Biosystems Engineering 263 (2026) 104375

Contents lists available at ScienceDirect

Biosystems Engineering

journal homepage: www.elsevier.com/locate/www.elsevier.com/locate/issn/15375110

Research Paper

Nonlinear model predictive control for autonomous driving in terrain ☆

Jere Knuutinen [a,b,*], Tabish Badar [a], Juha Backman [a,b], Arto Visala [a]

[a] Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland
[b] Farming technologies, Natural Resources Institute Finland (Luke), Lönnrotinkatu 7, 50100 Mikkeli, Finland

ARTICLE INFO

Keywords:
Path tracking
Dynamics
Kinematics
3D elevation models
Forest machines

ABSTRACT

Forestry machines used nowadays function in a wide variety of terrains. In order to make these vehicles operate autonomously advanced control methods are needed. However, current research related to autonomous driving generally assumes flat terrain in the control as well as in the estimation. Therefore concerning this matter, this paper addresses potential solutions by investigating and applying nonlinear model predictive control (NMPC) for autonomous driving in uneven terrain. This paper proposes a hybrid model derived from the dynamic six-degrees-of-freedom (6-DOF) model for motion control purposes. The developed NMPC method utilises a path tracking approach and aims to minimise the time-independent tracking error between the position of the vehicle and path by utilising the proposed hybrid model and three-dimensional (3D) terrain map. The effectiveness of the predictive controller is tested using three different test paths and terrains. An all-terrain electric vehicle (ATV) called Polaris is utilised to test and confirm the functionality of the control method. In addition, the paper proposes a rollover avoidance method and tests it in simulation environment. The method aims to lower the vehicle speed in the presence of high roll angles. The results from the actual tests with the implementation of the NMPC method indicate that accurate path-tracking results can be obtained with the proposed controller in the test paths used in this study with the tracking errors being 0.11 m, 0.07 m and 0.1 m.

1. Introduction

At the moment used machines in the forest industry operate in very variable terrains and normally the harvesting operation includes two human-operated machines, specifically a harvester and a forwarder. These machines are illustrated in Fig. 1. The trees are felled and the branches are removed by the harvester. Moreover, in the forest, it also cuts stems into logs. The tasks of the forwarder include loading and delivering the logs to the roadside from the forest. From the roadside, trucks take them to the wood processing plants. Due to the own mass and the heavy load of the forwarder, it may cause harm to the forest terrain. Therefore, this research aims to investigate a novel type of forest machine concept in which two lighter semi-autonomous forwarders are utilised alongside the traditionally operated harvester. This allows removing the cabin and crane from the forwarder. From this type of design, the machine becomes lighter and more sustainable for the forest ground. In this situation, the machine that goes first, namely the harvester, drives a desired path, and thus presents the driving paths for the forwarder. In addition to this, the harvester's tasks include loading the stems onto the forwarder. The forwarder can follow this predefined path using an optimal control method called nonlinear model predictive control (NMPC), which aims to optimise the control inputs by utilising a nonlinear system model of the vehicle and by minimising an objective function. A detailed description of this new forest machine chain can be found from Badar, Backman and Visala (2024) and Badar, Ouattara, et al. (2024).

A potential approach for autonomous driving in uneven terrains is the NMPC because of its competence to perform under constraints in both controls and states but at the same time, it can optimise the controls by predicting the responses of the nonlinear system in future time steps. The model that is often utilised and possesses the required components to examine the dynamic behaviour of the vehicle is a six-degree-of-freedom (6-DOF) model (Badar, Backman & Visala, 2024; Etkin & Reid, 1995; Schofield, 2006; Shim & Ghike, 2007). The computation times of the NMPC methods grow in conjunction with the count of system controls and states, as well as with the complexity of the underlying nonlinear system. In order to solve this issue in the case of the NMPC method, a potential solution is the utilisation of a reduced-order model. The main purpose of such a model is to decrease the quantity of the state variables, to reduce the overall system complexity, and hence overall computational resources required. Previous papers

☆ The research is funded by the Technology Industries of Finland Centennial Foundation and Jane and Aatos Erkko Foundation, Finland.
* Corresponding author at: Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland.
E-mail address: jere.knuutinen@aalto.fi (J. Knuutinen).

https://doi.org/10.1016/j.biosystemseng.2025.104375
Received 14 February 2025; Received in revised form 19 December 2025; Accepted 22 December 2025
Available online 19 January 2026

Nomenclature

Abbreviations

3D	Three dimensional
ATV	All-terrain electric vehicle
CDEKF	Continuous discrete extended Kalman filter
CG	Centre of gravity
CRA	Centre of the rear axle
DOF	Degree-of-freedom
ECU	Electronic control unit
FF-PID	Feed-forward proportional–integral–derivative
FL	Front left
FR	Front right
GNSS	Global Navigation Satellite System
IMU	Inertial measurement unit
LTR	Load transfer ratio
MPC	Model predictive control
NMPC	Nonlinear model predictive control
RL	Rear left
RR	Rear right
RTK	Real-time kinematic
SPAN	Synchronous position, attitude, and navigation
VC	Vehicle centre

Subscripts

b	body frame quantity
k	corner of the vehicle
m	measured quantity
t	tyre-ground contact patch frame quantity

Greek Symbols

δ	steering angle (rad)
ϕ	roll angle (rad)
ψ	yaw angle (rad)
θ	pitch angle (rad)

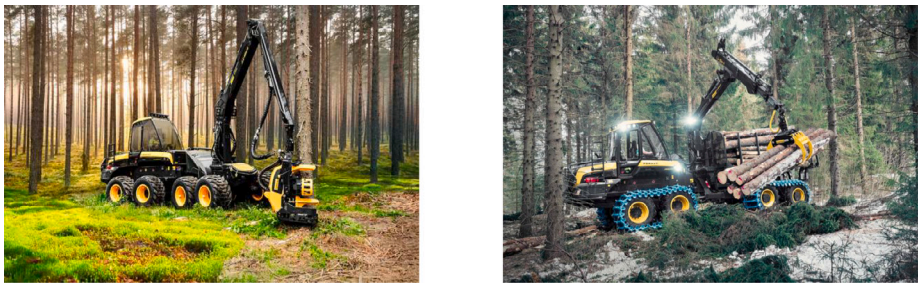
Symbols

$a_c, \mathrm{d}K_c$	forward acceleration command ($\mathrm{m\,s^{-2}}$) and rate of change of curvature command ($\mathrm{m^{-1}\,s^{-1}}$)
B_k	coefficient of stiffness of the kth corner ($\mathrm{N\,m^{-1}}$)
C_k	coefficient of damping of the kth corner ($\mathrm{N\,s\,m^{-1}}$)
F_x, F_y	total longitudinal force and total lateral force (N)
$F_{x,b}, F_{y,b}, F_{z,b}$	longitudinal force, lateral force, and normal force in body frame (N)
$F_{x,t}, F_{y,t}, F_{z,t}$	longitudinal force, lateral force, and normal force at tyre-ground contact patch frame (N)
$\mathcal{H}_k, \dot{\mathcal{H}}_k$	ground height (m), and rate of the change of ground height ($\mathrm{m\,s^{-1}}$)
I_{xx}, I_{yy}	moments of inertia around the body frame x and y-axes ($\mathrm{kg\,m^2}$)
J	objective function
K, K_c	curvature and curvature command ($\mathrm{m^{-1}}$)
L_k, M_k	rolling moment and pitching moment of the kth corner of the vehicle (Nm)
m	mass of the vehicle (kg)
N	prediction horizon
p, q, r	roll rate, pitch rate, and yaw rate about CG in body frame ($\mathrm{rad\,s^{-1}}$)
Q, R	weighting matrices of the states and controls
$\mathcal{U}_{\min}, \mathcal{U}_{\max}$	minimum and maximum control constraints
t, l, h	Width, length and height of the vehicle (m)
u, u_c, v, w	forward velocity, forward velocity command, lateral velocity, and vertical velocity of VC in body frame ($\mathrm{ms^{-1}}$)
X, Y, Z	inertial positions of the VC in global frame (m)
$\mathcal{X}_{\min}, \mathcal{X}_{\max}$	minimum and maximum state constraints

dealing with NMPC methods have also compared the effect of model complexity. For example, Chen et al. (2020) used the following three models in model predictive controller (MPC) module: a simple bicycle, an 8-DOF, and a 14-DOF. The system plant was the 14-DOF model in their study. The study stated that no supplemental advantage was gained when more sophisticated models than a bicycle model were used for path tracking in MPC methods. Ye et al. (2023) reported that the kinematic model provides adequate path-tracking performance when the applied speed references were kept under 15 $\mathrm{m\,s^{-1}}$. The kinematic model has also been used for path following, for example by Wang et al. (2024) and Yin et al. (2020). Moreover, in Chen and Chen (2020), the 8-DOF model was used in the MPC module, whereas the system plant was the 14-DOF model. The paper concluded that the proposed method was capable of giving good results for path following. In turn, in a study by Sun et al. (2024) a 3-DOF model was utilised together with a nonlinear tyre model. The paper also proposed a fuzzy controller that optimised the parameters of the MPC. In a study by Liu et al. (2016), the authors utilised four different 2-DOF models with different tyre models for obstacle avoidance as well as the 14-DOF model. The study concluded that the 2-DOF model with linear tyre model was capable of operating in the same way as the 14-DOF model when low speeds were utilised.

In Lee et al. (2013), the authors studied an MPC-based rollover avoidance method using the load transfer ratio (LTR). Different types of sophisticated controllers have also been proposed by classical control theory in the path tracking framework, as in Hu et al. (2016), or in rollover avoidance (Imine et al., 2012). In this paper, rollover avoidance is also discussed. However, particularly in the context of NMPC, the roll angle prediction is monitored, and the NMPC speed reference is calculated on the basis of that. In previous studies, the main limitation has been that the ground profile is expected to be flat, meaning that the models do not take into account the shape of the terrain. Therefore, these models cannot account for the pitching or rolling dynamics caused by uneven terrain (Badar, Backman & Visala, 2024). In contrast to the aforementioned studies, this paper focusses on autonomous driving in uneven terrain and takes into account the terrain in the proposed hybrid model. This paper also differs from other papers that consider uneven ground. For example, when comparing with Yu et al. (2021), this paper includes the terrain in the dynamics of the prediction model, whereas they use the terrain geometry. Similarly, the method proposed in Tan et al. (2022) is at its core an MPC, not an NMPC.

In this paper, the main objective is to determine and formulate a suitable real-time model that can be used for autonomous driving



(a) Ponsse's Cobra Harvester (b) Ponsse's Bison Forwarder

Fig. 1. In (a) a harvester is shown. In (b) a forwarder is shown. Both of these machines are part of the current forestry industry. *Source:* Ponsse.

