ISO 11783 network. Figure 4.15 depicts the physical structure of the navigation system.

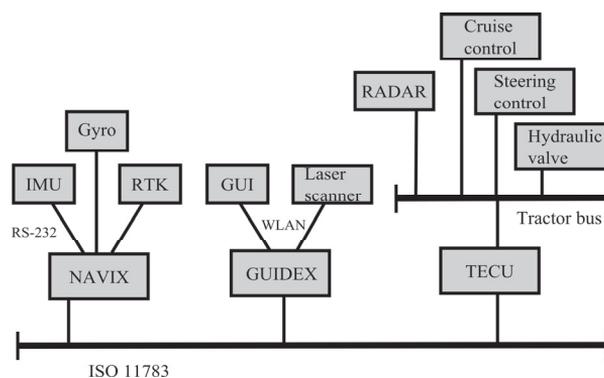


Figure 4.15. The physical architecture of the navigation system was built upon an ISO 11783 network.

The position estimation controller (labelled *Navix*) was used mainly to estimate the yaw angle more precisely than the RTK-GPS device does (Section 4.3.2). The global position was measured directly using the RTK-GPS device. The RTK-GPS measurement was then converted into the tractor coordinate system origin (the centre point of the rear axle at ground level) with the help of roll and pitch angles. The roll and pitch angles were collected directly from the commercial IMU.

The guidance system controller (labelled *Guidex*) was the main element of the navigation system. It contained the principal software that calculates the control commands and sends them through the ISO 11783 network to the actuating devices (ISO 11783-7:2009). In the presented application, the guidance system controller was based on a standard desktop computer and it was also used for handling the laser scanner measurements.

Logically, the physical components of the system are as follows: the tractor, the implement and the guidance system controller. The measuring devices for the tractor state and the steering system components were grouped into a single logical unit called simply the *Tractor*, whereas the measuring devices for the implement state and the implement control components were grouped into another logical unit called the *Implement*. This was done despite the fact that the mechanical control components were still located inside the tractor (the hydraulic valve). The logical grouping is depicted in Figure 4.16.

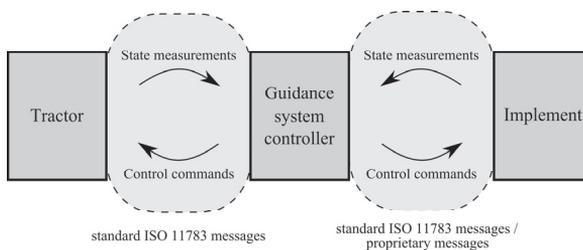


Figure 4.16. The logical architecture of the system from a navigational point of view.

Messages were transferred between the different components in the navigation system via the ISO 11783 network. The messages are listed in Table 4.1. The table also lists the physical device that was sending the message in question as well as the PGN number and the standard that defined the message.

Table 4.1. Messages sent between different components in the navigation system.

Function / Information	Device	From - To	PGN (hex)	Standard
Attitude (roll and pitch)	Navix	Tractor - Guidance	1F119	IEC 61162-3
Position	Navix	Tractor - Guidance	1F801	IEC 61162-3
COG&SOG (yaw and speed)	Navix	Tractor - Guidance	1F802	IEC 61162-3
GNSS Position data	Navix	Tractor - Guidance	1F805	IEC 61162-3
GNSS Pseudo noise statistics	Navix	Tractor - Guidance	1FA06	IEC 61162-3
Measured curvature	Steering control	Tractor - Guidance	AC00	ISO 11783-7
Set-point curvature	Guidex	Guidance - Tractor	AD00	ISO 11783-7
Measured wheel speed	Cruise control	Tractor - Guidance	FE48	ISO 11783-7
Measured ground speed	Radar	Tractor - Guidance	FE49	ISO 11783-7
Response set-point speed	Cruise control	Tractor - Guidance	FE0A	ISO 11783-7
Set-point speed	Guidex	Guidance - Tractor	FE0B	ISO 11783-7
Measured work state	Implement	Implement - Guidance	FF16	<i>proprietary</i>
Set-point work state	Guidex	Guidance - Implement	FF15	<i>proprietary</i>
Measured drawbar angles	Drawbar	Implement - Guidance	FF13	<i>proprietary</i>
Measured lateral distance	Laser scanner	Implement - Guidance	FF14	<i>proprietary</i>
Estimated valve flow	Valve	Tractor - Guidance	FE1x	ISO 11783-7
Set-point valve flow	Guidex	Guidance - Tractor	FE3x	ISO 11783-7

The messages sent between the Tractor and the guidance computer included the position and orientation messages, the speed information, the steering angle measurement and the control messages. These are all standard ISO 11783 messages (ISO 11783-7:2009).

In a headland operation, the implement has to be changed from a working state to an inactive state and back again after completing a turning manoeuvre. Since the objective is to control the driving autonomously without the driver intervening, the navigation system should also command the working state of the implement. In the application presented here, the state measurement and control were implemented using proprietary messages. The standard, however, allows for proprietary messages (ISO 11783-1:2007; ISO 11783-3:2007).

In the presented application, the implement also had a controllable joint in the drawbar. The low-level controller for this joint was implemented in the guidance system controller and the control messages consisted of the hydraulic valve commands (ISO 11783-7:2007). The actual angle is needed for both the controller and the state estimation. Currently, the ISO 11783 standard does not support these kinds of messages directly. That is the reason why the state information had to be carried out using proprietary messages.

4.4.2 Software architecture

The main goal was to get the software architecture for the navigation system to support all of the machinery configurations, while still remaining

as simple as possible. The logical structure of the software is depicted in Figure 4.17. The software has two concurrent loops: the *estimation* loop and the *optimisation* loop.

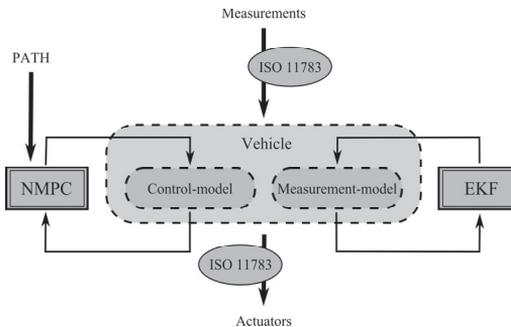


Figure 4.17. The logical structure of the software architecture has two concurrent loops: the estimation loop (EKF) and the optimisation loop (NMPC). The principal information flows from the measurements to the actuators through the ISO 11783 network.

The software architecture consists of four different modules: *PATH*, *NMPC*, *EKF* and *Vehicle*. The *EKF* and *Vehicle* modules are involved in the estimation loop, whereas the *PATH*, *NMPC* and *Vehicle* modules are involved in the optimisation loop. The information carried between these modules and loops is transferred through the *Vehicle* module. For the experimental implementation developed here, all of the modules were implemented using the object-oriented, C++ programming language. The whole implementation consists of approximately 57,000 lines of code.

NMPC module

There are many different methods for solving the constrained nonlinear optimisation problem described in Section 4.1. One commonly used numerical method is SQP (Schittkowski, 1983; Betts, 2001). There are many implementations of the method in different packages. In this thesis, a Nonlinear Model Predictive Control Tool, called HQP (Huge Quadratic Programming), was selected (Franke et al., 2008; Franke and Arnold, 1997). HQP has been used successfully, for instance, for batch process control (Nagy et al., 2007), energy systems and water systems. It uses an SQP algorithm to solve nonlinearly constrained problems. Convex quadratic sub-problems are solved using a polynomial time interior-point method. The Lagrangian function of the problem is approximated quadratically by a sparse Hessian matrix, which is updated numerically using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. The Jacobian matrices of the system equations and the cost function are, however, analytically solved.

The time constants in vehicle control are very different from the applications that have so far used the HQP control tool. Some modifications had to be made in order to fulfil strict time limits. The original software interface of the HQP control tool was removed and a new one, which serves the purposes of this study better, was implemented. Also, because there was no guarantee how long it would take to compute the NMPC optimisation problem, an interrupt routine was added. The scheduler is responsible for keeping the cycle time constant. If the time slot runs out and the NMPC has not yet acquired a new solution ($u = u^*(t_{k+1}|t_k)$), then the optimisation is interrupted and the previously calculated control values are used ($u = u^*(t_{k+1}|t_{k-1})$). (Note that the index k is now increased and the control value used is different from the one used in the previous step.) In the case of an interruption, the length of the prediction horizon (M) is also decreased. After ten feasible solutions, the prediction horizon starts to increase one step at a time.

EKF module

The Extended Kalman Filter was used for the state estimation and prediction. The state estimation methods were described in Section 4.3 and the algorithms described there were implemented without using any external toolkit.

PATH module

The *PATH* module was responsible for the higher-level planning, for keeping track of the traversed swaths and for determining how the area of the field should be processed. The current desired working state of the system (stop, working, headland, transfer) can also be obtained from the *PATH* module. A simplified path planning algorithm was used for the experimental navigation system (Section 4.2). However, the implementation of the path planning algorithm also makes more sophisticated planning algorithms possible.

Vehicle module

The *Vehicle* module was actually a data storage and information centre. Other modules exchange information through this module. It also consisted of the information from the controlled system: the kinematic model and all the parameters.

The *Model* class is an implementation of a generic state space model. It is a storage class for the current state estimate, controls and measurements, but it also includes methods for state transition and calculating the

Jacobian and state cost. The *Model* class also has a generic interface with both the *NMPC* and *EKF* modules, so all of the models that are inherited from this class can be used for the estimation and the optimisation process. The *NMPC* and *EKF* modules gain all the information they need from this class structure. The variables and methods of the *Model* class are presented in Figure 4.18.

Model
<pre> #_x: VECp #_u: VECp #_y: VECp #_f: VECp #_Q: MATp #_R: MATp #_dt: double </pre>
<pre> <<create>>-Model() <<destroy>>-Model() +dt(): double +dt(d: double): double +stateTransition(x: VECp, u: VECp, f: VECp): void +measure(x: VECp, y: VECp): void +step(): void +stateDifference(x: VECp, u: VECp, x_dif: VECp, x_der: VECp, u_dif: VECp, u_der: VECp): void +getWeights(Q: VECp, R: VECp): void +getStateJacobian(x: VECp, u: VECp, A: MATp): void +getControlJacobian(x: VECp, u: VECp, B: MATp): void +getMeasureJacobian(x: VECp, C: MATp): void +getStateNoise(Q: MATp): void +setStateNoise(Q: MATp): void +getInitialStateNoise(Q: MATp): void +getMeasureNoise(R: MATp): void +setMeasureNoise(R: MATp): void +getQualityVec(quality: VECp): void +getInitialStateNoise(R: MATp): void +getState(x: VECp): void +setState(x: VECp): void +getStateSize(): int +getInitialState(x: VECp): void +getStateLimits(xmin: VECp, xmax: VECp): void +getControl(u: VECp): void +setControl(u: VECp): void +getControlSize(): int +getControlLimits(umin: VECp, umax: VECp): void +getMeasure(y: VECp): void +setMeasure(y: VECp): void +getMeasureSize(): int +getMeasureDelay(d: VECp): void </pre>

Figure 4.18. The variables and methods of the *Model* class

Since the basic model of the system has been kept simple and exchangeable, there are also some auxiliary classes. These include *ModelConnect* for interconnecting two models, *ModelDelay* and *ModelIntegrator* for basic operations, and *ModelMeasureDelay* for measurement delays. All of these auxiliary classes are also inherited from the *Model* class, so the *ModelConnect* auxiliary class can be used to connect models and other auxiliary classes that have already been connected. In this way, there can be multiple simple kinematic models for the tractor and for the implement. The tractor or the implement can be changed and the rest of the model can be left untouched. Furthermore, the delay times of all of the measurements can be set independently.

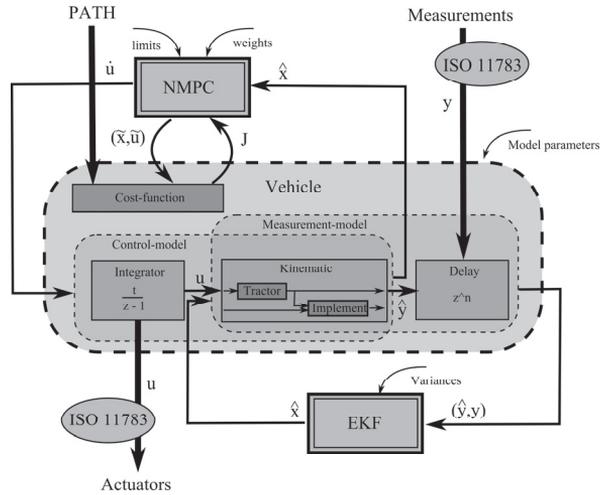


Figure 4.19. Information flows inside the software between the storage classes and the different modules.

Figure 4.19 shows the structure of the *Vehicle* module and how the information flows to and from the *NMPC* and *EKF* modules. The *Tractor* and the *Implement* models are deepest inside the *Vehicle* class. These models are interconnected, and together they comprise the *Kinematic* model. This model is further interconnected with the *MeasureDelay* auxiliary class, and together they comprise the *Measurement* model. Separate from all that, the *Kinematic* model is also connected to the *Integrator* model and together they comprise the *Control* model. The *Vehicle* class itself is not derived from the *Model* class.

4.4.3 Scheduling

Because the software was based on modules and also included several concurrent loops, some kind of scheduler was needed to synchronise the loops. In order to maintain the portability to different operation systems, the Boost C++ library (Boost, 2012) was used to create different threads and to handle the barriers and locks between the threads.

During the start-up phase, the main thread is responsible for creating and initialising all of the necessary modules. After every module is up and running properly, they are divided into four different threads: the thread for handling the laser scanner measurements, the main thread, the *NMPC* thread and the path planning thread. The thread for handling laser scanner measurements is independent of all the other threads. It runs when the new measurements arrive from the laser scanner and gives a new estimate of the swath position whenever the calculation is ready. The other threads are synchronised more precisely. The main thread is responsible for keeping the cycle time constant, in this case at 100 milliseconds. The CAN-bus

interface, the EKF calculation and the GUI messaging are part of the main thread. The NMPC has its own thread for the optimisation step and the path planning layer also has its own thread for time-consuming tasks. If the NMPC calculation is not completed before the main thread reaches the synchronisation point, the NMPC calculation is interrupted. Otherwise, the NMPC calculation waits for the new state estimate before the new calculation cycle begins. Figure 4.20 clarifies the timing schedule.

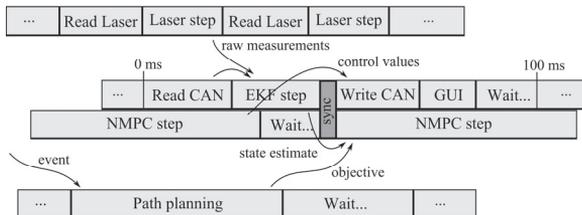


Figure 4.20. A scheduling diagram for all four threads: the thread handling the laser scanner measurements, the main thread, the NMPC thread and the path planning thread.

4.5 Simulator

Developing and testing advanced navigation algorithms and navigation systems is challenging, as a realistic environment and repeatability are required for testing the accuracy of the control actions. The more degrees of freedom and the more input the system has, the more challenging the tuning procedure becomes. Therefore, it is important that the developed algorithms can be tested in a simulator with realistic signals. Also, with the simulator the algorithms can be tuned or optimised within certain limits.

When developing embedded systems, a simulator can be used in two ways: a software-in-loop simulator can be used to test the algorithms or software against the test model in a different runtime environment (usually a development PC, with emulation), or a hardware-in-loop simulator can be used to run the software in the actual hardware, and the interface to the sensors and actuators is simulated. In this case, communication is crucial for the navigation system, as the commands and measurements are delivered over the network (Section 4.4.1). Therefore, the objective was to develop a hardware-in-loop simulator.

To simulate a vehicle, typically the following models have to be modelled: the kinematics of the vehicle, the dynamics of the system, random type noises related to inputs and outputs, and movement-based noises like wheel slipping. The simulator should emulate the ISO 11783 Tractor-ECU with a positioning and guidance option and the ISO 11783 implement (Figure 4.21). In this way, the interface of the guidance system does not need to be modified when changing from the simulation environment to the real system.

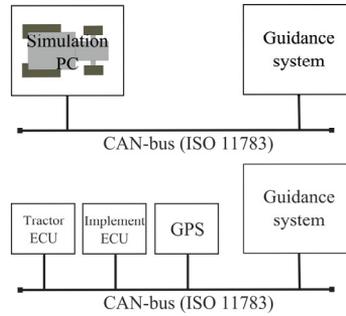


Figure 4.21. The simulation environment and the real system.

4.5.1 Simulation model

The simulator was developed completely within a Matlab Simulink environment. The overall structure of the simulator was constructed in modules (Figure 4.22). There were five different model groups: a *kinematic and dynamic model* (blue), an *environment model* (green), *error models* (red), an *interface model* (yellow) and an *auxiliary model* (gray).

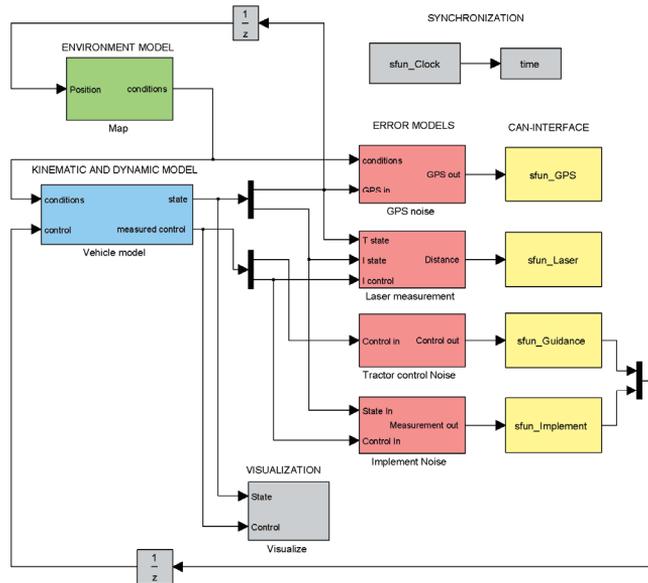


Figure 4.22. The overall structure of the simulator in a Matlab Simulink environment.

The measurement information flows from the environment model to the kinematic and dynamic model and from there to the error models, and finally, with added noise, to the external interface. Also, the control information flows from the interface model to the kinematic and dynamic model.

The environment model is basically a map that describes different conditions in the field. The condition vector can be anything, which is required in kinematic and dynamic models or in a GPS noise model.

Basically, though, this means the field conditions and available satellite configuration.

The kinematic and dynamic model is also modular. It contains separate models for the tractor and for the trailer. The models for the tractor and for the trailer can be further divided into actuator dynamic models and system kinematic models. In this way, it is easy to change the controlled system without making major modifications to the simulator itself.

The error models can be separated into a *GPS noise model*, a *laser measurement model*, a *tractor control noise model* and an *implement noise model*. The GPS noise model is an important part of the simulator, and therefore it is discussed separately in the following section. The laser measurement model uses a local auxiliary position measurement, which was used to track the previous driving line (Section 4.3.3). The tractor control noise model and the implement noise model include the added noise of the tractor and the implement measurements. The noise power is identified in the real system and is modelled as white noise with the same power in the simulator measurements.

The interface models include *GPS*, *laser*, *guidance* and *implement* models. They represent different physical devices in the real system. The measurements are packed in CAN-messages according to the ISO 11783 specifications. Also, control messages from the CAN-bus are read and transferred to Simulink. These models are developed using C-code and compiled as S-functions in Matlab.

The auxiliary models are used to control and visualize the execution of the simulator. Because the simulator is connected to the real guidance device, the simulation needs to be updated according to a real-time clock. Also, the movement of the simulated system is visualized in real time.

4.5.2 GPS noise model

There are two main procedures for constructing a realistic and controllable noise model. The first is to try to separately simulate all of the error sources and their respective errors. Unfortunately, that would require an absolute knowledge of the error structure for each factor, which is most likely impossible. The second is to try to simulate the errors themselves. The key is to define the phenomena that cause the changes and to provide a structure for the noise. To discover these phenomena and construct an adequate noise model, four separate data collections were performed.

Three GPS receivers were used to collect the data. Two of them were Trimble 5700 receivers, which were used autonomously and with the RTK-VRS correction. The third receiver was a NovAtel receiver with a decimetre-level, Omnistar, high-performance (HP) differential GPS correction, which

is quite typical for guidance usage. For each test, GGA, VTG and GST messages were collected. The message fields of interest included the coordinates, horizontal DOP, correction status, number of satellites, speed, direction, the standard deviation (SD) of the error ellipse and the SD of the coordinates. A 13° elevation mask was selected for each receiver and the DOP was not limited.

For the first test, static GPS measurements with a 1 Hz interval were collected during a 24 h period. The data was used to determine the typical variations in the daily satellite constellations. Then, the two autonomous Trimble 5700 receivers collected 10Hz of data for one hour. The data was used to determine white noise by removing the other detected appearances from it. White noise was determined by comparing data from the two identical autonomous GPS measurements.

The third test was a dynamic test that adapted horizontal positioning test parts from a dynamic GPS test standard, ISO 12188-1 (ISO 12188-1:2010). The RTK-GPS receiver was used as a reference, the driving speed was 10 km/h and the driving path was driven only three times. Dynamic tests were applied to determine rapid changes, the effects caused by movement and the changing environment. During the fourth test, the GPS antenna was completely covered for a while and the recovery time was examined.

When examining the 24 h and the dynamic data, significant correlations were only found between the latitude and longitude and the estimated errors for all of the axes. However, the north DOP is larger than the east DOP, at least in the mid-latitude area (Wu et al., 2006). For the dynamic test, which was done at 60° latitude, 52% of the horizontal error was caused by the north axis.

When examined closely, a correlation between the changes in the number of satellites (SV) and the rapid error changes was discovered based on the autonomous data. The situation was similar with the HDOP values, but the dynamic tests revealed that they were quite unstable.

The direction and speed were calculated based on the coordinates. The results correlated with the VTG message, although there were some inaccuracies caused by the projection and the delay was 1.5 measurements.

The number of satellites in view was selected as the basis for the noise model (Figure 4.23). The effect of a poor DOP was included in the daily variation equations (skyplot and positioning error). The error dynamics are characteristic of each receiver type.

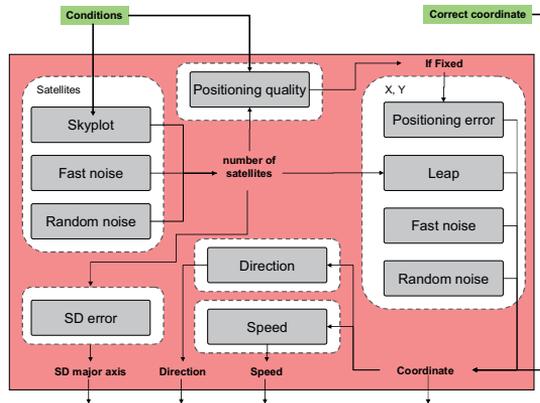


Figure 4.23. The structure of the GPS noise model

The environment model gives the correct coordinates, the level of obstructions for the satellites and the correction status information (Figure 4.23).

The skyplot represents the daily variations in the number of satellites. It was constructed from the 24 h measurements in such a way that rapid changes lasting less than 5 seconds were removed from the data (0.64%) and an equation was fitted to it using Matlab's Identification Toolbox. If requested, the number of obstructions decreased the amount of SVs.

The fast noise for the satellite amount was based on the dynamic tests so that the effect stemming from the movement could be captured. The results of the 24 h tests were removed from the dynamic test results. The fast noise removed only one SV at a time. The time of the effect was determined by first summarising the removal times from the data and then randomly selecting one.

Finally, random noise was added to the number of satellites. It temporally removed one SV with a random 0.64% occurrence.

The error models for the x and y coordinates, the direction, the speed and the standard deviation of the semi-major axis of error ellipse were then identified. In addition to the inputted absolute GPS coordinates, the position error equation also generated an error (Figure 4.23). The positioning error equation was generated similarly to the skyplot generation. Independent equations were developed for the autonomous and HP-corrected messages. For the HP positioning error model, errors measured in the dynamic tests were exploited as source data. First, a leap equation effect (which is presented in the next paragraph) was removed from the data. No significant errors or drifting in the HP data were found when using the static 24 h tests. State-space models were constructed for the HP positioning errors for the coordinate values.

Since the changes in the amount of satellites correlated with the rapid error changes, a leap equation was constructed. If the number of satellites decreases, then the leap equation adds an error. This error fades within the amount of time identified in the 24 h tests. With the Y error, 1-5 cm leaps in the elevation data lasting 0.5-2 seconds were discovered.

The fast noise was then constructed. The rapid changes filtered from the positioning error were used to determine the fast noise; its occurrence was then randomised. Finally, a random noise was added. The random noise was based on the white noise of the source data. The sign of the random values was kept constant for a random time period to introduce a small-detail drifting effect.

The direction and speed were calculated based on the noisy coordinates. The number of satellites was inputted into the state-space model for the semi-major axis of the error ellipse.

Finally, a delay was added for each of the components. The delays for the coordinates were one measurement, whereas they were three measurements for the other components.

4.6 Parameter estimation

In the navigation system, there are several parameters that must be tuned before the system is able to operate efficiently. Some of the parameters can be measured or identified easily, but some of them need more expertise so that they can be tuned by hand. In this section, several semi-automatic offline parameter estimation methods are described. With the described methods, the required level of expertise can be decreased.

4.6.1 Parameter types

The parameters in the navigation system can be classified into four categories:

- I. The parameters that cannot change during the operation and that can be directly measured using a measuring tape
- II. The parameters that usually do not change during the operation time, but that cannot be directly measured
- III. The parameters that can change during the operation time
- IV. The parameters of the actual control algorithm that usually are kept constant.

The first class of parameters is trivial. Those parameters can be measured once and then left untouched. The parameters that belong to this class are listed in Table 4.2.

Table 4.2. Constant physical parameters

Symbol	Unit	Description
a	m	The wheelbase of the tractor
b	m	The distance to the attachment point from the rear axle
c	m	The length of the drawbar
d	m	The distance to the seed coulters from the drawbar
lx	m	Lateral distance to the laser scanner from the centre of the seed drill
ly	m	Longitudinal distance to the laser scanner from the drawbar
$\max v $	m/s	Maximum speed
$\max \alpha $	rad	Maximum steering angle
$\max \beta $	rad	Maximum angle of the free joint
$\max \gamma $	rad	Maximum angle of the controlled joint

The second class of parameters must be estimated using some kind of estimator. The estimation process must be repeated every time that something is changed, for example when changing different implements. The parameters that belong to the second class are listed in Table 4.3.