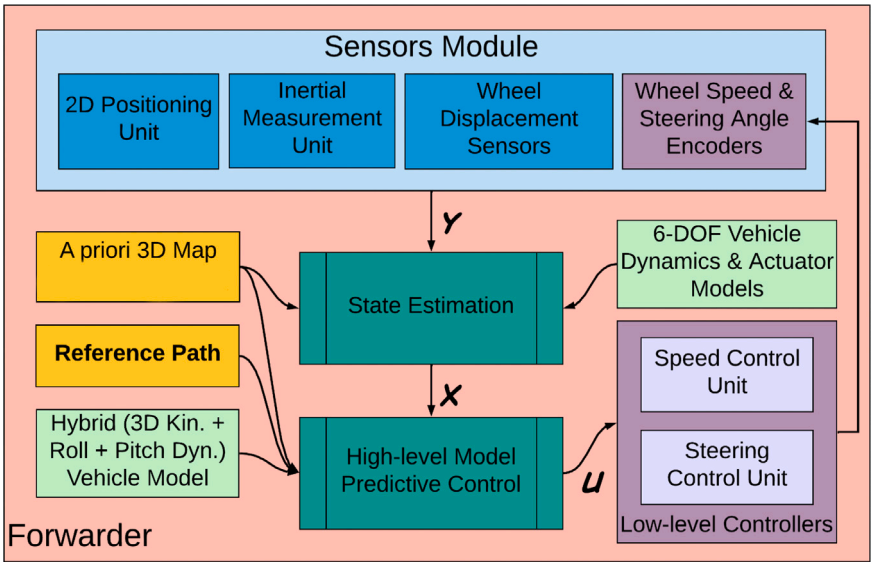


Fig. 2. Function block diagram of the forwarder unit (Tabish Badar, 2024).

Fig. 3. Research platform used (Badar, Backman & Visala, 2024).

in uneven terrain. Designing such a model is a two-fold problem, since it should be at the same time reasonably exact to describe the fundamental dynamic responses of the vehicle, but also sufficiently simple to be manageable for real-time implementation and operation. The main contribution of this paper when compared with previous studies is that the ground is not assumed to be flat in the control method. Instead, the height odometry methods published in Badar et al. (2023) and in Badar, Ouattara, et al. (2024) are used to get spatial positions for each wheel and the corresponding height values of the ground. This is accomplished by utilising measurements of the wheel heights as well as the attitude of the vehicle. After preprocessing, the wheel paths are used as a 3D map in the hybrid model-based NMPC and continuous discrete extended Kalman filter (CDEKF). Furthermore, the main goal of this paper is that with the combination of the hybrid model and the 3D terrain model, the NMPC method is not only capable of achieving relatively accurate path tracking results, but also able to capture the necessary dynamic behaviour of the vehicle when driving in terrain. For example, roll angle predictions should be sufficiently accurate so that they can be utilised to avoid rollover. More precisely, the speed reference of the NMPC module can be reduced or set to zero when it is suspected that the roll angle is close to the limit.

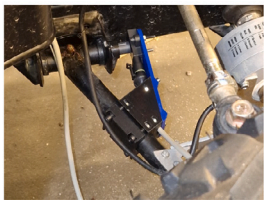
2. Materials and methods

This section first presents information on the research platform used. The section continues by presenting the test paths and the 3D maps that are used in the NMPC method and CDEKF. Furthermore, the

(a) EPEC 5050 Control Unit

(b) SPAN-IGM-S1 Unit

(c) Pinwheel's GNSS Antenna



(d) Honeywell's Hall-effect Sensor

Fig. 4. Sensors and other electronics equipment installed in Polaris (Badar, Ouattara, et al., 2024).

section also shows the actual NMPC method implemented. Finally, the state estimation method and simulation environment are described.

Fig. 2 shows the function block diagram that illustrates the overall architecture of forwarder control system. These modules are presented more in detail in the following sections.

2.1. Research platform

Fig. 3 presents the Polaris e-ATV, which is used as a research platform in this study. Polaris is equipped with different sensors enabling autonomous driving. The most relevant electronic devices and sensors to support this and in particular for this paper are two electronic control units (ECUs) from EPEC company, a synchronous position, attitude, and navigation (SPAN) unit, Pinwheel's global navigation satellite system (GNSS) antenna, and four Hall-effect sensors. These devices are shown in Fig. 4.

The ECUs that are present in Polaris are responsible for the low-level velocity and steering control in automatic mode. The ECUs are used to receive control commands from the main computer, take care of low-level communication with the sensors in the Polaris, as well as send the necessary sensor data via the J1939 bus to the main computer. The used ECU model is illustrated in Fig. 4(a). Furthermore, the SPAN unit is used to obtain position information of the vehicle. It is used together with the Pinwheel GNSS antenna, real-time kinematics (RTK) corrections via 4G link, and the inertial measurement unit (IMU). The SPAN unit and Pinwheel's GNSS antenna are shown in Figs. 4(b) and (c), respectively. In turn, the wheel heights are measured using the Hall-effect sensors. These sensors are mounted in each shaft of the four wheels as shown in Fig. 4(d) (Badar, Ouattara, et al., 2024) (Badar, Backman & Visala, 2024).

2.2. Test paths and 3D maps

In this study, three different paths are studied. The first path is located in Otaniemi, Espoo (Finland). In addition, uneven concrete and forest paths located in Vakola, Vihti (Finland) are used to examine the performance of the control system. Throughout the rest of the paper, these paths are called Otaniemi path, Vakola concrete path, and Forest path.

Fig. 7 shows a satellite image of the area, from which the Vakola concrete and the Forest path can be seen. The Otaniemi path is a relatively flat asphalt path that is located in a parking lot. In turn, the Vakola concrete path is a circular-shaped path with considerable height differences and fairly large variations in slope for the Polaris-type vehicle. The used Forest path, on the other hand, is located near the Vakola concrete path and is used to demonstrate autonomous driving in the forest environment.

In the case of the Otaniemi path, the available point cloud data from the National Land Survey of Finland is utilised (National Land Survey of Finland, 2024). Moreover, Fig. 5 shows the interpolated map points in the Otaniemi path and the path. Fig. 6, on the other hand, shows part of the original point cloud, which has been modified by adding unevenness to the path. It is used to test the proposed vehicle speed reduction method in the presence of high roll angles in a simulation environment.

In the case of Vakola concrete and Forest paths, 3D maps of the test paths for both the NMPC and the EKF module are obtained using the methods introduced in Badar et al. (2023) and Badar, Ouattara, et al. (2024). Badar, Ouattara, et al. (2024) also includes a discussion of transmitting the generated height map from the harvester to the forwarder. In the forest path, UAV-based map data is used together with the methods to obtain the height of the path. The methods presented output the 3D path for each wheel. The methods require that the path is driven beforehand and that RTK-GNSS, IMU, and wheel height measurements are recorded at 20 Hz. This also demonstrates the semi-autonomous harvester forwarder concept where the harvester goes first and the forwarder follows the path. The wheel paths from the right and left rear wheels are used to create the map. In the forest path, moving average filtering is used to reduce spikiness. The used wheel paths are visible in Fig. 8. In the case of the Forest path, a suitable shorter path is selected from the entire path shown in Fig. 7. This is done because the experiments showed that the low-level speed controller as well as the GNSS receiver worked in that particular path in an acceptable manner. Furthermore, the 3D wheel paths obtained are first resampled using scattered interpolation (SCATTERINTERPOLANT) with the nearest method available in MATLAB. This is done to obtain a rectangular point grid and the corresponding height values with a step size of 0.3 m in both the x and y directions on the Vakola concrete path and 0.5 m in both the x and y directions on the forest path. After that, these sampled grid value pairs are utilised in the B-spline based multivariate interpolate method presented in Lee et al. (1997) and used in C++ implementations of the NMPC module and CDEKF. The key characteristics of the paths are summarised in Table 1.

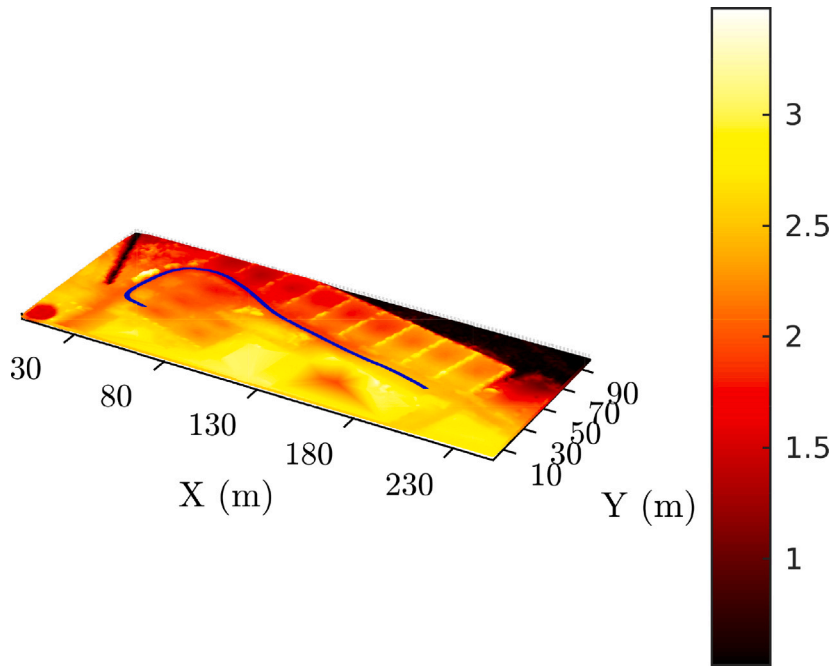


Fig. 5. Open-source point cloud provided by the National Land Survey of Finland and the path from Otaniemi. The colour indicates the height of the terrain. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

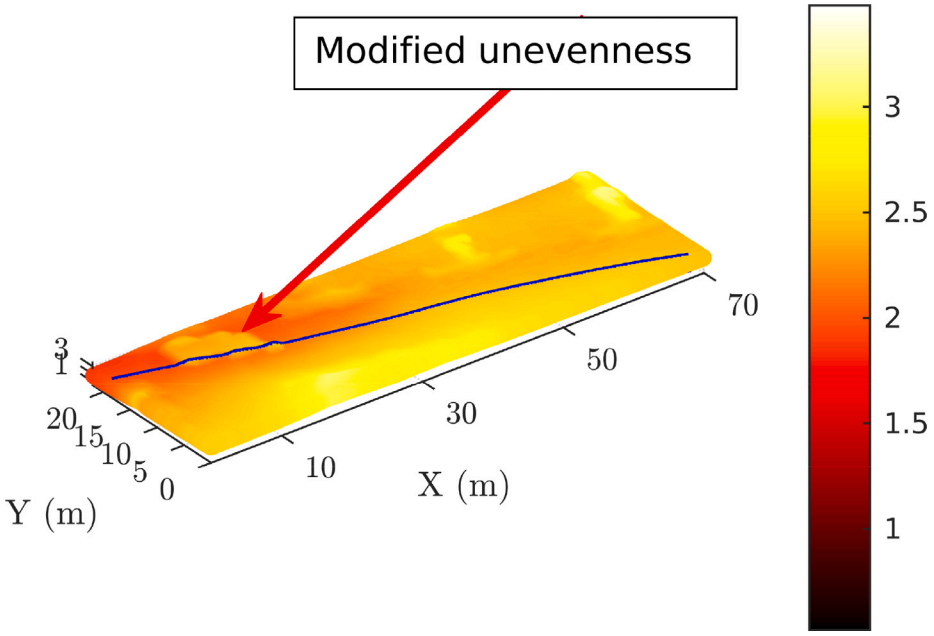


Fig. 6. Modified point cloud and the path used to test the rollover avoidance method. The colour indicates the height of the terrain. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Terrain characteristics.