Table 4.3. Constant estimated parameters

Symbol	Unit	Description
$\tau(x_{R,meas})$	ms	Delay in the position measurement (RTK-GPS)
$\tau(y_{R,meas})$	ms	Delay in the position measurement (RTK-GPS)
$\tau(\theta_{meas})$	ms	Delay in the heading measurement (Fibre Optic Gyroscope)
$\tau(v_{meas})$	ms	Delay in the speed measurement
$\tau(\alpha_{meas})$	ms	Delay in the steering measurement
$\tau(\beta_{meas})$	ms	Delay in the joint angle measurement
$\tau(\gamma_{meas})$	ms	Delay in the joint angle measurement
$\max \dot{v} $	m/s ²	Maximum acceleration
$\max \dot{\alpha} $	rad/s	Maximum steering rate
$\max \dot{\gamma} $	rad/s	Maximum joint angle rate
$\sigma(\theta_{meas})$	rad	Standard deviation of the angle measurement noise
$\sigma(v_{meas})$	m/s	Standard deviation of the speed measurement noise
$\sigma(\alpha_{meas})$	rad	Standard deviation of the steering measurement noise
$\sigma(\beta_{meas})$	rad	Standard deviation of the joint angle measurement noise
$\sigma(\gamma_{meas})$	rad	Standard deviation of the joint angle measurement noise

The third class of parameters must be initialised before the operation can be started, but it must also be updated during the operation. The parameters that belong to the third class are listed in Table 4.4.

Table 4.4. Time-varying estimated parameters

Symbol	Unit	Description
$\sigma(x_{R,meas})$	m	Standard deviation of the GPS position measurement
$\sigma(y_{R,meas})$	m	Standard deviation of the GPS position measurement
k_v	-	Dynamics of the speed
k_α	-	Dynamics of the steering
k_γ	-	Dynamics of the joint control
δ	-	Slipping factor of the front wheels

The fourth class of parameters cannot be measured or estimated. Certain rules of thumb can be used to tune these parameters, but generally these parameters are manually tuned once and then left constant. The parameters that belong to the third class are listed in Table 4.5.

Table 4.5. NMPC parameters

Symbol	Description
M	NMPC prediction horizon
$R_{u\{\dot{v}\}}$	Weight of the speed change
$R_{u\{\dot{\alpha}\}}$	Weight of the steering change
$R_{u\{\dot{\gamma}\}}$	Weight of the joint control
$R_{u\{v\}}$	Weight of the steady state speed
$R_{u\{\alpha\}}$	Weight of the steady state steering
$R_{u\{\gamma\}}$	Weight of the steady state joint
$Q_{\{x_R, y_R\}}$	Weight of the tractor lateral error
$Q_{\{x_E, y_E\}}$	Weight of the trailer lateral error
$Q_{\{\theta\}}$	Weight of the tractor angular error

From now on, the focus will be on how to estimate the second class of parameters and initialise the third class of parameters offline based on the recorded measurements. The same methods can be used partially online, in the operation time, to update the third class of parameters.

4.6.2 Estimation process

At the beginning of the tuning process, no valid assumption can be made about the initial values of these parameters. Therefore, the state estimator and closed-loop control cannot be used; it is not possible to guarantee the stability of the controlled system. Only the parameters of class I can be set according to manual measurements. It is also assumed that the process model presented in Section 3.4 is valid and that the unmodelled disturbances are random white noise with a zero mean.

The input and output signals for parameter estimation are generated using the open-loop control. It is assumed that the input signals are rich enough to produce sufficient information about the controlled system. For example, stepwise changes to the input signals with varying amplitudes can be used. The parameters of the navigation system are estimated offline from the recorded input and output signals.

The estimation procedure contains four steps:

1. Estimating the dynamics of the actuators and the delays in the actuator measurements;
2. Estimating the delays in the heading and position measurement and estimating the average of the slipping factor of the front wheel;
3. Simulating the remaining measurements and calculating the delays in the remaining measurements;
4. Calculating the standard deviation of the measurement noises.

Step 1: Estimating the dynamics of the actuators and the delays in the actuator measurements

The behaviours of the actuators are modelled using first-order difference models and discrete time delays (Equations 3.3 and 3.6). The change in the input signal is also limited using rate-limit values.

The delays in and the rate-limit values of the controls are estimated iteratively. The estimation process is presented via a pseudo code in Algorithm 4.6. One step of the iteration contains two phases: limiting the change in the control value and estimating the dynamics of the actuator using several different delay values. The rate-limit value and delay-time pair with a minimum MSE are chosen and the corresponding dynamic parameter is used.

The changes in the input control values are limited as follows:

$$u_l(t_{k+1}) = \begin{cases} u_l(t_k) - T \max |\dot{u}| & \text{if } u(t_{k+1}) - u_l(t_k) < -T \max |\dot{u}| \\ u(t_{k+1}) & \text{if } |u(t_{k+1}) - u_l(t_k)| \leq T \max |\dot{u}| \\ u_l(t_k) + T \max |\dot{u}| & \text{if } u(t_{k+1}) - u_l(t_k) > T \max |\dot{u}| \end{cases}, \quad (4.44)$$

where u_l is the limited control value, u is the original unlimited control value (v , α or γ), $\max |\dot{u}|$ is the maximum change of that particular control value and T is the sampling time, $t_{k+1} - t_k$.

The parameters of the dynamic models of the actuators (Equations 3.3 and 3.6) are estimated using the least-squares estimate:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y}, \quad (4.45)$$

where $\hat{\theta}$ is the parameter estimate (k_v , k_α or k_γ). The \mathbf{Y} and Φ matrices are arranged as follows:

$$\Phi = \begin{bmatrix} y(t_\tau) - u_l(t_1) \\ \vdots \\ y(t_{N-1}) - u_l(t_{N-\tau}) \end{bmatrix} \quad (4.46)$$

$$\mathbf{Y} = \begin{bmatrix} y(t_{\tau+1}) - u_l(t_1) \\ \vdots \\ y(t_N) - u_l(t_{N-\tau}) \end{bmatrix},$$

where y , u_l and τ are the corresponding measurement, limited control and delay values.

Algorithm 4.6. Estimate actuator dynamics

Input: $u(\cdot)$: control input sequence
 $y(\cdot)$: measurement sequence

Output: $\hat{\theta}$: estimated dynamic parameter
 $\max|\dot{u}|$: estimated rate limit value
 τ : estimated delay time

for $\max|\dot{u}| := \min$ **to** max rate
 $u_t = \text{limit } u$ according to equation 4.44 and current $\max|\dot{u}|$

for $\tau := \min$ **to** max delay
 $\hat{\theta} = \text{calculate}$ using equation 4.45
 $\hat{y} = \text{calculate}$ using current parameters and equation 3.3 or 3.6

MSE = calculate mean square error between y and \hat{y}
if MSE < best_estimate
use current parameter estimates as output candidates
end if

end for
end for

Step 2: Estimating the delays in the heading and position measurement and the slipping factor of the front wheel

The estimates for the delays in the heading and position measurement and the slipping factor of the front wheel are all based on the heading angle.

The change in the heading angle is estimated using the kinematic equation (3.1) and setting the slipping factor of the front wheels to one:

$$\hat{\theta} = \hat{v} \frac{\tan \hat{\alpha}}{a}, \quad (4.47)$$

where $\hat{\alpha}$ is the steering angle filtered from the control values using the dynamic equation for the steering (3.3) and \hat{v} is the speed filtered using the dynamic equation for the speed (3.3). The parameters for the dynamic equations are obtained from the previous step.

The delay in the heading measurement is estimated by minimising the squared difference between the measured ($\dot{\theta}_{meas}$) and estimated ($\hat{\theta}$) change in the heading:

$$\tau(\theta_{meas}) = \min_{\tau} \sum_k \left(\hat{\theta}(t_k) - \dot{\theta}_{meas}(t_{k+\tau}) \right)^2. \quad (4.48)$$

The position measurement delay is estimated similarly by minimising the squared difference between the measured heading and the heading derived from the position measurement:

$$\begin{aligned} \tau(x_{R,meas}) &= \tau(y_{R,meas}) \\ &= \tau(\theta_{meas}) + \min_{\tau} \sum_k \left(\theta_{meas}(t_k) - \theta_{pos}(t_{k+\tau}) \right)^2. \end{aligned} \quad (4.49)$$

The heading is derived from the position measurement as follows:

$$\theta_{pos} = \text{atan2}(\dot{y}_{R,meas}, \dot{x}_{R,meas}), \quad (4.50)$$

where $\dot{y}_{R,meas}$ and $\dot{x}_{R,meas}$ are the time derivatives for the position measurement.

Finally, the slipping factor of the front wheels is estimated by taking the average of the ratio between the estimated steering angle and the hypothetical steering angle from the heading measurement:

$$\delta = \frac{1}{N} \sum_{k=1}^N \left(\frac{\arctan \left(a \frac{\dot{\theta}_{meas}(t_{k+\tau(\theta_{meas}))}}{\hat{v}(t_k)} \right)}{\hat{\alpha}(t_k)} \right), \quad (4.51)$$

where N is the total number of used sample points.

Step 3: Simulating the remaining measurements and calculating the delays in the remaining measurements

The only measurement that is left is the freely moving angle between the tractor and the trailer. The change in the freely moving joint is estimated using the joint model:

$$\hat{\beta} = \frac{-a\hat{v} \sin(\beta+\hat{\gamma}) + \hat{v}(d+c \cos \beta + b \cos(\beta+\hat{\gamma})) \tan \hat{\alpha} - a d \hat{\gamma}}{a(d+c \cos \hat{\gamma})}. \quad (4.52)$$

The delay in the joint angle measurement is estimated by minimising the squared difference between the measured ($\hat{\beta}_{meas}$) and estimated ($\hat{\beta}$) change of the angle

$$\tau(\beta_{meas}) = \min_{\tau} \sum_k \left(\hat{\beta}(t_k) - \hat{\beta}_{meas}(t_{k+\tau}) \right)^2. \quad (4.53)$$

Step 4: Calculating the standard deviations of the measurement noises

In the previous steps, all measurements are estimated using kinematic and dynamic equations and the parameter estimates obtained from the measurements. Also, the delays in the measurements are estimated. It was assumed that the remaining difference between the measurement and the estimate is white noise. Therefore, the measurement noises are calculated as follows:

$$\sigma(y) = \sqrt{\frac{1}{N} \sum_{k=1}^N \left(y(t_k) - \hat{y}(t_{k+\tau(y)}) \right)^2}, \quad (4.54)$$

where y is the measurement, \hat{y} is the estimate of the measurement and $\tau(y)$ is the discrete time delay of the corresponding measurement.

5. Results

The previous chapters presented the proposed methods for an experimental navigation system and the vehicle and positioning device configurations used in the experiments. In this chapter, the results of the experiments are presented. Two vehicle and positioning device configurations were used in this thesis. The results are for vehicle configuration A-I and vehicle configuration B-II.

The parameter estimation results, which are presented first, are for vehicle configuration B-II; they do not include Section 5.1.3, 'Parameters used in the tests', which refer to Section 5.3, 'Path tracking'. Section 5.3 presents the results related to path tracking, including the accuracy, state estimation and computational capacity of the system. Those results are mainly for vehicle configuration A-I. However, the collision avoidance results are for vehicle configuration B-II. Before discussing the path tracking results, the results for the path generation methods are presented, since the tracking methods depend on a path. Path generation is not dependent upon vehicle configuration, since the physical limits of the steering actuators and the wheelbases of both tractors are the same.

5.1 Parameters of the system

The input signals for the calibration sequence were left undefined in Section 4.6 due to differences in the hardware capabilities of the different vehicles. With the equipment used in this thesis, the calibration sequence contains a step response test for all controls separately with different step sizes (Figure 5.1). The step response tests for steering and joint control were realised at a nominal driving speed (2 m/s). From the calibration sequence, the parameters of the system are estimated automatically.

The proposed methods were tested both in a simulator, where the parameters were known, and with real equipment, where the parameters were unknown.

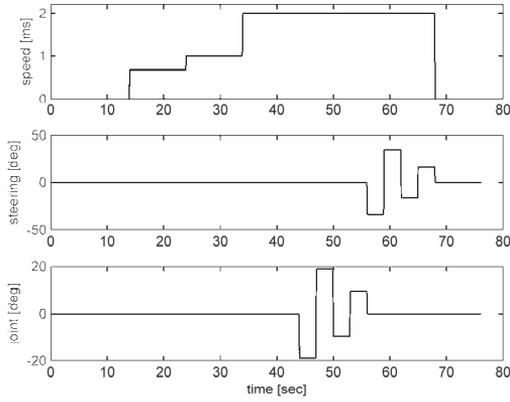


Figure 5.1. Input control signals for identification: speed, steering and joint control trajectories from top to bottom respectively.

5.1.1 Estimated parameters in simulator

The simulator test consisted of two different sets of tests. The parameters of the simulator were modified for the two different test sets. Both test sets contained ten different calibration sequences. The results of the calibrations are listed in Table 5.1, which contains the means and the standard deviations of each parameter together with their true values.

Table 5.1. Parameter estimation results in the simulator.