Terrain characteristics/path	Vakola	Forest	Otaniemi	Otaniemi modified
Terrain material	Concrete	Gravel	Asphalt	Asphalt
Max height (m)	39.77	70.44	2.53	2.53
Min height (m)	39.30	69.12	1.57	1.57
Max slope (deg)	22.9	5.2	2.0	22.4
Min slope (deg)	−13.8	−5.0	−1.9	−15.53

2.3. Vehicle modelling for motion control

The proposed hybrid model for motion control is derived from the dynamic 6-DOF model presented in Badar, Backman and Visala (2024). The reason to adopt a hybrid model instead of a 6-DOF dynamical model is that the NMPC is computationally infeasible at 20 Hz for the 6-DOF model. As described in Badar, Knuutinen et al. (2024), the 6-DOF model fits the high-fidelity simulations, whereas the hybrid model is adapted for real-time NMPC implementation. Fig. 9 shows a 3D schematic of the 6-DOF vehicle. The figure also shows the different coordinate systems and the sign convention used in the modelling. This paper adapts the 6-DOF conventions presented there.

Fig. 7. Locations of the Vakola concrete path and the Forest path (Badar, Backman & Visala, 2024).

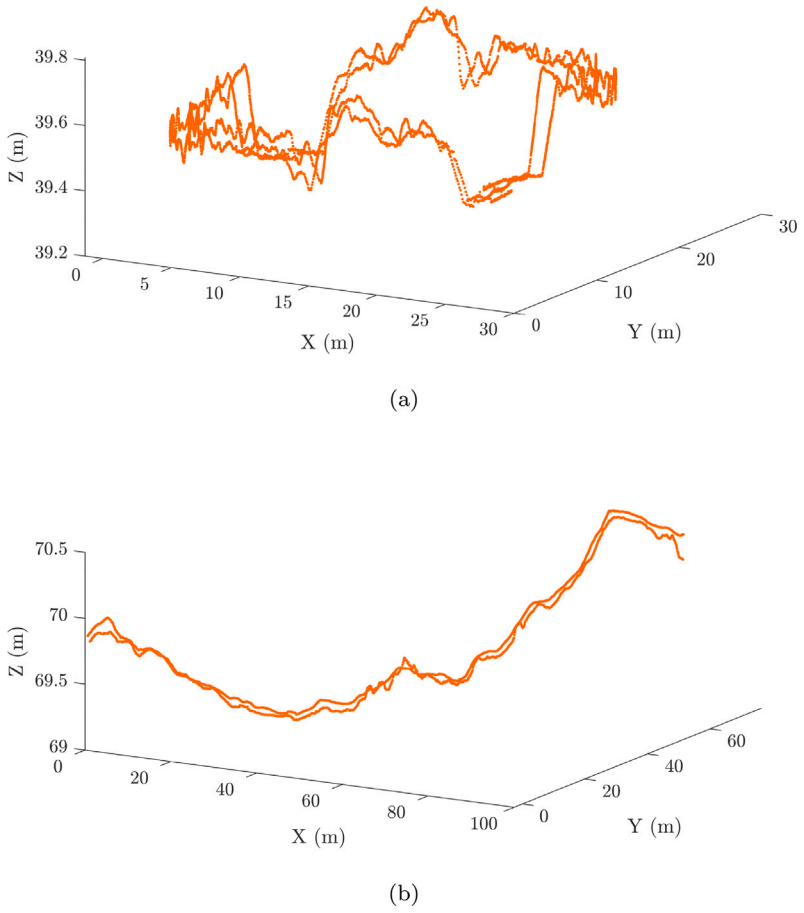


Fig. 8. (a) Rear left and rear right 3D wheel paths in the Vakola concrete path. (b) Rear left and rear right 3D wheel paths in the Forest path.

The proposed hybrid model leverages the rigid body dynamics assumption, and it combines kinematic and dynamic parts. The kinematic part is used to model the position and heading of the vehicle in the inertial frame. Moreover, it is presumed that the centre of gravity (CG) and the centre of the vehicle (VC) align. This simply means that it is assumed that the vehicle has, during its operation period, symmetric dimensions and an unvarying homogeneous mass distribution. As shown in Fig. 9, the body frame is tied to the VC. For the dynamic part of the model, the 2-DOF dynamic element is utilised. It is used to describe in the body frame the pitching and rolling dynamics.

The velocities in the inertial frame are determined by utilising the following

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \mathbf{T}_1 \begin{bmatrix} u \\ 0 \\ 0 \end{bmatrix}, \tag{1}$$

where $\dot{X}$, $\dot{Y}$, $\dot{Z}$ correspond to the inertial velocities of the VC and u is the forward velocity in the body frame. Moreover, $\mathbf{T_1}$ is the matrix from the body frame to the inertial frame. It can be defined as in Etkin

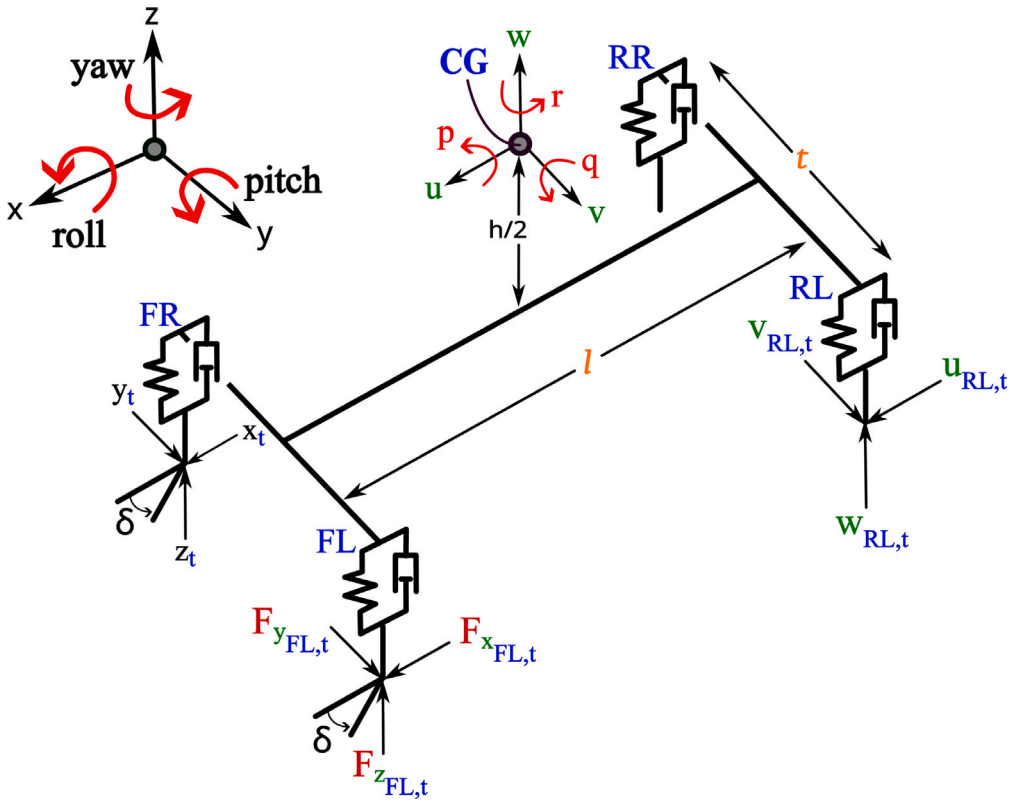


Fig. 9. 3D schematic of the 6-DOF model presented in Badar, Backman and Visala (2024). The same coordinate systems and the sign convention are used also in this paper.

and Reid (1995).

$$\mathbf{T}_1 = \underbrace{\begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}_z(\psi)} \underbrace{\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}}_{\mathbf{R}_y(\theta)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}}_{\mathbf{R}_x(\phi)}, \tag{2}$$

where ϕ, θ, ψ are the roll, pitch, and heading angles, respectively. From Eq. (1) it can be seen that it presents a 3D kinematic model and takes into account the forward velocity of the vehicle. It also incorporates the terrain changes, but it neglects the lateral v and vertical w velocities of the VC, which are present in the 6-DOF model and can also be seen in Fig. 9.

In the case of the dynamic 6-DOF model, the angular velocities in the body frame can be utilised to obtain the rate of change of the roll, pitch, and yaw angles using:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{3}$$

where p is roll rate, q is pitch rate, and r is yaw rate (Etkin & Reid, 1995).

However, in the context of the hybrid model, the yaw rate in the body frame is calculated by utilising

$$r = uK = u\frac{\tan\delta_k}{l}, \tag{4}$$

where K is curvature, δ_k is steering angle and l is wheelbase.

It should be noted that defining the curvature in this way applies to the centre of the rear axle (CRA) of the vehicle. Eq. (4) still maintains its validity also for the centre of the vehicle, since the hybrid model assumes rigid body dynamics and zero slipping conditions. Therefore, when we apply Eq. (4) in Eq. (3), the following updated equations are derived for the hybrid model:

$$\dot{\psi} = \left(q\sin\phi + uK\cos\phi\right)/\cos\theta, \tag{5}$$

$$\dot{\theta} = q\cos\phi - uK\sin\phi, \tag{6}$$

$$\dot{\phi} = p + \dot{\psi}\sin\theta. \tag{7}$$

Note that, when both the roll and pitch angles are zero, then Eq. (5) yields to $\dot{\psi} = uK$. That is commonly utilised in 2D plane based kinematic modelling.