Symbol	Test 1		Test 2	
	Est.	True	Est.	True
$\tau(x_{R,meas})$	630 (245)	400	590 (233)	500
$\tau(y_{R,meas})$				
$\tau(\theta_{meas})$	340 (52)	300	390 (32)	400
$\tau(v_{meas})$	440 (51)	300	510 (57)	400
$\tau(\alpha_{meas})$	360 (52)	300	410 (32)	400
$\tau(\beta_{meas})$	660 (52)	500	660 (52)	600
$\tau(\gamma_{meas})$	1320 (63)	500	1720 (79)	600
$\max \dot{v} $	0.5 (0.0)	0.5	0.7 (0.0)	0.7
$\max \dot{\alpha} $	0.5 (0.0)	0.48	0.79 (0.03)	0.72
$\max \dot{\gamma} $	5.41 (1.77)	2.2	6.6 (0.0)	2.2
$\sigma(\theta_{meas})$	0.003097 (0.002567)	0	0.009196 (0.001206)	0.01
$\sigma(v_{meas})$	0.022341 (0.001409)	0.02	0.024166 (0.000768)	0.02
$\sigma(\alpha_{meas})$	0.012944 (0.000827)	0.01	0.013791 (0.001971)	0.01
$\sigma(\beta_{meas})$	0.075294 (0.002587)	0.045	0.113412 (0.005169)	0.063
$\sigma(\gamma_{meas})$	0.057122 (0.004257)	0.032	0.066766 (0.001791)	0.045
k_v	0.880474 (0.006740)	0.9	0.712917 (0.024116)	0.8
k_α	0.544548 (0.082713)	0.67	0.441449 (0.095754)	0.4
k_γ	0.821790 (0.016349)	0.9	0.770314 (0.033090)	0.9
δ	0.895788 (0.021277)	1	0.786503 (0.020340)	0.8

5.1.2 Parameters of the test configuration

With Vehicle configuration B-II, described in Chapter 3.1 and 3.2, the proposed methods were tested several times under different conditions. The results of the three different tests are listed in Table 5.2. The first calibration test was performed in a grass field and the seed drill was in a transport state. The second calibration test was performed in a threshed crop field and the seed drill was in a working state. The third calibration test was performed in a harrowed field and the seed drill was again in a working state. The standard deviation of the position measurement, ($\sigma(x_{R,meas})$ and $\sigma(y_{R,meas})$), was not estimated since the noise estimate (GNSS Pseudo noise statistics) offered by the RTK-GPS receiver was used directly.

Table 5.2. Test results with the real system.

Symbol	Test 1 Est.	Test 2 Est.	Test 3 Est.
$\tau(x_{R,meas})$	200 [ms]	300 [ms]	300 [ms]
$\tau(y_{R,meas})$			
$\tau(\theta_{meas})$	100 [ms]	200 [ms]	200 [ms]
$\tau(v_{meas})$	500 [ms]	600 [ms]	100 [ms]
$\tau(\alpha_{meas})$	100 [ms]	100 [ms]	200 [ms]
$\tau(\beta_{meas})$	200 [ms]	200 [ms]	100 [ms]
$\tau(\gamma_{meas})$	100 [ms]	200 [ms]	200 [ms]
$\max \dot{v} $	0.5 [ms/s ²]	0.4 [ms/s ²]	0.4 [ms/s ²]
$\max \dot{\alpha} $	0.4 [rad/s]	0.4 [rad/s]	0.4 [rad/s]
$\max \dot{\gamma} $	1.9 [rad/s]	2.2 [rad/s]	2.2 [rad/s]
$\sigma(\theta_{meas})$	0.005672	0.005553	0.004272
$\sigma(v_{meas})$	0.065747	0.106305	0.123000
$\sigma(\alpha_{meas})$	0.019499	0.011546	0.018715
$\sigma(\beta_{meas})$	0.038527	0.029015	0.036244
$\sigma(\gamma_{meas})$	0.012112	0.009382	0.012608
k_v	0.871893	0.915468	0.944871
k_α	0.657935	0.508497	0.573319
k_γ	0.851840	0.837685	0.826468
δ	0.750245	0.743750	0.741343

5.1.3 Parameters used in the tests

The path tracking accuracy test described in Section 5.3 was performed with Vehicle configuration A-I. The controllers for Vehicle configuration A-I were first tested and tuned in a simulation environment and the final tunings were then performed by hand in real world environment and with the actual hardware.

The physical dimensions of the vehicle were as follows:

$$\begin{aligned}
 a &= 2.8 \text{ [m]} & l_x &= 2.7 \text{ [m]} \\
 b &= 1.7 \text{ [m]} & l_y &= 1.48 \text{ [m]} \\
 c &= 2.3 \text{ [m]} & p_x &= 1.1 \text{ [m]} \\
 d &= 3.3 \text{ [m]} & p_y &= 1.48 \text{ [m]}.
 \end{aligned}$$

The physical limitations of the control variables and joint angles

$$\begin{aligned} \max|\dot{v}| &= 1 \text{ [m/s}^2\text{]} & \max|v| &= 5 \text{ [m/s]} & \max|\beta| &= 1.57 \text{ [rad]} \\ \max|\dot{\alpha}| &= 0.7 \text{ [rad/s]} & \max|\alpha| &= 0.7 \text{ [rad]} & \max|\gamma| &= 0.33 \text{ [rad]} \\ \max|\dot{\gamma}| &= 0.33 \text{ [rad/s]} & & & & \end{aligned}$$

and the parameters of the control dynamics were

$$k_v = 0.88 \quad k_\alpha = 0.54 \quad k_\gamma = 0.82.$$

The standard deviations of the state variables and measurements were:

$$\begin{aligned} \sigma(x_R) &= 0.002 \text{ [m]} & \sigma(x_{R,meas}) &= 0.03 \text{ [m]} \\ \sigma(y_R) &= 0.002 \text{ [m]} & \sigma(y_{R,meas}) &= 0.03 \text{ [m]} \\ \sigma(\theta) &= 0.00002 \text{ [rad]} & \sigma(\theta_{meas}) &= 0.0035 \text{ [rad]} \\ \sigma(v) &= 0.00007 \text{ [m/s]} & \sigma(v_{meas}) &= 0.000067 \text{ [m/s]} \\ \sigma(\alpha) &= 0.009 \text{ [rad]} & \sigma(\alpha_{meas}) &= 0.0066 \text{ [rad]} \\ \sigma(\delta) &= 0.00001 & \sigma(\beta_{meas}) &= 0.0055 \text{ [rad]} \\ \sigma(\beta) &= 0.000001 \text{ [rad]} & \sigma(\gamma_{meas}) &= 0.0002 \text{ [rad]} \\ \sigma(\gamma) &= 0.000002 \text{ [rad]} & \sigma(x_{E,meas}) &= 0.038 \text{ [m]} \\ \sigma(x_E) &= 1 * 10^{-10} \text{ [m]} & \sigma(y_{E,meas}) &= 0.038 \text{ [m]} \\ \sigma(y_E) &= 1 * 10^{-10} \text{ [m]}. & & \end{aligned}$$

The measurement delays were as follows:

$$\begin{aligned} \tau(x_{R,meas}) &= 300 \text{ [ms]} & \tau(v_{meas}) &= 100 \text{ [ms]} & \tau(\gamma_{meas}) &= 200 \text{ [ms]} \\ \tau(y_{R,meas}) &= 300 \text{ [ms]} & \tau(\alpha_{meas}) &= 100 \text{ [ms]} & \tau(x_{E,meas}) &= 0 \text{ [ms]} \\ \tau(\theta_{meas}) &= 500 \text{ [ms]} & \tau(\beta_{meas}) &= 200 \text{ [ms]} & \tau(y_{E,meas}) &= 0 \text{ [ms]} \end{aligned}$$

The weights of the NMPC controller were experimentally studied. The following weights were used in the test drives:

$$\begin{aligned} R_{\dot{u}_{\{v\}}} &= 0.02 & R_{u_{\{v\}}} &= 20 & Q_{\{x_R, y_R\}} &= 0.1 \\ R_{\dot{u}_{\{\alpha\}}} &= 0.004 & R_{u_{\{\alpha\}}} &= 0.04 & Q_{\{x_E, y_E\}} &= 0.005 \\ R_{\dot{u}_{\{\gamma\}}} &= 0.004 & R_{u_{\{\gamma\}}} &= 0.001 & Q_{\{\theta\}} &= 0.1. \end{aligned}$$

The parameters that affect the stability and that are partially caused by the underlying ISO 11783 network are the delays. The time-delay and time-constant parameters of the system, which were identified experimentally, are listed again in Table 5.3. The table also shows the calculated maximum latencies of the corresponding messages in the ISO 11783 network that were used in the identification process. The theoretical maximum latencies were calculated using equations provided by Tindell et al. (1995). The impact of the CAN-bus latency was insignificant compared to the total dynamic delay. However, because the ISO 11783 network is an open system and a farmer can connect any number of machines to it, the maximum latencies cannot be guaranteed in every circumstance. The values in Table 5.3 were calculated for a test configuration where the CAN bus load was approximately 30%.

Table 5.3. Some of the identified time-delay and time-constant parameters of the system and theoretical maximum latencies of the corresponding control and measurement messages.

Description	Identified		CAN latencies	
	Time-Delay	Time-constant	control	meas.
Position	300 ms	-	-	1.0 ms
Heading	500 ms	-	-	1.0 ms
Speed	100 ms	740 ms	6.5 ms	16.5 ms
Steering	100 ms	120 ms	2.0 ms	1.5 ms
Free joint angle	200 ms	-	-	17.5 ms
Controlled Joint angle	200 ms	450 ms	13.0 ms	17.5 ms

5.2 Path generation

The path generation methods were evaluated by first comparing the different turning types with Dubins' Curves. Then, one complete agricultural operation was reported and extra attention was given to the case in question, where the curvature was limited.

5.2.1 Headland types

Figure 5.2 provides an illustration of three different turning scenarios. The first scenario corresponds to normal turning into the adjacent row. The second scenario corresponds to turning over several rows. The last scenario could only happen by changing the row in the middle of the field or, for example, by changing from one subfield to another. The blue line was generated using the Spiral Connection method, whereas the red line was generated using Dubins' Curves.

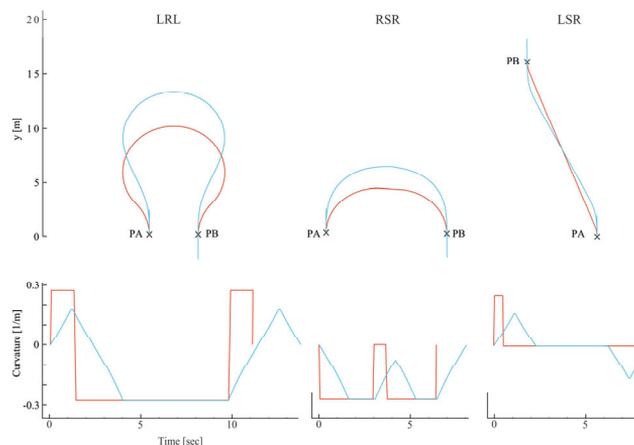


Figure 5.2. Comparison of different turning types: LRL, RSR and LSR. The red line is generated by using Dubins' Curves and the blue line using Spiral Connection method.

The calculation time and the path length were further analysed by generating turns between 1000 randomly chosen starting and ending points using the following parameters: $\max|\alpha| = 0.65$, $\max|\dot{\alpha}| = 0.4$, $a = 2.8$ and $dt = 0.1$. The average calculation time of Dubins' Curves was approximately 2 ms, and with the Spiral Connection method, it was approximately 35 ms

with a non-optimized Matlab code. The average ratio between the path lengths was 1.14, meaning that the headland turnings when using the Spiral Connection methods were, on average, 14% longer than Dubins' Curves. In the worst case, the Spiral Connection path was 25% longer than the Dubins' Curves path.

5.2.2 Complete field

Figure 5.3 provides an illustration of one complete agricultural operation on a real field: seeding with towed implement. In this particular case, the field was first driven around seven times to insure sufficient space in the headland. After that, the inner area of the field was operated by driving back and forth, always turning into the adjacent swath. The north-west corner of the field is enlarged here to emphasise the effect of the path smoothing. Also, the turnings at the south end of the field were enlarged to emphasise the Spiral Connection method.

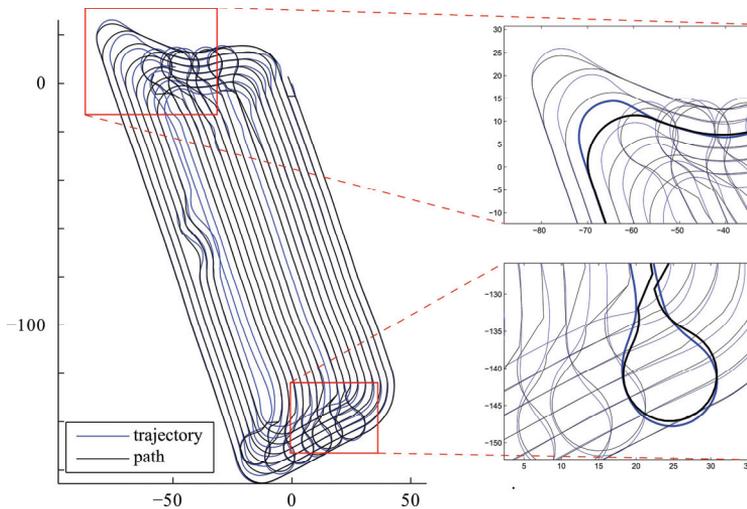


Figure 5.3. Driven trajectories (blue) and generated path (black) on a real field.

With the test equipment and with the presence of slipping, the maximum curvature that the tractor is capable of driving is 0.14 m^{-1} , meaning a 23 degree steering angle with a 2.8 m wheelbase. If the desired distance to the adjacent swath is 3 m, then, according to Equation 4.25, the maximum curvature of the followed path is 0.10 m^{-1} . Figure 5.4 shows just a part of the entire path that is shown in Figure 5.3. The original path curvature was -0.14 m^{-1} at most, meaning that the tractor was turning full right. The smoothed path curvature was -0.10 m^{-1} at most. Along the original path, the radius of the turning circle was 7.1 m, whereas the radius was 10 m along the smoothed path. Therefore, if the following distance is 3 m, the

radius of the turning circle will remain the same for approximately the 7 m that the tractor is capable of driving.

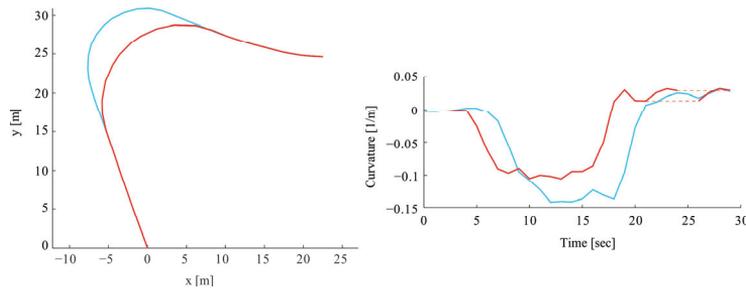


Figure 5.4. Path smoothing in the corner of the field. The original path is blue and path with limited curvature is red. Corresponding curvatures are shown in the image on the right.

5.3 Path tracking

The NMPC-based path tracking methods were compared to traditional, geometric-type methods, which are described in PUB I and referred here as Target Point (TP) algorithm. Geometric-type path tracking methods are most commonly used in mobile robotics (Snider, 2009). After that, the results of the estimation methods based on the local measurements are shown. Finally, the computational capacity is examined.

5.3.1 Comparison to traditional path tracking

The different path tracking methods were compared during two test procedures. In the first test, denoted in this thesis as the ‘straight path’ procedure, the driver first made straight driving line, turned onto a headland path and then switched the guidance system on. After that, the speed was kept constant and the previous driving line was followed for 30 metres. In the second test, denoted in this study as the ‘curved path’ procedure, the driver started the test in the same way as in the first procedure, by making the first driving line. At this time, the driving line was curved with a 50 metre wavelength and a 4 metre amplitude (Figure 5.5). The guidance system followed this particular curved driving line for the next four driving lines.

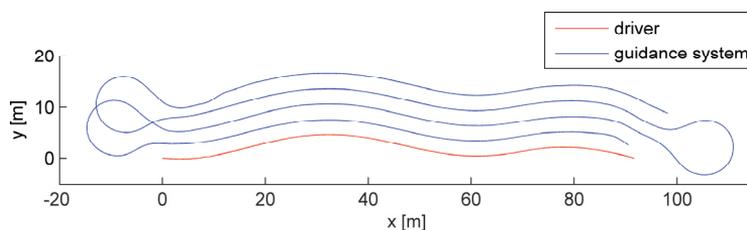


Figure 5.5. Trailer trajectory in the curved driving line.

The results of the first tests are presented in three different figures. The figures represent the tracking errors of the tractor (Figure 5.6) and the trailer (Figure 5.7), which were calculated using the state estimates according to method described in Chapter 4.1.1, i.e. the control error, and the distance to the adjacent driving line (Figure 5.8), which was calculated using the raw GPS measurements, i.e. the absolute error. The measurements are illustrated in the form of a box-and-whiskers plot. The box in the plot represents the middlemost half of the data and the line inside the box represents the median value. The whiskers represent the smallest value and the largest value. The crosses represent outliers. The tracking errors were calculated in real time using the state estimation of the tractor-trailer system, and the same measurements were used as error values in the controllers. The distance to the adjacent driving line was calculated afterwards using the raw VRS-GPS measurements. The target distance, or the working width, was 2.95 m.

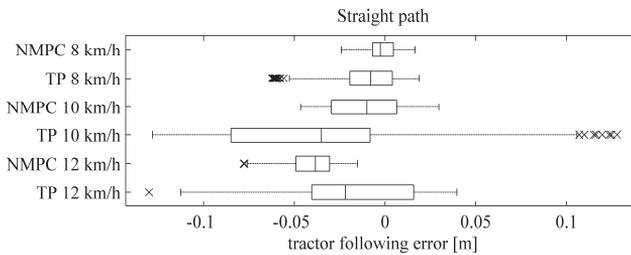


Figure 5.6. Box-and-whiskers plot of the tracking errors of the tractor in straight-path-following tests.

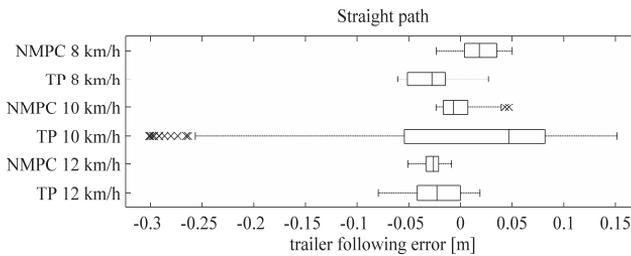


Figure 5.7. Box-and-whiskers plot of the tracking error of the trailer in straight-path-following tests.

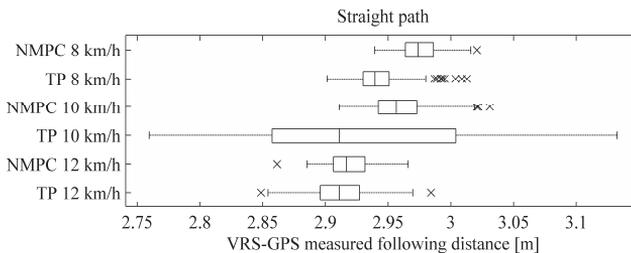


Figure 5.8. Box-and-whiskers plot of the distance to the adjacent driving line measured using VRS-GPS in straight-path-following tests.

The results of the second tests are given in same way as the results from the first test: the tracking error of the tractor (Figure 5.9) and the trailer (Figure 5.10) and the distance to the adjacent driving line (Figure 5.11). All of the measurements were taken under steady-state conditions after making the transition from the headland path.

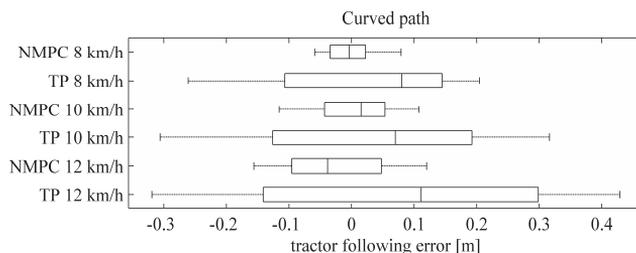


Figure 5.9. Box-and-whiskers plot of the tracking error of the tractor in curved-path-following tests.

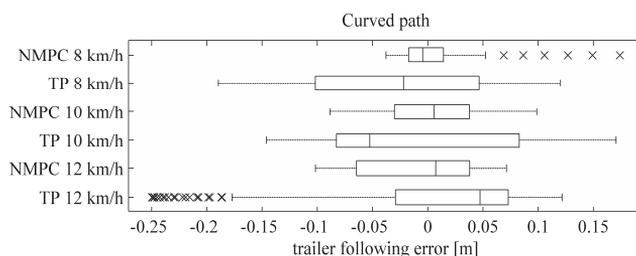


Figure 5.10. Box-and-whiskers plot of the tracking error of the trailer in curved-path-following tests.

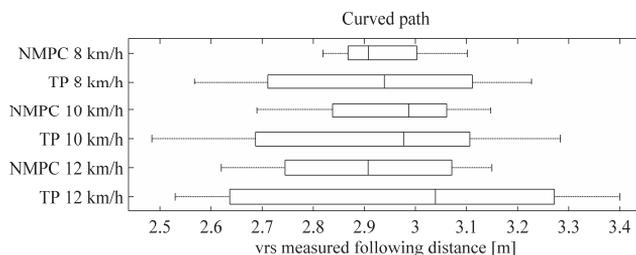


Figure 5.11. Box-and-whiskers plot of the distance to the adjacent driving line measured using VRS-GPS in curved-path-following tests.

5.3.2 Estimate correction with laser scanner measurements

The position estimation with and without laser scanner measurements was analysed by doing similar test drives as in the path tracking comparison. The driving speed was 8 km/h in every test. For clarity, unreliable laser scanner measurements were removed from the pictures and from the mean error calculations as well.

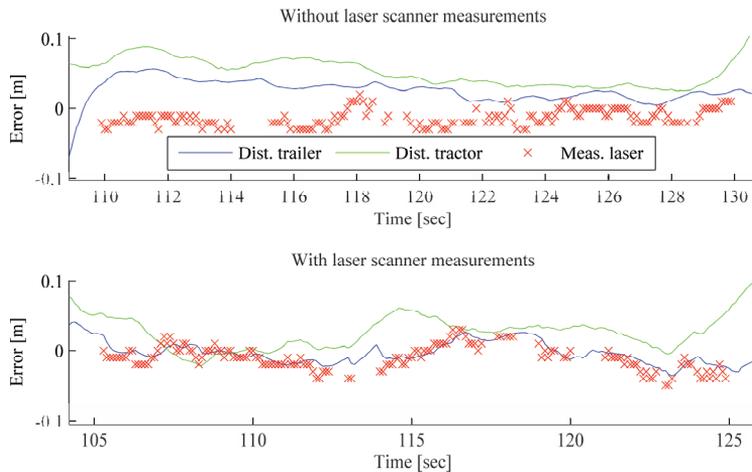


Figure 5.12. Tracking errors in straight-line-following tests without (on the top) and with (on the bottom) laser scanner measurements in the Kalman filter. The errors were measured using a laser scanner and VRS-GPS.

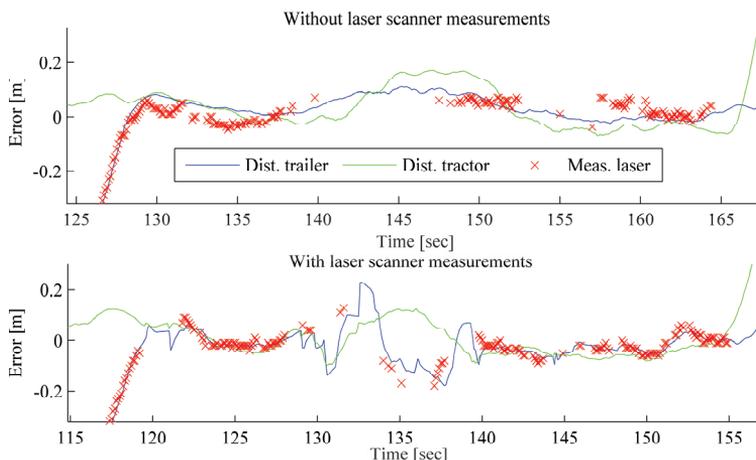


Figure 5.13. Tracking errors in curved-line-following tests without (on the top) and with (on the bottom) laser scanner measurements in the Kalman filter. The tractor went over the mark-furrow and the laser scanner was unable to measure the distance for the whole time.

Table 5.4 presents the same results as Figure 5.12 (the straight-line-following test) and Figure 5.13 (the curved-line-following test). Table 5.4 also shows the mean and the variance of the calculated tracking errors for the tractor and the trailer. The column labelled ‘Laser’ consists of tracking errors calculated directly from the laser scanner measurements. The column labelled ‘Laser Diff.’ is the calculated difference between the estimated tracking error of the trailer and the laser scanner measurements. The column labelled ‘VRS Diff.’ is the calculated difference between the estimated tractor positions and the true delayed measurements. The statistical values for the laser scanner measurement and the difference between the laser and estimated trailer error were calculated only using the values obtained from acceptable laser scanner measurements, whereas the other values were calculated using the whole time slot.

Table 5.4. Path tracking and state estimation results in statistical form.

	Laser	Laser OK [%]	Tractor $\mu(\sigma)$ [m(m)]	Trailer $\mu(\sigma)$ [m(m)]	Laser $\mu(\sigma)$ [m(m)]	Laser Diff. $\mu(\sigma)$ [m(m)]	VRS Diff. $\mu(\sigma)$ [m(m)]
Straight line	ON	77.5	0.022 (0.020)	-0.004 (0.014)	-0.010 (0.017)	0.005 (0.010)	0.001 (0.013)
	OFF	85	0.051 (0.028)	0.028 (0.013)	-0.013 (0.012)	0.041 (0.020)	0.006 (0.011)
Curved line	ON	52.8	0.002 (0.063)	-0.037 (0.108)	-0.056 (0.108)	-0.005 (0.024)	0.003 (0.022)
	OFF	49.2	0.031 (0.073)	0.021 (0.076)	-0.007 (0.102)	0.004 (0.054)	0.011 (0.022)

5.3.3 Collision avoidance

In this thesis, a plastic tube with foam covering was used as an artificial obstacle (Figure 5.14). The size and shape of the tube are equivalent to an electricity pole. Since the obstacles are considered to be mainly electricity poles, the maximum distance from the measurement point to the cluster centre in the clustering algorithm was set at 0.3 m. The maximum iteration time in the clustering algorithm was set at 10 iterations. The obstacle was considered to be confident if it was seen 20 times and if the detection counting was stopped when the obstacle had been seen 300 times. This means that obstacle recognition takes at least 267 milliseconds and that a reliable obstacle has to be within sight of the scanner and not associated with any cluster for at least 4 seconds, until it is removed.



Figure 5.14. Test obstacle in front of the tractor.

Using the settings mentioned above, the obstacle was recognised and added to known obstacles approximately 8-10 metres ahead of the tractor. For this reason, the nominal avoiding distance, D , was set at 6 metres in tractor-alone navigation and at 8 metres in combined navigation. This means that the minimum allowed distance between the tractor and the

obstacle was 1.5 metres (3 metres from the centre), whereas it was 2.5 meters (4 meters from the centre) between the trailer and the obstacle.

The obstacle avoidance method was tested with tractor-alone navigation and with combined tractor-implement navigation. In the tests, the speed was varied from 2 m/s to 3.5 m/s (7.2 km/h to 12.6 km/h).

Table 5.5 lists the standard deviations of the recognized pole positions and also the maximum deviations from the mean values. The standard deviation was below 10 cm and the maximum deviation was below 50 cm at all of the tested speeds.

Table 5.5. Deviations of the recognized pole positions.

	Driving speed			
	2 m/s	2.5 m/s	3 m/s	3.5 m/s
std [m]	0.07	0.08	0.07	0.04
max [m]	0.30	0.39	0.48	0.20

Table 5.6 lists the minimum distance to the pole and the size of the gap in tractor-alone navigation. The driven trajectories, recognized pole position and minimum distance to the pole are illustrated in Figure 5.15

Table 5.6. The minimum distance to the pole and the size of the gap in tractor-alone navigation.