The rate of change of angular velocities in the body frame is determined using the following

$$\dot{p} = \sum_k L_k / I_{xx}, \tag{8}$$

$$\dot{q} = \sum_k M_k / I_{yy}, \tag{9}$$

where I_{xx} and I_{yy} are the terms related to moments of inertia around the body frame x and y-axes, respectively. The subscript k corresponds to the kth corner of the vehicle. Moreover, the rolling (L_k) and pitching (M_k) moments at each corner can be written using the following:

$$\begin{bmatrix} L_k \\ M_k \\ - \end{bmatrix} = \left(\mathbf{p}_k + \left(\mathbf{R}_z(-\psi)\mathbf{T}_1\right)^{-1} \begin{bmatrix} 0 \\ 0 \\ -\Delta_k \end{bmatrix}\right) \times \begin{bmatrix} F_{x_{k,b}} \\ F_{y_{k,b}} \\ F_{z_{k,b}}. \end{bmatrix}. \tag{10}$$

In Eq. (10) the forces ($F_{x_{k,b}}, F_{y_{k,b}}, F_{z_{k,b}}$) applied are in the body frame, and the subscript b is utilised to indicate that. In turn, $\mathbf{p}_k$ is used to describe the position of the kth wheel centre from the VC. In turn, $\times$ is a cross-product operator. Eq. (10) also takes into account the deflection of the kth suspension system. This is achieved by using Δ_k, as over the period of compression or elongation, it has an impact on the moment arm length. Furthermore, Δ_k is transformed to the body frame as in Shim and Ghike (2007).

2.3.1. Modelling tyre forces for hybrid model

In the case of the hybrid vehicle model, the individual lateral or longitudinal tyre forces are not taken into account. Alternatively, it is assumed that the total longitudinal and centripetal force experienced by CG is known, thus simplifying the force calculation. This is because tyre specific parameters such as cornering stiffness, side-slip angles, and longitudinal slip parameter, are difficult to estimate. Moreover, the distribution of the resulting total forces to each kth corner of the vehicle is carried out according to its mass m_k. The total longitudinal force in the body frame is calculated by neglecting the longitudinal and lateral slips of the tyres as follows:

$$F_x = m\dot{u}. \tag{11}$$

In turn, the total lateral force in the body frame is calculated as follows

$$F_y = mu^2 K, \tag{12}$$

where m is the mass of the vehicle. Hence, the longitudinal, lateral, and normal forces at each tyre-ground contact patch of the vehicle are calculated as written below:

$$F_{x_{k,t}} = \mu_x F_{z_{k,t}} \tag{13}$$

$$F_{y_{k,t}} = \mu_y F_{z_{k,t}} \tag{14}$$

$$F_{z_{k,t}} = -B_k(Z_k - \mathcal{H}_k) - C_k(\dot{Z}_k - \dot{\mathcal{H}}_k), \tag{15}$$

where

$$\mu_x = F_x \Big/ \left(\sum_k F_{z_{k,t}}\right); \quad \mu_y = F_y \Big/ \left(\sum_k F_{z_{k,t}}\right).$$

In Eq. (15) B_k and C_k are the coefficients of stiffness and damping of the suspension system, respectively. Z_k and $\dot{Z}_k$ are the height and inertial velocity of the kth wheel, respectively. The height of the ground under the kth wheel is denoted by H_k. H_k is obtained from the 3D map using the multivariate interpolation method mentioned in Section 2.2. Moreover, $\dot{H}_k$ is obtained by calculating the rate of change in ground height under the kth wheel. The calculation of H_k and $\dot{H}_k$ is as follows:

$$\mathcal{H}_k = \mathbf{map}\left(X_k, Y_k\right), \quad \dot{\mathcal{H}} = \frac{\partial \mathcal{H}_k}{\partial X_k}\dot{X} + \frac{\partial \mathcal{H}_k}{\partial Y_k}\dot{Y}, \tag{16}$$

where **map** is the interpolation method. As Eq. (10) requires that the forces are in the body frame, these must be transformed from the tyre-ground contact patch frame by utilising the following

$$\begin{bmatrix} F_{x_{k,b}} \\ F_{y_{k,b}} \\ F_{z_{k,b}} \end{bmatrix} = \mathbf{R}_x(-\phi)\mathbf{R}_y(-\theta) \begin{bmatrix} F_{x_{k,t}} \\ F_{y_{k,t}} \\ F_{z_{k,t}} \end{bmatrix}. \tag{17}$$

2.3.2. Actuator dynamics for hybrid model

The kinematic model often takes velocity and curvature as input (Backman et al., 2012). Hence, to model the actuator dynamics of the platform a first order low pass filter type speed and curvature dynamics are utilised. These are included in the hybrid model as follows:

$$\dot{u} = c_\alpha u + c_\beta u_c \tag{18}$$

$$\dot{K} = c_\kappa K + c_\lambda K_c. \tag{19}$$

In Eqs. (18) and (19), the terms $c_{(\cdot)}$ imply time constants. In turn, u_c and K_c are the actual commands. As a result, the state vector for the hybrid vehicle model is:

$$\mathcal{X}(\tau) = (X, Y, Z, \phi, \theta, \psi, u, p, q, K). \tag{20}$$

2.4. Nonlinear model predictive control

In general, the NMPC calculates the optimal control trajectory by minimising the objective function by utilising the current state information and the nonlinear dynamic model of the system, which is used to predict the system responses. Once the optimisation is completed, the first controls calculated are applied to the system. Thereafter, the optimisation is carried out again with the new state information obtained. In this paper, the NMPC module uses the states estimates from the CDEKF module, the hybrid model shown in Section 2.3 for the prediction, and the 3D map information presented in Section 2.2 to account for vehicle dynamics on uneven ground. The NMPC problem can be written as:

$$\begin{aligned} \text{minimise} \quad & J(\mathcal{X}(t_k), \mathcal{U}(t_k)), \\ \text{subject to:} \quad & \mathcal{X}(t_{k+1}) = F(\mathcal{X}(t_k), \mathcal{U}(t_k)) \\ & \mathcal{X}_{\min} \le \mathcal{X}(t_k) \le \mathcal{X}_{\max} \\ & \mathcal{U}_{\min} \le \mathcal{U}(t_k) \le \mathcal{U}_{\max}, \end{aligned} \tag{21}$$

where $\mathcal{X}(t_k)$ and $\mathcal{U}(t_k)$ are state and control vector of the system, respectively. The state vector $\mathcal{X}(t_k)$ is defined as in Eq. (20) and optimised controls are designated to be the acceleration $a_c = \dot{u}_c$ and the rate of change of curvature $\mathrm{dK}_c = \dot{K}_c$. Moreover, in Eq. (21) $F(\mathcal{X}(t_k), \mathcal{U}(t_k))$ is a nonlinear dynamical model and $J(\mathcal{X}(t_k), \mathcal{U}(t_k))$ is an objective function. They are utilised to formulate the optimal control problem. In this case $F(\mathcal{X}(t_k), \mathcal{U}(t_k))$ is the hybrid model presented in Section 2.3. As can be seen from Eq. (21), formulating the control problem like this offers a way to handle state ($\mathcal{X}_{\min}$, $\mathcal{X}_{\max}$) and control ($\mathcal{U}_{\min}$, $\mathcal{U}_{\max}$) constraints during the optimisation.

The objective function $J(\mathcal{X}, \mathcal{U})$ is generally given as

$$\sum_{k=0}^{N-1} \left(\|\mathcal{X}(t_{k+1}) - \mathcal{X}_{\mathrm{ref}}(t_{k+1})\|_Q^2 + \|\mathcal{U}(t_k) - \mathcal{U}_{\mathrm{ref}}(t_k)\|_R^2 \right) \tag{22}$$

where N is the prediction horizon and $\mathcal{X}(t_{k+1})$ and $\mathcal{U}(t_k)$ being the predicted state and control values at each prediction step, respectively. In turn, $\mathcal{X}_{\mathrm{ref}}(t_{k+1})$ and $\mathcal{U}_{\mathrm{ref}}(t_k)$ are the state and control reference values, respectively. Moreover, Q and R are positive semi-definite matrices, and they are used to give weights to the states and control inputs, respectively. Thus, by minimising Eq. (22) the control values are calculated at each time step. In this case, the objective function depends on two controls a_c and dK_c, and five states, the x and y positions, heading angle (ψ), forward velocity (u), and roll angle (ϕ).

In this paper, the NMPC module is designed by utilising the VIATOC toolkit (Kalmari et al., 2015). The gradient projection method and the following two steps provide a solution to the control problem. The state trajectory is first obtained using the previous control trajectory and integrating the dynamics. The nonlinear dynamics are also linearised in this point. After that, a line-search based optimisation is utilised to solve the problem. In general, the gradient projection method can be utilised to minimise the function in the presence of linear constraints (Rosen, 1960). In VIATOC, the number of iterations can be defined by the user. As mentioned in Kalmari et al. (2015), computation times are often more prioritised than the precise optimal solution found by the NMPC method. Therefore, the possibility to define the number of iterations is important in order to control this factor of the problem. Other reasons for selecting VIATOC include its code generation capabilities as well as its support for automatic differentiation, which computes the Jacobian of the nonlinear system. Finally, the VIATOC allows modification of the objective function of the optimal control problem. The objective function does not have to be quadratic, as mentioned in Kalmari et al. (2015). The limitation is that the first derivatives must be defined in the objective function. Therefore, in VIATOC the path tracking method is utilised, which means that the NMPC tries to track the 2D reference position path.

As mentioned, optimised controls are designated to be the acceleration $a_c = \dot{u}_c$ and the rate of change of curvature $\mathrm{dK}_c = \dot{K}_c$. In

the NMPC method, limits can be set on these variables, preventing rapid changes in the actual control commands u_c and K_c. This kind of approach is also used in Backman et al. (2012). Table 2 shows the general information about the NMPC parameters that were utilised. Table 3 shows the parameters used in the objective function.

As mentioned, the path tracking method is utilised instead of the traditional reference tracking. The cost for the x and y position in the case of reference tracking can be expressed with the following equation:

$$J_{x,y}(t_k) = \|\mathcal{X}_{x,y}(t_{k+1}) - \mathcal{X}_{\mathrm{ref,x,y}}(t_{k+1})\|^2_{Q_{x,y}}, \tag{23}$$

where $J_{x,y}(t_k)$ is the cost related to x y position, $\mathcal{X}_{x,y}(t_{k+1})$ is predicted x y position at predicted time t_{k+1}, the $\mathcal{X}_{\mathrm{ref,x,y}}(t_{k+1})$ is time dependent x y reference position and $Q_{x,y}$ is weighting matrix used with x y position. The difference between path tracking and reference tracking is that in path tracking the path is not time-dependent, but for each predicted position, a point on the path is found that is closest to the predicted position (Kalmari et al., 2017). This results in the following cost for the x and y positions:

$$J_{x,y}(t_k) = \min_i \|\mathcal{X}_{x,y}(t_{k+1}) - \mathcal{X}_{\mathrm{ref,x,y}}(i)\|^2_{Q_{x,y}}, \tag{24}$$

as shown for example in Kalmari et al. (2014). In Eq. (24), $\mathcal{X}_{\mathrm{ref,x,y}}(i)$ is non time dependent x y reference position. The index i^* that minimises Eq. (24) is found in the following way

$$i^* = \arg\min_i((\mathcal{X}_x(t_{k+1}) - \mathcal{X}_{\mathrm{ref,x}}(i))^2 + (\mathcal{X}_y(t_{k+1}) - \mathcal{X}_{\mathrm{ref,y}}(i))^2), \tag{25}$$

which eventually produces the following cost for the x and y states at prediction time t_k

$$J_{x,y}(t_k) = (\mathcal{X}_x(t_{k+1}) - \mathcal{X}_{\mathrm{ref,x}}(i^*))^2 + (\mathcal{X}_y(t_{k+1}) - \mathcal{X}_{\mathrm{ref,y}}(i^*))^2. \tag{26}$$

From there the partial derivatives with respect to x and y states can be calculated as follows

$$\frac{\partial J_{x,y}(t_k)}{\partial x} = 2(\mathcal{X}_x(t_{k+1}) - \mathcal{X}_{\mathrm{ref,x}}(i^*)); \quad \frac{\partial J_{x,y}(t_k)}{\partial y} = 2(\mathcal{X}_y(t_{k+1}) - \mathcal{X}_{\mathrm{ref,y}}(i^*)).$$

To further highlight the difference between these two approaches, the following example can be used, when the reference path is aligned with the x-axis as $\mathcal{X}_{\mathrm{ref,x,y}} = \{\{t = 0\,\mathrm{s}, (0,0)\}, \{t = 1\,\mathrm{s}, (1,0)\}, \{t = 2\,\mathrm{s}, (2,0)\}, \ldots\}$ and the current x y position of the vehicle is (1,1), the partial derivative with respect to x is zero in the path tracking approach. However, in the case of reference tracking, this depends on time and the path. This might cause the vehicle to accelerate to keep up with the time requirement. The same derivative results can also be obtained using distance and line-segment based approach as in Backman et al. (2012).

To make the path tracking search procedure computationally feasible, only the previous closest point and the near path points that follow the previous closest point are examined in the calculations instead of the whole path. Such a procedure considerably decreases the necessary computation times. For the heading, the reference value is tied to the path points, meaning that each found closest point will also contain associated heading reference information. The path tracking approach also needs a forward velocity reference in order for the vehicle to move. In this paper, both the forward velocity and the roll angle have constant reference values.

2.5. Adjusting speed based on roll angle

To take into account rollover avoidance in the NMPC module, this paper uses the following logic. Since the NMPC module predicts the future responses of the nonlinear system, this information can also be used to advantage in a situation where it is suspected that the roll angle will become too large. In such a case, the prediction of the roll angle (ϕ) of the NMPC module is utilised to linearly lower the forward velocity reference (u_{ref}) in the NMPC module. The method is presented in Algorithm 1. The speed reduction starts when the predicted roll angle exceeds 0.20 rad, and the reference speed is reduced to zero when it exceeds 0.25 rad. The speed reduction mechanism is tuned in simulation.

In addition, the weight of the objective function related to the velocity state can be increased in the NMPC module when the velocity reference goes below a certain threshold, for example, in this case below 0.1 $\mathrm{m\,s^{-1}}$. This forces the vehicle to stop faster.

Table 2
NMPC Parameters used.

Parameters	Limits/Values	Units
Forward Acceleration Command (a_c)	±5	$\mathrm{ms^{-2}}$
Change of Curvature Command ($\mathrm{dK_c}$)	±0.5	$\mathrm{m^{-1}s^{-1}}$
Reference Acceleration Command ($a_{c,\mathrm{ref}}$)	0	$\mathrm{ms^{-2}}$
Reference Change of Curvature ($\mathrm{dK_{c,ref}}$)	0	$\mathrm{m^{-1}}$
Reference Velocity (u_{ref})	1.0 or 1.5	$\mathrm{ms^{-1}}$
Reference roll angle (ϕ_{ref})	0	rad
Velocity (u)	[0, 3]	$\mathrm{ms^{-1}}$
Curvature (K)	±0.15	$\mathrm{m^{-1}}$
Roll Angle (ϕ)	±20	degrees
Prediction Horizon (N)	100	–
Discretisation step (h)	0.05	s

Table 3
NMPC weighting parameters of the objective function.

Parameters	Value
Position (x, y)	1
Yaw angle (ψ)	5
Forward velocity (u)	50
Roll angle (ϕ)	0.0001
Forward acceleration (a_c)	0.5
Change of Curvature Command ($\mathrm{dK_c}$)	2

Algorithm 1 Adjusting speed based on roll angle

```
max_roll ← max(controller_roll_trajectory)
if (max_roll > max_roll_prev ∧ max_roll ≥ 0.2) then
    max_roll_prev ← max_roll
    if (max_roll ≥ 0.20 ∧ max_roll < 0.25) then
        speed_ref ← LinearSpeedDecrease();
    else if (max_roll ≥ 0.25) then
        speed_ref ← 0.0
    end if
end if
```

2.6. State estimation

In the state estimation module, the methods and models presented in Badar, Backman and Visala (2024) are used. The state estimation is required in the control loop because the NMPC module needs to get the most current state information in order to function properly. In addition, the actuators of the vehicle have their own dynamics. Real-world measurements also always contain some noise and time delays that need to be estimated. Therefore, the dynamic 6-DOF model and CDEKF are used (Badar, Backman & Visala, 2024; Särkkä & Solin, 2019). Furthermore, the state augmentation method proposed in Moore and Tam (1973) is utilised to filter the delays caused by sensors and actuators throughout the system.

In the state estimation, the same 3D map is utilised as in the NMPC. Therefore, the measurement vector is written as:

$$y_k = (X_{\mathrm{m}}, Y_{\mathrm{m}}, \phi_{\mathrm{m}}, \theta_{\mathrm{m}}, \psi_{\mathrm{m}}, V_{\mathrm{E}}, V_{\mathrm{N}}, V_{\mathrm{U}}, K_{\mathrm{m}}, u_{\mathrm{m}}, \Delta_{\mathrm{k_m}}). \tag{27}$$

In the measurement vector shown in Eq. (27), (X_{m}, Y_{m}) is the spatial position of the VC in inertial frame, ϕ_{m}, θ_{m}, ψ_{m} are measured roll, pitch and yaw angles, respectively. V_{E}, V_{U} and V_{U} are east, north, and up velocities, respectively. These measurements are obtained from the SPAN unit. Furthermore, K_{m}, u_{m} are measured values for the

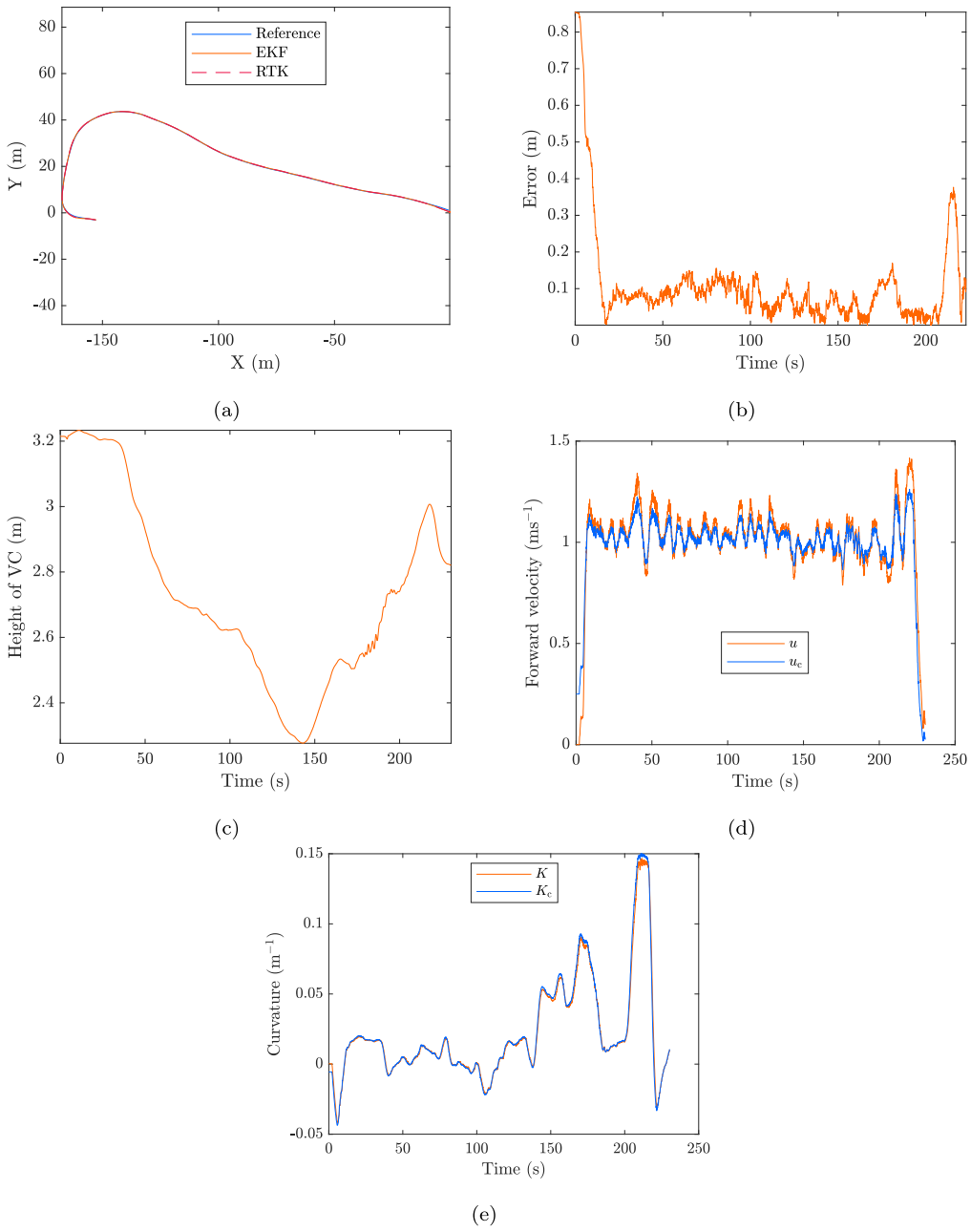


Fig. 10. Results from the Otaniemi path. (a) shows the tracking result in the local ENU frame. (b) shows the absolute tracking error. (c) shows the estimated height of the VC. (d) shows the estimated and commanded velocity. (e) shows the estimated and commanded curvature.

curvature and forward velocity. Finally, Δ_{k_m} is the spring deflection measurement. In turn, these are obtained from the ECUs. Each sensor provides data at 20 Hz rate to the main computer. Therefore, the state estimation module also runs at 20 Hz. The reader should notice that the height measurement (Z_m) from the SPAN unit is not utilised in the process of state estimation, instead the 3D map information presented in Section 2.2 is used inside the state estimator to take into account uneven ground profile. The states of the system and the measurements have the following correspondences

$$\begin{aligned} X_m = X; \quad Y_m = Y; \\ V_E = \dot{X}; \quad V_N = \dot{Y}; \quad V_U = \dot{Z}; \\ \phi_m = \phi; \quad \theta_m = \theta; \quad \psi_m = \psi; \\ K_m = K; \quad u_m = u; \quad \Delta_{k_m} = Z_k - \mathcal{H}_k \end{aligned} \tag{28}$$

Table 4 shows the used state covariance values (**Q**), while Table 5 shows the covariance values (**R**) for the measurements. A more detailed description of the state estimation method can be found in Badar, Backman and Visala (2024).