	Driving speed			
	2 m/s	2.5 m/s	3 m/s	3.5 m/s
minimum distance	3.2 m	2.9m	3.0 m	2.9m
The size of the gap	25.6 m ²	31.5 m ²	26.9 m ²	39.7 m ²

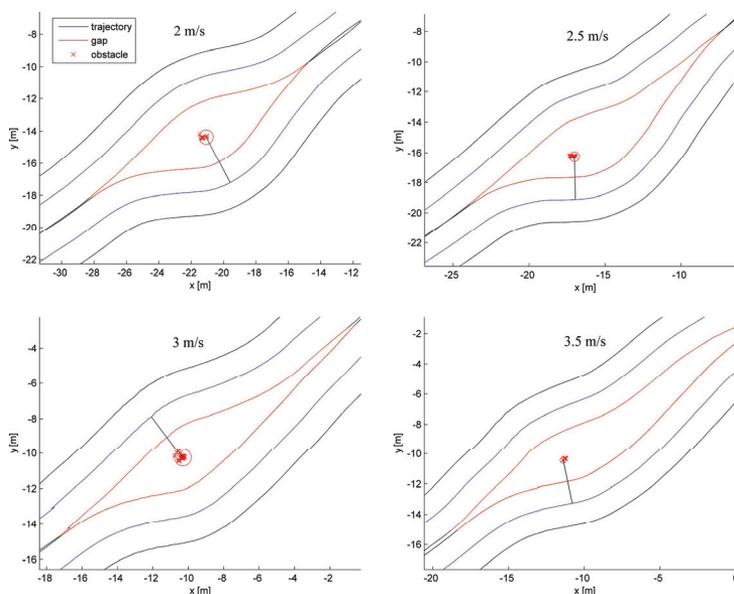


Figure 5.15. Collision avoidance results with different speeds in tractor-alone navigation.

Using combined tractor-implement navigation, collision avoidance was tested at a nominal working speed. In this thesis, the nominal working

speed was considered to be about 2.5 m/s, which is typical for seeding applications. Figure 5.16 presents an illustration of two different collision avoidance manoeuvres with a towed implement. The size of the unworked area on the left side of Figure 5.16 is 28.9 m² and on the right side it is 31.5 m². The size of the overlapping area on the left side of Figure 5.16 is 17.8 m² and on the right side it is 32.4 m².

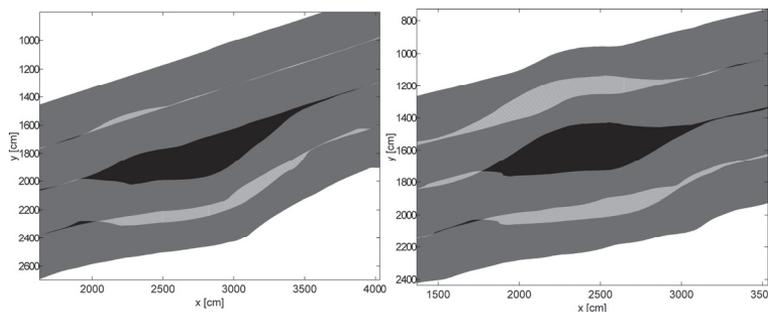


Figure 5.16. Worked area in tractor-implement navigation with collision avoidance. The areas that are worked once are depicted in gray, the unworked areas are depicted in black and the overlapping areas are depicted in light gray

5.3.4 Prediction horizon length in the optimization

With the navigation computer used in this study, which was powered by a Core 2 Duo E8600 processor and 2GB memory, the prediction horizon was changing from a 10 time steps to 30 time steps when using a 100 ms control cycle (Figure 5.17). Especially in the headland, the prediction horizon was reduced to a minimum.

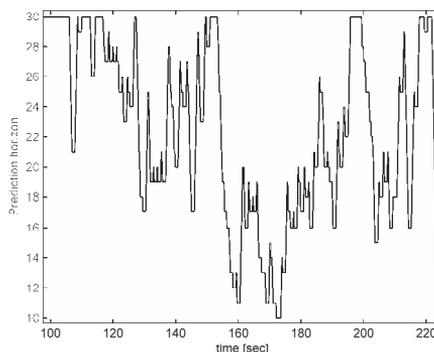


Figure 5.17. The evolution of the prediction horizon in a typical path. The path included a headland turning between 160 and 180 seconds.

Path tracking accuracy was tested using straight driving lines and curved driving lines at different speeds. The mean lengths of the NMPC prediction horizon in the different tests are listed in Table 5.7. The prediction horizon was not reduced from the maximum only in the straight driving line test at a speed of 8 km/h. In all other tests, the computation time for the NMPC

with a full-length prediction horizon took longer than the control cycle and the horizon had to occasionally be reduced.

Table 5.7. Mean length of the prediction horizon under different conditions.

	Driving speed			
	8 km/h	10 km/h	12 km/h	14 km/h
Straight line	30	29.8	28.0	-
Curved line	28.2	27.6	26.0	24.6

6. Discussion

This study shows that a combined navigation system can be built on top of the ISO 11783 network and that the architecture itself works. The results show that the accuracy of the combined navigation system is better with an NMPC than with using two separate SISO controllers; one for tractor and one for implement. Furthermore, the results show that the navigation system can accomplish the entire agricultural operation, including the driving line determination and obstacle avoidance.

The parameters that affect the stability and that are partially caused by the ISO 11783 network are the delays. In Section 5.1.3, it was shown that the delays caused by the communication network are not crucial compared to the dynamic time constants of the system.

In this chapter, each part of the navigation system is further discussed. Furthermore, this chapter contains notes on the results.

6.1 Parameter estimation

The results show that the proposed methods give reasonably accurate parameter estimates for the controlled system. Usually, however, the estimated values are greater than the true values. This is natural, especially with the noise parameters, because it is assumed that the model is perfect and that no internal noise of the process is present. In reality, the noise is divided between measurement and process noises. However, in practice the ratios between the different measurement noises are more important than the absolute values within the EKF.

Other drawbacks of the proposed methods are that the estimated delay-times are multiples of the sampling time and that the calibration sequence demands quite a large open area. However, the driver is usually present to make sure that the calibration can safely be performed, and there are usually some fields on the farm that can be used. Also, the discrete delay-time steps seemed to be sufficient for the control system to work properly.

The weight matrices were tuned by hand using rules of thumb. The weights were set such that they equalised the average error of each state, which are interdependent. For example, the costs caused by the permissible

heading error and the permissible position error of the tractor are the same order of magnitude regardless the quantity of the unit. The weight ratios between the interdependent variable sets (for example, tractor vs. implement) had to be found manually so that the desired system behaviour could be achieved. In practice, the method worked well and the tuning was intuitive at the end.

6.2 Path generation

The results show that the path generation methods work both in theory and in practice. Although the path is not solved analytically, it is still faster to calculate it using proposed methods than with numerical optimisation methods. The iteration times are bounded to be, at maximum, the number of half spiral elements squared in the LSL and RSR turnings, which constitute the worst-case scenarios. Given the parameters used in the experiments, this means 289 iterations in the worst-case scenario. The comparison of the calculation time and path length showed that the Spiral Connection method takes about 15 times longer to calculate than Dubins' Curves, but the calculation time is still short even when the code is not optimised. The path length with the Spiral Connection method is naturally longer than with Dubins' Curves, being at maximum 24 % longer in the worst-case scenario.

The steering rate was constrained by using the maximum derivative of the steering angle. Other solutions found in the existing literature use the maximum derivative of the curvature. The proposed solution can be modified to limit the steering rate by any function that is dependent on the current steering angle. In reality, however, the steering actuator is a dynamical system and the steering rate cannot change infinitely fast. Therefore, the second derivative of the curvature should also be taken into account. However, the impact of the second derivative would be negligible. Also, the actual maximum physical steering angles and steering rates are not used in the path planning. Instead, a small offset is used in order to leave the controller enough space to work within the limits of the physical system.

In this thesis, the objective of the path planning method was limited to convex field plots only. However, the method also works in practice for non-convex field plots. The field reported in the results is not convex, but it can be covered by this algorithm. Also, the algorithm can be extended to support most of the field types by combining it, for example, with the split-and-merge algorithm (Oksanen and Visala, 2009), where the field is first divided into convex subfields. The Spiral Connection method and path

smoothing can also be used separately with different path planning methods.

6.3 Path tracking

The first objective of this thesis was to develop a navigation system for a tractor-trailer system, which is able to drive at least 12 km/h with less than 10 cm lateral error. The results show that the goal was reached most of the time. However, there are situations where the controller was not able to keep the lateral error within the accepted range. This was caused mainly by the slow dynamics of the drawbar and uncertainties in the measurements. Also, the mechanics of the vehicle were not designed to support more accurate navigation. As it can be seen in Figure 6.1, there was major backlash, e.g. in the implement mounting.



Figure 6.1. The mounting of the implement allows the implement to move a few centimetres sideways.

The results also show the superiority of the proposed method compared to the simple geometrical path tracking and separate implement control methods. However, the better accuracy does come with the cost of making the system more complex. The algorithm presented in this study is computationally a lot heavier and not as reliable as the comparison methods. It is advisable to implement a backup method, which would be simpler and more robust but not necessarily useful for accurate path tracking. The backup system could take over the control when the NMPC controller fails in real-time.

The computational capacity can be kept at the maximum level with the variable prediction horizon length. The results show that with the full prediction horizon length, it occasionally takes more time to find the solution than the control cycle time. This implies that the NMPC controller cannot be used to realise real-time control in a tight loop without having any external interruption or backup method to ensure strict time limits. Also, a permanent reduction in the prediction horizon could lead to decreased accuracy in navigation.

The method for estimating the state seemed to be sufficient for the required path tracking accuracy. As shown in the results, the controller was able to keep the errors within the same ranges in both cases: when the laser

correction was on and when it was off. But the standard deviation was much smaller in the straight-line-following tests than in the curved-line-following tests. The most interesting values are the mean values and the standard deviations of the difference between the estimated trailer errors and laser scanner measurements (Table 5.4, column 'Laser Diff.'). The deviation is more significant when the laser scanner correction is off. If it is assumed that laser scanner measurement is correct, then this implies that the estimated position of the trailer is not correct when the laser scanner is not used. By using the laser scanner, a more reliable position estimation can be achieved. It is also likely, that the laser scanner helps the position estimation when the GPS signal is temporally not available, for example when the tree canopy blocks the direct view of the sky. However, if the implement width is the same as the tractor width, then it is possible that tractor tyres will go over the marking furrow. This results either in an erroneous measurement or in no measurement at all. With a wider implement, these situations might occur less frequently. Also, with different types of agricultural operations, a different marking and recognition process should be considered.

Because of the delays, the estimation method predicts the future state of the system. The standard deviation of the difference between the estimated positions and the VRS-GPS measurements (Table 5.4, column 'VRS Diff.')

are of the same order of magnitude as the VRS-GPS accuracy by itself, so it cannot be known for sure which of the two is more accurate without using other additional measurements. But it can be said that the prediction ahead does not increase the error or reduce the accuracy.

With the NMPC controller, the deviation of the tractor from the path was also taken into account in the cost function, although this was not the objective of the control problem. This was done because the stability is remarkably better with the cost of the tractor than without it. Also, the weight ratio between the tractor and trailer was empirically established in order to obtain sufficient tracking accuracy for the trailer on the one hand and stable behaviour of the tractor on the other. An alternative final state constraint or infinite prediction horizon would also work, but those are impractical to implement.

The cost function of the NMPC controller was implemented in a similar manner as the commonly used potential field method (Murphy, 2000). The difference is that the NMPC controller predicts the future states and the cost function is calculated in several positions ahead of the tractor's current position. In this way, the tractor is able to follow the path accurately along an almost optimal trajectory with respect to the physical constraints.

Moreover, the avoidance of static obstacles using the scheme presented in this study.

6.4 Collision avoidance

Collision avoidance was divided into two different subproblems: detecting the obstacle and avoiding the obstacle.

The obstacles were detected using the 2D laser scanner measurements with the help of a clustering algorithm. There was also a list of recognised obstacles, which reduced false positive and false negative recognitions. Overall, the obstacles were recognised quite accurately. The standard deviation of the recognised pole positions was less than 10 centimetres at all test speeds.

The obstacle avoidance method was included to the path tracking. Because the computational capacity had already been exhausted without the collision avoidance, the form of the original NMPC controller was left unchanged. The solution was to modify the cost function near an obstacle. An artificial avoided area was created, where the obstacles are not allowed to be. If there was an obstacle inside the avoided area, then the cost of the path was changed to the cost of the obstacle. The results show that obstacle avoidance worked at speeds lower than 3.5 m/s (12.6 km/h). The results also show that the minimum distance to the pole was about the same that it was set to be. Figure 5.15 shows that the avoidance manoeuvres were smoother at higher speeds and that the deviation from the original path was longer. This is because the dynamic restrictions were taken into account in the NMPC controller. These restrictions will lead to larger gaps at higher speeds than at lower speeds.