2.7. Simulation environment

MATLAB Simulink based simulation environment is created and used to test novelties of the proposed method in addition to real field tests. It is important to note that the simulation environment enables testing of features that are difficult to test in a real-world environment, such as rollover testing and a closer look at predicted trajectories. As a vehicle model, the 6-DOF model proposed in Badar, Backman and Visala (2024) is used. The VIATOC toolkit is also able to generate

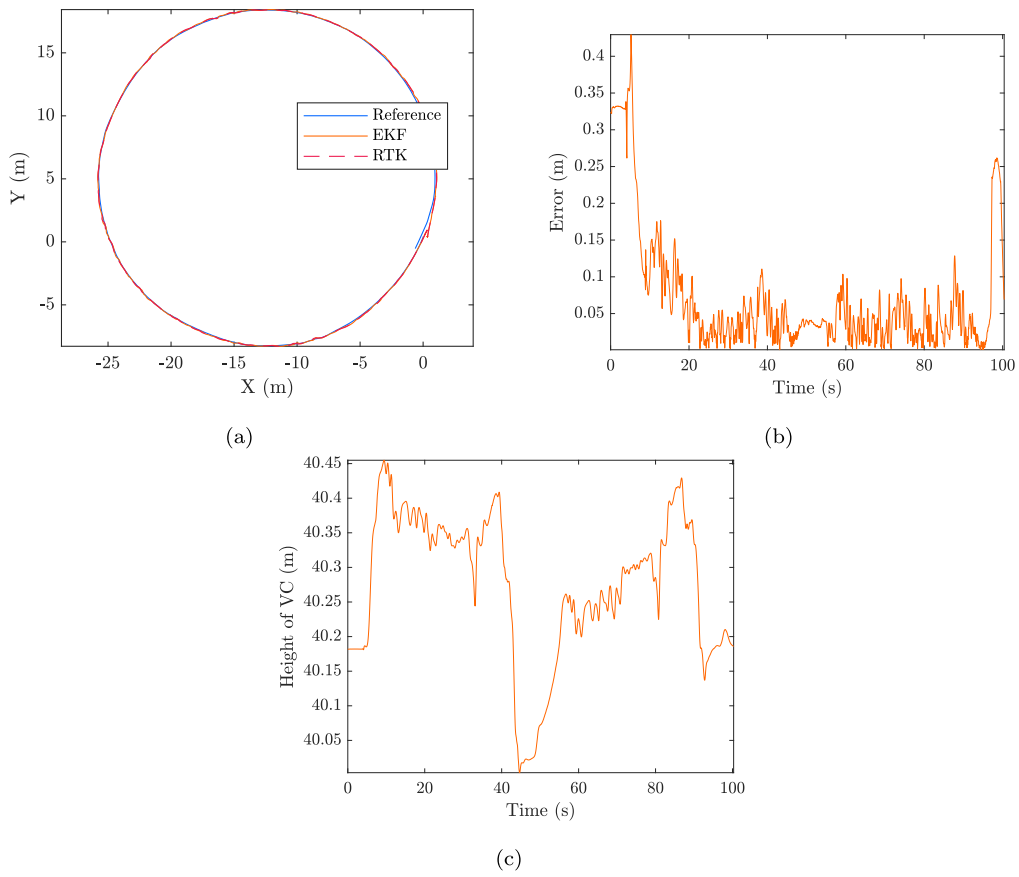


Fig. 11. Results from the Vakola concrete path. (a) shows the tracking result in the local ENU frame. (b) shows the absolute tracking error. (c) shows the estimated height of the VC.

Table 4
EKF state covariance values.

State	Covariance Value
X	1×10^{-6}
Y	1×10^{-6}
Z	1×10^{-8}
u	1×10^{-6}
v	1×10^{-8}
w	1×10^{-8}
ϕ	1×10^{-8}
θ	1×10^{-8}
ψ	1×10^{-8}
p	1×10^{-8}
q	1×10^{-8}
r	1×10^{-8}
K	1×10^{-6}

Table 5
Measurement covariance values.

Measurement	Covariance Value
Δ_{k_m}	1×10^{-5}
X_m	1×10^{-6}
X_m	1×10^{-6}
ϕ_m	2×10^{-6}
θ_m	2×10^{-6}
ψ_m	1×10^{-7}
V_E	7×10^{-5}
V_N	7×10^{-6}
V_U	7×10^{-6}
u_m	1×10^{-6}
K_m	7×10^{-6}

a Simulink S-function that can be used to test the NMPC method in the Simulink simulation environment. In practice, this means that it is possible to test the actual C++ controller code in a closed-loop simulation.

More specifically, two things are tested and analysed in the simulation environment. The first thing to test is rollover avoidance. On the modified Otaniemi path, rollover avoidance is tested using the method presented in Section 2.5. Secondly, the roll, pitch and height predictions obtained from the NMPC method are analysed in the case of Vakola concrete path. This is used to further confirm the functionality of the method.

3. Results

3.1. Field tests

3.1.1. Otaniemi path

This section shows the results from the Otaniemi path when a 1.0 $m\,s^{-1}$ reference velocity was used in the NMPC module. Fig. 10(a) shows the path tracking result, and Fig. 10(b) shows the corresponding tracking error. Fig. 10(c) shows the estimated height of the VC. Figs. 10(d) and (e) show the velocity and curvature commands and the corresponding estimated values, respectively. In Fig. 10(a), the labels Reference, EKF and RTK refer to the reference path, the estimated position of the CDEKF, and the measured position, respectively. This naming convention is also used throughout the results section.

The results indicated that the controller was able to track the path even when a sparse open-source 3D map was used. The mean tracking error was 0.11 m. The results also showed that the controller did not violate the curvature constraint which was 0.15 m^{-1}, instead, it kept the curvature command in limit after 200 s. This can be seen in Fig. 10(e).

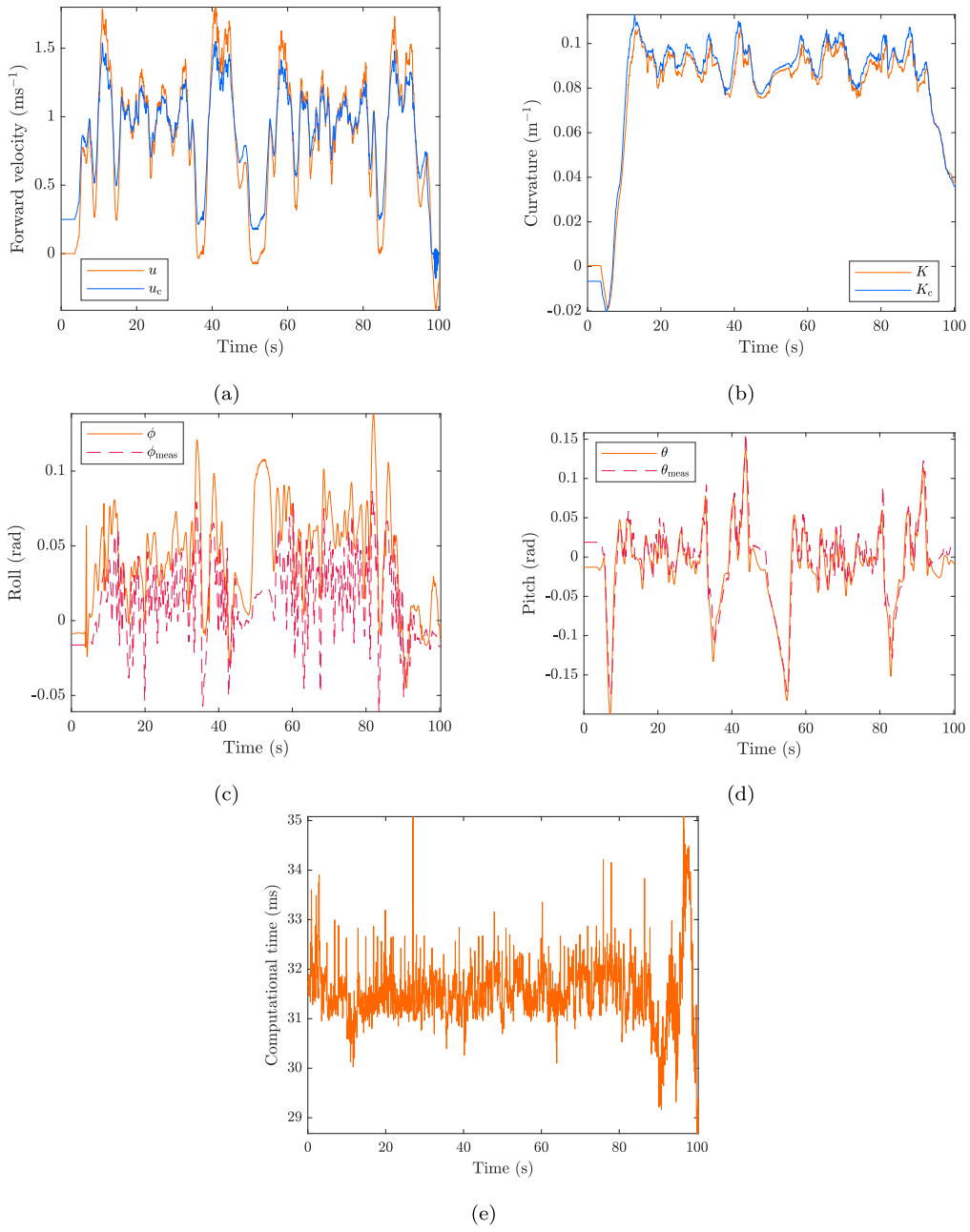


Fig. 12. Results from the Vakola concrete path. (a) shows the estimated and commanded velocity. (b) show the curvature values. (c) and (d) show the estimated and measured roll and pitch values, respectively. (e) shows the computation times of the NMPC module.

3.1.2. Vakola concrete path

This section shows the results from the Vakola concrete path when 1.0 $\mathrm{m\,s^{-1}}$ was used as the reference velocity in the NMPC module. Figs. 11(a) and (b) show the tracking performance and the corresponding tracking error between the path points and the CDEKF estimate. Fig. 11(c) shows the height estimate of the VC. In turn, Figs. 12(a) and (b) show the estimated and commanded values for both forward velocity and curvature. In Fig. 12(c) and (d), the estimated and measured roll and pitch angles are shown, respectively. Fig. 12(e) shows the computation times of the NMPC module.

The results showed that the proposed NMPC method was able to follow the path relatively accurately, although the path had reasonably large differences in terrain height for the Polaris-type vehicle. The mean tracking error was 0.07 m. The height and terrain variations can be observed in Fig. 11(c) and in Figs. 12(c) and (d). The results also revealed that the velocity control was unable to maintain its 1 $\mathrm{m\,s^{-1}}$ reference velocity, but the velocity varied extensively. On the other hand, it could be seen that for the curvature, the first-order low-pass filter described the dynamics satisfactorily.

From a computational point of view, the proposed control method was capable of calculating the control values in the required time when an Asus® (ROG Zephyrus M16) laptop was used that used an Intel® i9-12900H ×20 CPU 5.00 GHz processor.

3.1.3. Forest path

This section shows the results from the Forest path when the 1.5 $\mathrm{m\,s^{-1}}$ reference velocity was used in the NMPC module. Figs. 13(a) and (b) show the tracking performance and the corresponding tracking

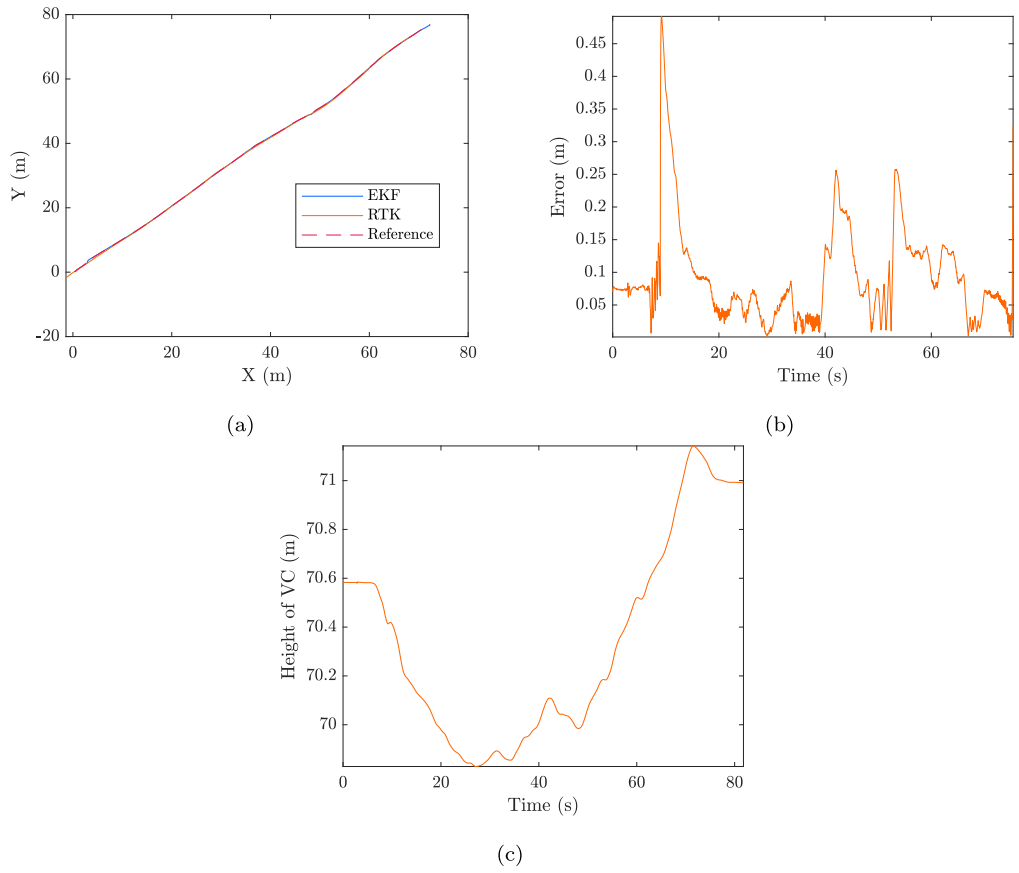


Fig. 13. Results from the Forest path. (a) shows the tracking result in the local ENU frame. (b) shows the absolute tracking error. (c) shows the estimated height of the VC.

Table 6
Path tracking error summary.

Path tracking errors/path	Vakola	Forest	Otaniemi
Avg error (m)	0.07	0.11	0.10
Max error (m)	0.43	0.85	0.49

error between the path points and the CDEKF estimate. Fig. 13(c) shows the estimated height of the VC. Figs. 14(a) and (b) show the estimated and commanded values for both forward velocity and curvature, respectively. In Fig. 14(c) and (d), the estimated and measured roll and pitch angles are shown, respectively.

The results showed that the calculated absolute error contained jumps. The mean absolute tracking error was 0.10 m. It was observed that in the forest path, the controller was able to keep the vehicle velocity close to the reference velocity. It could be observed that for the curvature dynamics, the first order low pass filter based dynamics were suitable for the Forest path.

3.1.4. Result summary from field tests

Table 6 summarises the path-tracking errors, reporting the average and maximum errors in all three test paths. The results indicated that the proposed control method performed reliably across various environments and terrains in terms of path-tracking accuracy.

3.2. Simulation tests

3.2.1. Rollover results

This section shows the simulated rollover avoidance results from the modified Otaniemi path when the reference velocity of 1.0 $\mathrm{m\,s^{-1}}$ was used in the NMPC module. Fig. 15(a) shows the xy position of the VC and the reference path. Figs. 15(b) and (c) show the roll and pitch angles from the simulation. Figs. 15(d) and (e) show both the velocity and curvature commands, the velocity and curvature of the 6-DOF simulation plant, as well as the acceleration and the rate of change of curvature commands. Figs. 16(a) and (b) show the roll and pitch angles when the rollover avoidance method is not used. In turn, in Fig. 17, the roll and pitch angles, and the wheel height predictions are shown from selected time instances. The NMPC module utilised these roll angle predictions when reducing the velocity reference.

The results showed that the NMPC module was able to predict the dynamic behaviour of the vehicle and that the NMPC module lowered the vehicle's forward velocity if the roll angle limits shown in Algorithm 1 were exceeded in the predictions. The NMPC module achieved this by observing the roll angle predictions and calculated the reference velocity based on the method shown. A comparison of Fig. 15(b) and Fig. 16(a) shows that the rollover avoidance method was used to reduce vehicle speed and ensured that the roll angle remained within the allowable range.

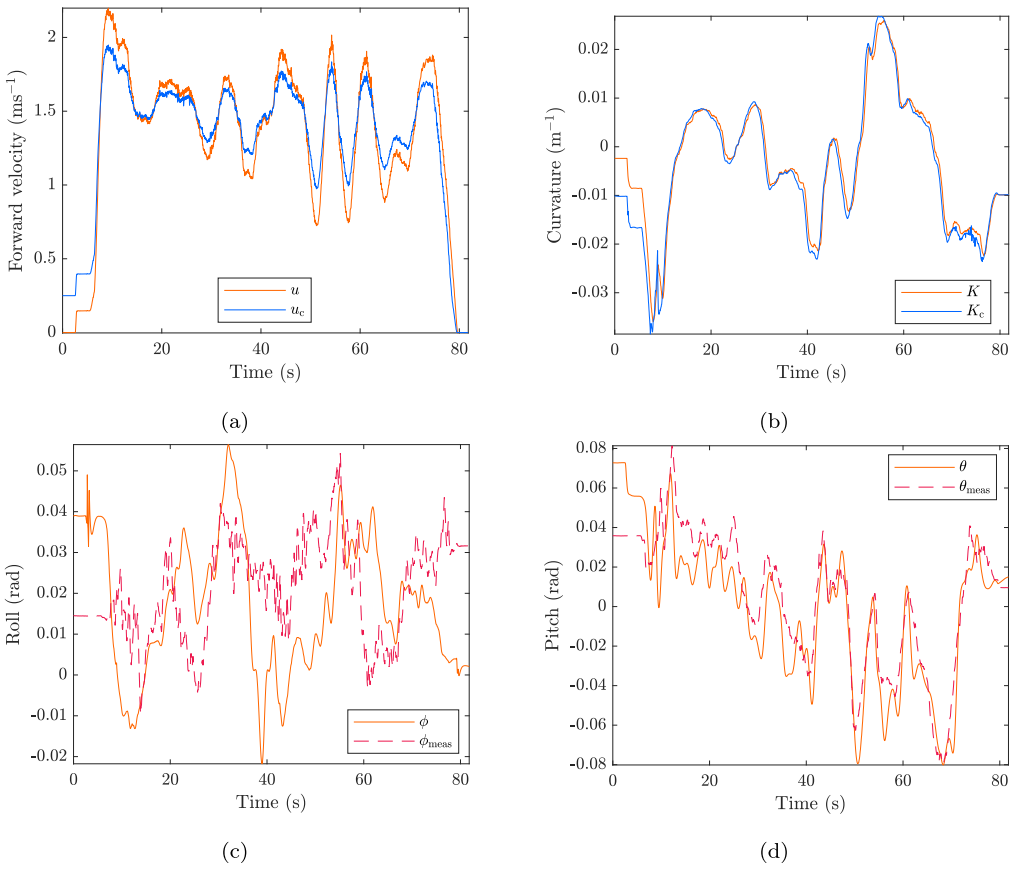


Fig. 14. Results from the Forest path. (a) shows the estimated and commanded velocity. (b) shows the estimated and commanded curvature. (c) and (d) shows the estimated and measured roll and pitch values, respectively.

3.2.2. Analysis of NMPC predictions

This section examines the NMPC predictions in more detail in the same Vakola concrete path that was used in real-world tests. The section concentrates in particular on the evaluation of the VC height, pitch, and roll angle predictions. Figs. 18(a) and (b) show the pitch prediction at times 3 s and 5 s from the NMPC module as well as the actual 6-DOF pitching dynamics between the 3 s–8 s and 5 s–10 s. Figs. 18(c) and (d) present the corresponding results for the Z position, and Figs. 18(e) and (f) for the roll angle.

The results showed that the NMPC module was able to predict the pitch and roll angle sufficiently well when compared with the 6-DOF output. The results also indicated that, even though the Z position prediction overshooted, the NMPC module was still able to predict angles with sufficient accuracy.

4. Discussion

The path tracking results showed that the proposed method was suitable for various terrains and environments, including forests. The results from the forest path showed that the estimated VC contained jumps. This was because within a dense forest, the position methods based on RTK-GNSS positioning are not reliable (Badar et al., 2023). Due to this fact, the state estimation result was not as consistent within the forest. To solve jumps in estimation, LiDAR-based positioning could be utilised in the future within the forest environment (Hyyti & Visala, 2013; Ouattara et al., 2022). Despite the jumps, the controller remained stable.