6.5 Navigation system architecture

The second objective of this thesis was to discuss and present the ways in which a decentralised and generic combined navigation system can be realised using the ISO 11783 network. The ISO 11783 standard contains a remote control message for commanding tractor steering by standard means. Furthermore, the standard makes it possible to obtain crucial information from the vehicle and from the GNSS system. In the approach presented here, the measurements and the actuators were distributed and the messaging went through a common bus so decentralisation according to these components can be accomplished, which was demonstrated via the test configuration. The guidance control system can only be decentralised in cases where the controllers for the tractor and for the implement are separated from one another, i.e. when the tractor is steered based on a GNSS receiver placed on top of the tractor and the implement is steered

based on another global or local measurement placed on the implement side. The NMPC algorithm presented here was used to realise a true MIMO controller for steering both the tractor and the implement in an optimal manner. For the NMPC algorithm, the information needs to be processed using a single processor because the state and model are needed from both the tractor and the implement. There are methods for using a decentralised NMPC algorithm as well (Scattolini, 2009), but none of them is applicable for combined navigation because the information flow is limited and the trailer state influences the tractor control only through the global cost function and combined model. The state estimation could be distributed, but standardising the state estimate transmission throughout an ISO 11783 network would be complicated. Therefore, the guidance system controller cannot be decentralised in a generic case.

6.6 Proposed changes to the ISO 11783 standard

Proprietary messages were used in the presented system to carry information about the working state and the steering of the implement. There are at least two different ways to use standard messages for working state information and command purposes. The first way is to use the ISO 11783-10 and implement the Task Controller server functionalities in the navigation system. Another way is to use the ISO/DIS 11783-14 and implement either the Sequence Control Master functionalities or other functionalities to trigger the sequence in the separate Sequence Control Master. However, currently the standard only allows one Task Controller to control the implement and the standard specifies that sequences should be started manually by the user. The working state problem is not discussed further in this thesis. However, the ISO 11783 standard does not include a method for managing the implement steering.

The ISO 11783 standard makes generalisations about the different tractor kinematics in such a way that the quantity used for guidance ‘steering’ is curvature (the unit is km^{-1}). This generalisation supports front-wheel steering systems, rear-wheel steering systems and tracked vehicles as well as articulated tractors. However, for guidance purposes, it is important for the guidance controller to know the offset from the functional point to the rotating point of the vehicle.

The generic framework for a combined guidance system is a tractor and a single implement connected to the rear of the tractor. To be generic and support most of the common structures used in modern machinery, up to four types of active steering systems on the implement side were considered: A) a hitch mounted with a side shift; B) a passive trailer and a drawhook side shift; C) a trailer with an articulated joint in the drawbar;

and D) a trailer with steering wheels. The four types are presented in Figure 6.2.

For standardisation purposes, a more generic quantity is needed; it should be simple enough to be generic, but at the same time give precise enough information for a combined guidance controller. The quantity of the ‘steady state side shift’, S_{ss} , was proposed in this thesis. Figure 6.2 shows how the ‘steady state side shift’ should be considered for each type of controller. The steady state side shift should correspond to the side shift of the implement’s functional point from the straight driving line that the

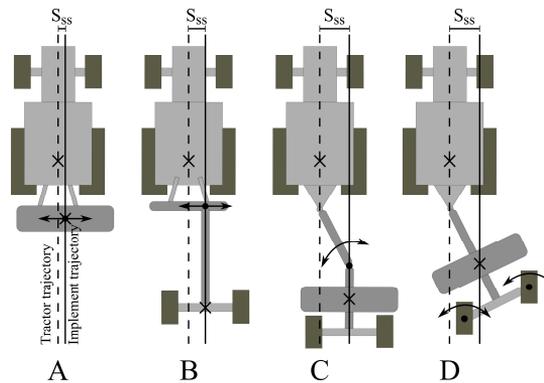


Figure 6.2. Kinematic types for implements with active steering: A) a hitch mounted with a side shift; B) a passive trailer and a drawhook side shift; C) a trailer with an articulated joint in the drawbar; D) a trailer with steering wheels.

With the proposed concept, the combined guidance controller commands are implemented by sending a setpoint for the S_{ss} , and the implement has to use its proprietary means to realise the setpoint. Practically speaking, the implement may have its own hydraulic power system, which is operated by the tractor’s PTO, or the implement may utilise ISO 11783 Tractor Class 3 remote control messages to control the hydraulic valves of the tractor, or by other means.

In the proposed framework, an implement has to transmit two measurements to the combined guidance system. The first is an estimated theoretical steady state side shift (S_{ss}) and the second is the measured angle of a free joint of the trailer. The estimated theoretical steady state side shift corresponds to the setpoint, which means that the implement does not need to know the actual side shift, i.e. a straight driving line with slippage-free conditions can be assumed. It can be computed based on the position of the hydraulic cylinder or other mechanical position measurements on the implement side.

In addition to communicating the theoretical side shift for real-time control, the message layout must also issue commands for the active/non-

active quantities and any other similar quantities that are used in the ISO 11783 tractor guidance message (ISO 11783:7, 2009). Furthermore, the NMPC controller and other similar controls require having a certain amount of knowledge about the physical dimensions of the implement and its limits. For this purpose, the ISO 11783:10 standard already defines the offset and coordinate framework that can be used. However, more details about the physical limits of the implement are needed for a guidance system, e.g. the maximum steady state side shift. The mandatory fields required by a combined guidance system need to be defined. Information about the offsets and the other coordinate systems in the implement are transmitted to a combined guidance system by using the ISO 11783:10 means, in the device description.

7. Conclusions

This thesis had two main objectives. The first objective of this thesis was to build a combined navigation system for a tractor-trailer system with implement steering, which is able to drive at a speed of at least 12 km/h with less than 10 cm lateral error. Furthermore, the first objective was defined so that the navigation system would be based on path tracking and active implement control methods. The second objective was to present the ways in which a decentralised and generic combined navigation system can be realised using the ISO 11783 network.

The combined navigation system concept was presented in the thesis. Based on the concept, both a tractor and an implement were steered in a field to lay swaths side-by-side by means of the ISO 11783 communication standard. The underlying algorithm, a Nonlinear Model Predictive Control, for handling multiple degrees of freedom was proposed. This algorithm requires a kinematic model for the vehicle and also some dynamic parameters. The algorithm also requires a method for estimating the current state of the model as well as a method for calculating the control objective, i.e. the path to be tracked.

The thesis differs from other studies that use an NMPC algorithm in that the objective was to keep both the tractor and the trailer on a path. This study is useful because, unlike other similar studies where a tractor-trailer system is controlled, the uncertainties in the environment due to slipping and sliding are considered, and additionally, the control problem is considered as a multivariate nonlinear control problem rather than as a separate or linearised problem.

The proposed method for calculating the cost function and the trajectory using the NMPC algorithm make the controller suitable for path tracking, where the desired positions and the time are not coupled. Other similar studies that use the NMPC algorithm typically use the controller for trajectory tracking, where the desired positions and the time are coupled. Furthermore, collision avoidance was augmented for the NMPC algorithm in such way that it did not increase the algorithmic complexity of the

original optimisation problem. Typical solutions for the same problem use either additional constraints or the additional costs of the obstacles.

The feasibility of the path is crucial for the calculation time of the NMPC algorithm. In this thesis, the Spiral Connection method was proposed. With the proposed method, the desired path is always feasible with respect to the constraints of the steering system. The Spiral Connection method was applied to modify the well-known shortest path principle, Dubins' Spirals, such that the curvature of the resulting path is continuous. The method can also be applied to smoothen or constrain the curvature of an arbitrary path.

The NMPC also requires an accurate state estimate in order to be stable. The global positioning system, GPS, was used together with the model of the system in an Extended Kalman Filter. The heading estimation of the GPS was improved by using an inertial measurement unit and a fibre optic gyroscope. The position estimate of the trailer was also improved by using a local relative measurement that recognised the adjacent driving lines using a 2D laser scanner. Also, the internal state of the vehicle was measured using potentiometer-type measurement devices.

The tuning of the NMPC controller and the associated EKF filter requires a great deal of expertise. With the methods presented in this thesis, the NMPC algorithm can be introduced in any vehicle with similar kinematics without needing to have expert knowledge about control engineering. The methods do not extract the absolutely correct parameters of the system, but they do give rough estimates that lead to stable and sufficient performance of the EKF filter and the NMPC controller.

The results showed that NMPC is a feasible method for realising the path tracking of the tractor-trailer system. The lateral error of the trailer was well below the required 10 cm for straight paths and within the boundaries in curved paths up to a driving speed of 12 km/h.

By using the case study, the information flows required for a combined guidance system with NMPC were illustrated. The thesis shows what kind of information can and cannot be transmitted between a tractor, an implement and a combined guidance controller using ISO 11783 standard messages. The conclusion is that the information flows inside the combined guidance controller cannot be transferred easily over the ISO 11783 network, i.e. the combined guidance controller cannot be decentralised in a general case. However, all other information, including different measurements and controls, can be transferred over the ISO 11783 network.

In conclusion, all the objectives of this thesis were met.

References

- AGCO Advanced Technology Solutions (2012). *Topcon System 150 Auto Steering*. Available online at: <http://www.agcotechologies.com/en/autoguide.htm> (accessed 25.4.12).
- Amidi, O. (1990). Integrated mobile robot control. *Tech. Report CMU-RI-TR-90-17*. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.
- Arkin, R.C., & Balch, T. (1997). AuRA: Principles and Practice in Review. *Journal of Experimental and Theoretical Artificial Intelligence*, 9, 175-189.
- Blackmore, B.S., Have, H., & Fountas, S. (2002). A proposed system architecture to enable behavioural control of an autonomous tractor. In Zhang, Q. (Ed.), *Automation Technology for Off-Road Equipment*. ASAE, USA, 12-23.
- Blackmore, B.S., Griepentrog, H.W., Nielsen, H., Nørremark, M., & Resting-Jeppersen, J. (2004). Development of a deterministic autonomous tractor. *CIGR International conference*, Beijing, China.
- Bell, T. (1999). *Precision robotic control of agricultural vehicles on realistic farm trajectory*. (Doctoral dissertation). Palo Alto: Stanford University.
- Betts, J. T. (2001). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, Philadelphia.
- Bevly, D.M. (2001). *High speed, dead reckoning, and towed implement control for automatically steered farm tractors using GPS*. (Doctoral dissertation). Palo Alto: Stanford University.
- Bochtis D.D., & Sørensen C.G. (2009). The vehicle routing problem in field logistics part I. *Biosystems Engineering*, 104 (4), 447-457.
- Bochtis D.D., & Sørensen C.G. (2010). The vehicle routing problem in field logistics: Part II. *Biosystems Engineering*, 105 (2), 180-188.
- Boost (2012). *Boost C++ libraries*. Available online at: <http://www.boost.org> (accessed 25.4.12).
- Borenstein, J. (1991). The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 7 (3), 278-288.
- Can Specification Version 2.0 (1991), Robert Bosch GmbH
- Cariou, C., Lenain, R., Thuilot, B., & Martinet, P. (2010). Autonomous maneuver of a farm vehicle with a trailed implement: motion planner and lateral-longitudinal controllers. *IEEE International Conference on Robotics and Automation*. Anchorage, Alaska, USA, 3819-3824.
- Congwei, H., Chen, W., Chen, Y., & Dajie L. (2003). Adaptive Kalman Filtering for Vehicle Navigation. *Journal of Positioning Systems*, 2 (1), 42-47.
- Darpa (2007). *Darpa Grand Challenge*. Available online at: <http://archive.darpa.mil/grandchallenge05/> (accessed 25.4.12)

- Darr, M.J., Stombaugh, T.S. & Shearer, S.A. (2005). Controller Area Network Based Distributed Control for Autonomous Vehicles. *Transactions of the ASAE*, 48 (2), 479-490.
- Deere & Company (2009a). *iGuide Operators Manual*. Available online at: http://manuals.deere.com/omview/OMPFP10808_19/ (accessed 16.9.2011)
- Deere & Company (2009b). *iSteer Operators Manual*. Available online at: http://manuals.deere.com/omview/OMZ200263_19/ (accessed 16.9.2011)
- Dickmanns, E.D. (2007). *Dynamic Vision for Perception and Control of Motion*. Springer, London.
- Dubins, L.E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79, 497-516.
- Ehrl, M., & Auernhammer, H. (2007) X-By-Wire via ISOBUS Communication Network. *Agricultural Engineering International: the CIGR Ejournal*, 9, 12.
- Encarnacao, P. & Pascoal, A. (2001). Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft. *Proceedings of the 40th IEEE Conference on Decision and Control*. 964 – 969.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., & Hrovat, D. (2007). Predictive Active Steering Control for Autonomous Vehicle System. *IEEE Transaction on Control System Technology*, 15 (3), 566-580.
- Farm Journal Inc. (2011). *Today's Auto-Steering Systems*. <http://www.agweb.com/assets/1/6/TodaysAuto-SteeringSystems_2011MG.pdf> (accessed 6.11.2012)
- Fernandes, C., Gurvits, L., & Li Z. L. (1991). A variational approach to optimal nonholonomic motion planning. *Proceedings of the IEE Int. Conf. on Robotics and Automation*. 680-685.
- Fleury, S., Souères, P., Laumond, J., & Chatila R. (1995). Primitives for Smoothing Mobile Robot Trajectories. *IEEE Trans. on Robotics and Automation*, 11 (3), 441-447.
- Fraichard, T., & Scheuer, A. 2004. From Reeds and Shepp's to Continuous-Curvature Paths. *IEEE Trans. on Robotics and Automation*, 20 (6), 1025-1035.
- Fiorini, P., & Shiller, Z. (1998). Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research*, 17 (7), 760-772.
- Franke, R., & Arnold, E. (1997). Applying new numerical algorithms to the solution of discrete-time optimal control problems. In: Warwick, K., Kárný, M. (Eds.) *Computer-Intensive Methods in Control and Signal Processing: The Curse of Dimensionality*. Birkhäuser Verlag, Basel, 105-118.
- Franke, R., & Arnold, E. (2008). *HQP: a solver for sparse nonlinear optimization*. Available online at: <http://hqp.sourceforge.net/> (accessed 15.02.11)
- Gu, D., & Hu, H. (2006). Receding Horizon Tracking Control of Wheeled Mobile Robots. *IEEE Transactions on Control Systems Technology*, 13 (4), 743-749.
- Heraud, J.A., & Lange, A.F. (2009). Agricultural Automatic Vehicle Guidance from Horses to GPS: How We Got Here, and Where We Are Going. *ASABE Distinguished Lecture Series*, 33, 1-67.
- Hauser, J. & Hindman, R. (1995). Maneuver regulation from trajectory tracking: Feedback linearizable systems. In: *Proceedings of IFAC Symp. Nonlinear Control Systems Design*. Lake Tahoe, CA, USA, 595-600.

- IEC 61162-3. (2008). Maritime navigation and radiocommunication equipment and systems -- Digital interfaces -- Part 3: Serial data instrument network, Geneva, standard
- ISO 11783-1. (2007). Tractors and machinery for agriculture and forestry -- Serial control and communications data network -- Part 1: General standard for mobile data communication, Geneva, standard
- ISO 11783-3. (2007). Tractors and machinery for agriculture and forestry -- Serial control and communications data network -- Part 3: Data link layer, Geneva, standard
- ISO 11783-7. (2009). Tractors and machinery for agriculture and forestry -- Serial control and communications data network -- Part 7: Implement messages application layer, Geneva, standard
- ISO 11783-9. (2002). Tractors and machinery for agriculture and forestry -- Serial control and communications data network -- Part 9: Tractor ECU, Geneva, standard
- ISO 11783-10. (2009). Tractors and machinery for agriculture and forestry -- Serial control and communications data network -- Part 10: Task controller and management information system data interchange, Geneva, standard
- ISO/DIS 11783-14. (2009). Tractors and machinery for agriculture and forestry -- Serial control and communications data network -- Part 14: Sequence control, Geneva, draft
- ISO 12188-1. (2010). Tractors and machinery for agriculture and forestry -- Test procedures for positioning and guidance systems in agriculture -- Part 1: Dynamic testing of satellite-based positioning devices, Geneva, standard
- Jetto, L., Longhi, S., & Venturini, G. (1999a). Development and Experimental Validation of an Adaptive Extended Kalman Filter for the Localization of Mobile Robots. *IEEE Transactions on Robotics and Automation*, 15 (2), 219-229.
- Jetto, L., Longhi, S., & Vitali, D. (1999b). Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted Kalman filter. *Control Engineering Practice*, 7 (6), 763-771.
- John Deere (2012). *GreenStar Precision Solutions*. Available online at: http://www.deere.com/wps/dcom/en_US/products/equipment/greenstar_precision_solutions/greenstar_precision_solutions.page (accessed 25.4.12)
- Jørgensen, R.N., Sørensen, C.G., Pedersen, J.M., Havn, I., Olsen, H.J., & Søgaard, H.T. (2006). Hortibot: An Accessory Kit Transforming a Slope Mower into a Robotic Tool Carrier for Hight-Tech Plant Nursing - Part I. *ASABE Annual International Meeting*, Portland, Oregon.
- Kaplan, E.D., & Hegarty, C.J. (2005). *Understanding GPS: Principles and Applications*. Artech House.
- Karkee, M., & Steward, B.L. (2010). Study of the open and closed loop characteristics of a tractor and a single axle towed implement system. *Journal of Terramechanics*, 47 (6), 379-393.
- Karkee, M., & Steward, B.L. (2011). Parameter estimation and validation of a tractor and single axle towed implement dynamic system model. *Computers and Electronics in Agriculture*, 77, 135-146
- Keicher, R., & Seufert, H. (2000). Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture*, 25, 169-194.
- Keiviczky, T., Falcone, P., Borrelli, F., Asgari, J., & Hrovat, D. (2006). Predictive Control Approach to Autonomous Vehicle Steering. In: *Proceedings the 2006 American Control Conference ACC*. Minneapolis, Minnesota, USA, 4670-4675.

- Kim, B., Neculescu, D., & Sasiadek, J. (2001). Model Predictive Control of an Autonomous Vehicle. In: *Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Como, Italy, 1279-1284.
- Klančar, G., & Škrjanc, I. (2007). Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, 55 (6), 460-469.
- Kühne, F., Gomes da Silva Jr., J.M., & Lages, W.F. (2005a). Mobile Robot Trajectory Tracking Using Model Predictive Control. *VII SBAI II LARS*. São Luís, Brazil.
- Kühne, F., Lages, W.F, & Gomes da Silva Jr., J.M. (2005b). Point Stabilization of Mobile Robots with Nonlinear Model Predictive Control. In: *Proceedings of 2005 IEEE International Conference on Mechatronics and Automation*. Niagara Falls, Canada, 3, 1163-1168.
- Lenain, R., Thuilot, B., Cariou, C., & Martinet, P. (2005). Model Predictive Control for Vehicle Guidance in Presence of Sliding: Application to Farm Vehicles Path Tracking. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. ICRA, Barcelona, Spain, 885-890.
- Lenain, R., Thuilot, B., Cariou, C., & Martinet, P. (2006). Mobile robot control in presence of sliding: Application to agricultural vehicle path tracking. *45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 6004-6009.
- Maciejowski, J.M. (2002). *Predictive control: with constraints*. Pearson Education, Harlow.
- Min, M., Ehsani, M., & Salyani, M. (2008). Dynamic accuracy of GPS receivers in citrus orchards. *Journal of the Applied Engineering in Agriculture*, 24 (6), 861-868.
- Mohamed, A.H., & Schwarz, K.P. (1999). Adaptive Kalman Filtering for INS/GPS. *Journal of Geodesy*, 73, 193-203.
- Morin, P., & Samson, C. (2008). Motion Control of Wheeled Mobile Robots. In: Siciliano, B. and Khatib, O. (Eds.), *Handbook of Robotics*, Springer, Berlin, Heidelberg, 799-826.
- Murphy, R.R. (2000). *Introduction to AI Robotics*. MIT Press, Cambridge.
- Nagasaka, Y. (2009). An Autonomous Rice Transplanter Guided by Global Positioning System and Inertial Measurement Unit. *Journal of Field Robotics*, 26 (6-7), 537-548.
- Nagy, Z.K., Mahn, B., Franke, R., & Allgöwer, F. (2007). Evaluation study of an efficient output feedback nonlinear model predictive control for temperature tracking in an industrial batch reactor. *Control Engineering Practice*, 15 (7), 839-850.
- Noguchi, N., Will, J., Reid, J., & Zhang Q. (2004). Development of a master-slave robot system for farm operations. *Computers and Electronics in Agriculture*, 44, 1-19.
- Odelson, B.J., Lutz, A., & Rawlings, J.B. (2006). The Autocovariance Least-Squares Method for Estimating Covariances: Application to Model-Based Control of Chemical Reactors. *IEEE Transactions on Control System Technology*, 14 (3), 532-540.
- Oksanen, T., Linja, M., & Visala, A. (2005). Low-cost positioning system for agricultural vehicles. In: *Proceedings of The IEEE International Symposium on Computational Intelligence in Robotics and Automation*. CIRA, Espoo, Finland, 297-302.

- Oksanen, T., & Visala, A. (2004). Optimal control of tractor-trailer system in headlands. *Proceedings of ASAE International Conference on Automation Technology for Off-road Equipment*, Kyoto, Japan, 255-263.
- Oksanen, T. (2007). *Path Planning Algorithms for Agricultural Field Machines*. (Doctoral dissertation). Helsinki University of Technology.
- Oksanen, T., & Visala, A. (2009). Coverage Path Planning Algorithms for Agricultural Field Machines. *Journal of field robotics*, 26 (8), 651-668.
- Omead, A. (1990). *Integrated Mobile Robot Control*. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- Parlangeli, D., & Idiveri, G. (2010). Dubins inspired 2D smooth paths with bounded curvature and curvature derivative. *Proceedings of the 7th IFAC Symposium on Intelligent Autonomous Vehicles*. 216-221.
- Pirti, A., (2008). Accuracy Analysis of GPS Positioning Near the Forest Environment. *Croatian Journal of Forest Engineering*, 29 (2), 189-199.
- Rankin, J. (1994). GPS and differential GPS: An error model for sensor simulation. *IEEE Position Location and Nav. Symp.* 260-266.
- Reeds, J.A., & Shepp, L.A. (1990). Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145 (2), 367-393.
- Reid, J.F., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture*, 25, 155-167.
- Rintanen, K., Mäkelä, H., Koskinen, K., Puputti, J., Sampo, M., & Ojala, M. (1996). Development of an autonomous navigation system for an outdoor vehicle. *Control Engineering Practice*. 4 (4), 499-505.
- Ruckelshausen, A., Klose, R., Linz, A., Marquering, J., Thiel, M., & Tölke, S. (2006). Autonomous robots for weed control [Autonome Roboter zur Unkrautbekämpfung]. *Journal of Plant Diseases and Protection*, 20, 173-180.
- SAE J1939 (1994). Recommended Practice for a Serial Communication Network (Class C), Society of Automotive Engineers, standard
- Sasiadek, J.Z., Wang, Q., & Zeremba, M.B. (2000). Fuzzy Adaptive Kalman Filtering for INS/GPS Data Fusion. In *Proceedings of the 15th IEEE International Symposium on Intelligent Control*. ISIC, Rio, Patras, Greece. 181-186
- Scattolini, R. (2009). Architectures for distributed and hierarchical Model Predictive Control – A review. *Journal of Process Control*, 19, 723-731.
- Schittkowski, K. (1983). On the convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function. *A Journal of Mathematical Programming and Operations Research*, 14 (2), 197-216.
- Siew, K.W., Katupitiya, J., Eaton, R., & Pota, H. (2009). Simulation of an articulated tractor-implement-trailer model under the influence of lateral disturbances. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Singapore. 951-956
- Snider, J.M. (2009). Automatic Steering Methods for Autonomous Automobile Path Tracking. *Tech. Report CMU-RI-TR-09-08*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA USA
- Stombaugh, T., Shearer, S., Fulton, J., & Ehsani, M. (2002). Elements of a dynamic GPS test standard. *ASAE Paper No. 021150*. St. Joseph, Mich.: ASAE.
- Strentz, A., Dima, C., Wellington, C., Herman, H., & Stager, D. (2002). A System for Semi-Autonomous Tractor Operations. *Autonomous Robots*, 13 (1), 87-103.

- Tindell, K., Burnds, A., & Wellings, A.J. (1995). Calculating controller area network (CAN) message response times. *Control Eng. Practice*, 2 (8), 1163-1169.
- Tilove, R. (1990). Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials. In: *1990 IEEE International Conference on Robotics and Automation*, ICRA, Cincinnati, OH USA. 566-571.
- Torii, T. (2000). Research in autonomous agriculture vehicles in Japan. *Computers and Electronics in Agriculture*, 25, 133-153.
- Tu, X. (2013). *Robust navigation control and headland turning optimization of agricultural vehicles*. (Doctoral dissertation). Iowa State University.
- Vougioukas, S., Blackmore, S., Nielsen, J. & Fountas, S. (2006). A Two-Stage Optimal Motion Planner for Autonomous Agricultural Vehicles. *Precision Agriculture*, 7 (5), 361-377.
- Vougioukas, S.G. (2007). Reactive Trajectory Tracking for Mobile Robots based on Non Linear Model Predictive Control. *2007 IEEE International Conference on Robotics and Automation*. ICRA, Roma, Italy. 3074.
- Vougioukas, S.G. (2009). A framework for motion coordination of small teams of agricultural robots. In: van Henten, E.J., Goense, D., Lokhorst, C. (Eds.), *Precision agriculture '09. Proceedings of the Joint International Agricultural Conference*. Wageningen Academic Publishers, Wageningen, Netherlands. 585-593.
- Vougioukas, S.G. (2012). A distributed control framework for motion coordination of teams of autonomous agricultural vehicles. *Biosystems Engineering*, 113, 284-297.
- Weidong, D., Wang, J., Rizos, C., & Kinlyside, D. (2007). Improving Adaptive Kalman Estimation in GPS/INS Integration. *The Journal of Navigation*, 60, 517-529.
- Werner, R., Kormann, G., & Mueller, S. (2012). Path tracking control of tractors and steerable implements based on kinematic and dynamic modeling. In *Proc. 11th International Conference on Precision Agriculture*. Indianapolis, Indiana USA, July 15-18.
- Werner, R., Kormann, G., & Mueller, S. (2013). System Model Based Path Tracking Control of Actively Steered Implements in Simulation and Experiment. In *Proc. Agricontrol 2013, IFAC International Conference*. Helsinki, Finland, August 28-29.
- Wit, J. Crane III, C.D., & Armstrong, D. (2004). Autonomous Ground Vehicle Path Tracking. *Journal of Robotics Systems*, 21 (8), 439-449
- Wojznsnis, W., Gudaz, J., Blevins, T., & Mehta, A. (2003). Practical approach to tuning MPC. *ISA Transactions*, 42 (1), 149-162.
- Wong, J.Y. (2008). *Theory of Ground Vehicles 4th ed.* John Wiley and Sons.
- Wu, C., Ayers, P.D., & Anderson, A.B. (2006). Influence of Travel Direction on GPS Accuracy for Vehicle Tracking. *Transactions of the ASABE*, 49 (3), 623-634
- Yang, K., & Sukkarieh, S. (2010). An Analytical Continuous-Curvature Path-Smoothing Algorithm. *IEEE Trans. on Robotics*, 26 (3), 561-568.

Appendix: Publications



ISBN 978-952-60-5392-9
ISBN 978-952-60-5391-2 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

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

**BUSINESS +
ECONOMY**

**ART +
DESIGN +
ARCHITECTURE**

**SCIENCE +
TECHNOLOGY**

CROSSOVER

**DOCTORAL
DISSERTATIONS**