The results indicated that the NMPC controller was able calculate optimal control values in real time. The execution frequency was set to 20 Hz and the prediction horizon was kept at 100 steps, it practically meant that the NMPC module predicted the dynamic responses of the system by about 5 m when the velocity was 1.0 m s^{-1}. This was shown to be enough for the rollover avoidance method shown in this study.

The NMPC module was able to predict the dynamic behaviour of the vehicle with sufficient accuracy. This was demonstrated in the simulation environment, where the controller was shown to be able to limit the roll angle and eventually stop the vehicle. It was also shown that the controller angle predictions matched well with the 6-DOF model. The overshooting of the Z position prediction was due to the fact that the hybrid model was kinematic in nature.

In the NMPC module, the path-tracking approach was used instead of the traditional reference tracking. This approach allowed the vehicle position to be not dependent on time, which was shown to be a suitable approach for this type of control task. This required modifications to the regular quadratic objective function, since the closest point for each predicted trajectory point needed to be searched. The VIATOC toolkit offered a way to modify the objective function of the NMPC. The weights of the objective function were found using the trial-and-error method first in the simulation environment and later tuned using a real vehicle. For example, the position weight of the objective function was quite low when compared with the velocity weight. This was done because of the practical limitations of the test platform's low-level speed controller (FF-PID).

Moreover, on the Otaniemi and Forest paths, the results indicated that the low-level speed controller performed better than it did on the

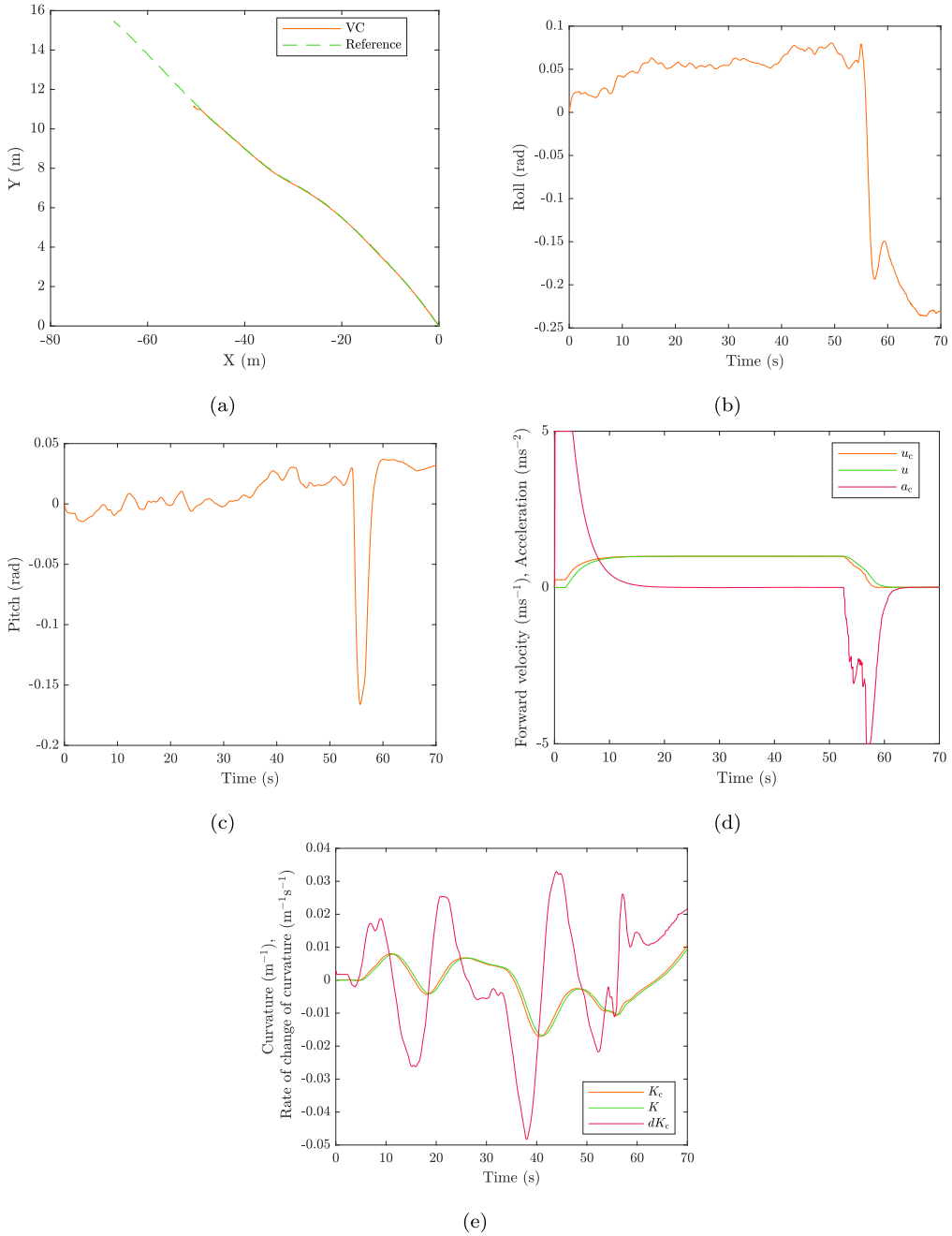


Fig. 15. Simulation results from the modified Otaniemi path. (a) shows the xy position and reference path in the local ENU frame. (b) and (c) show the roll and pitch values, respectively. (d) and (e) show the command values as well as corresponding plant values.

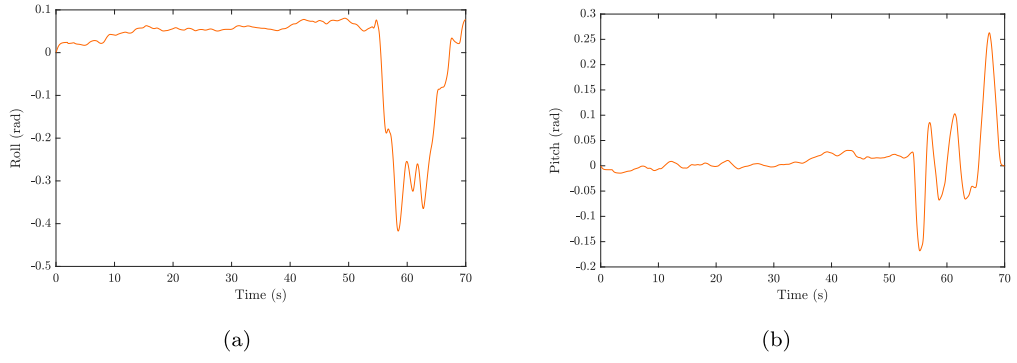


Fig. 16. Simulation results from the modified Otaniemi path. (a) and (b) show roll angles and pitch angles if roll-over avoidance method is not used.

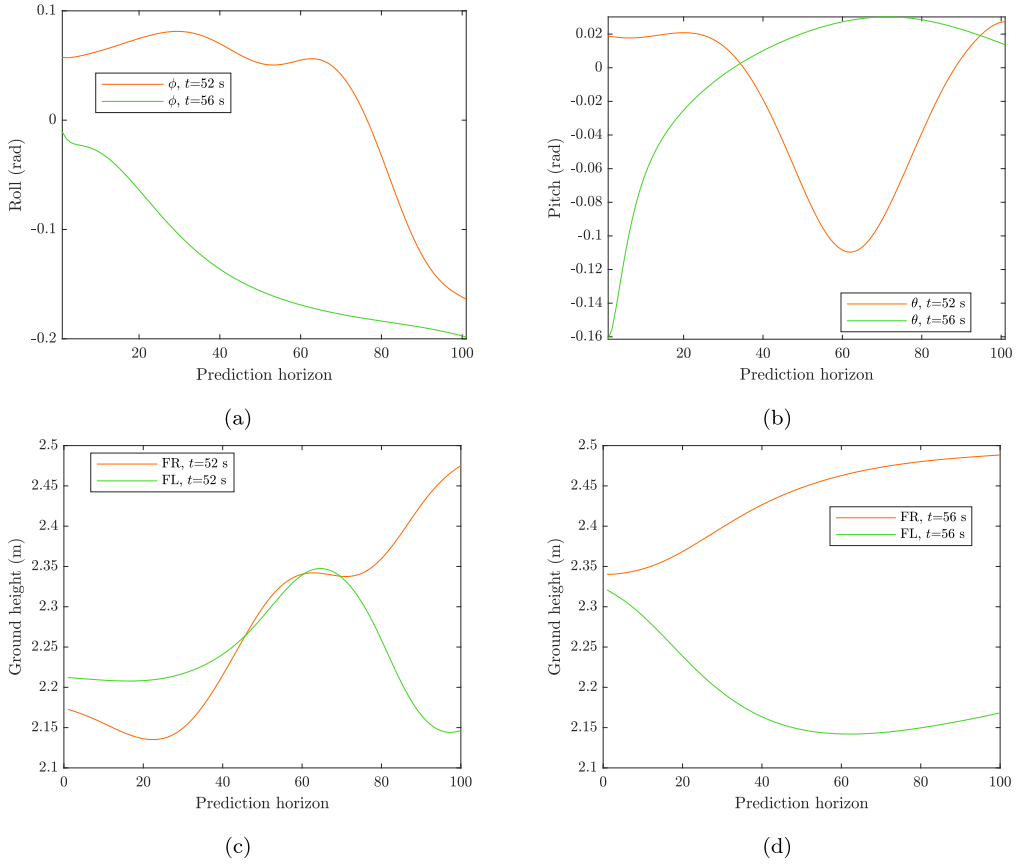


Fig. 17. Simulation results from the modified Otaniemi path. (a) and (b) show the roll angle and the pitch angle predictions at two time instances, respectively. (c) and (d) show the ground height predictions under front right and front left wheels at two time instances.

Vakola concrete track. This was due to the fact that the NMPC assumed the ideal first-order low-pass filter-type velocity dynamics, which basically implied that the low-level controller should be able to realise and maintain the velocity command calculated by the NMPC. This meant that the low-level controller should do the compensation, for example, when driving up a steep hill. To get better velocity control results on highly uneven ground, the current feed-forward proportional–integral–derivative (FF-PID) type low-level velocity controller should be retuned or adaptive low-level control algorithm should be developed to avoid retuning process between different environments. However, this was not within the scope of this paper.

5. Conclusion

This study has shown a hybrid vehicle model that can be used together with the nonlinear model predictive controller (NMPC) to track paths in uneven terrain. The paper also presented other required components needed to implement the NMPC method in uneven terrains, namely the method to obtain 3D maps of the driving paths, the path tracking approach, and the state estimation method. The real world results with the actual research platform from Otaniemi, Vakola concrete, and Forest paths indicated that the controller can track the path with the required accuracy. In turn, the simulation results showed that the NMPC can reduce vehicle speed by monitoring roll angle prediction and in this way prevent rollover from happening.

CRediT authorship contribution statement

Jere Knuutinen: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tabish Badar:** Writing – review & editing, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Juha Backman:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization. **Arto Visala:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

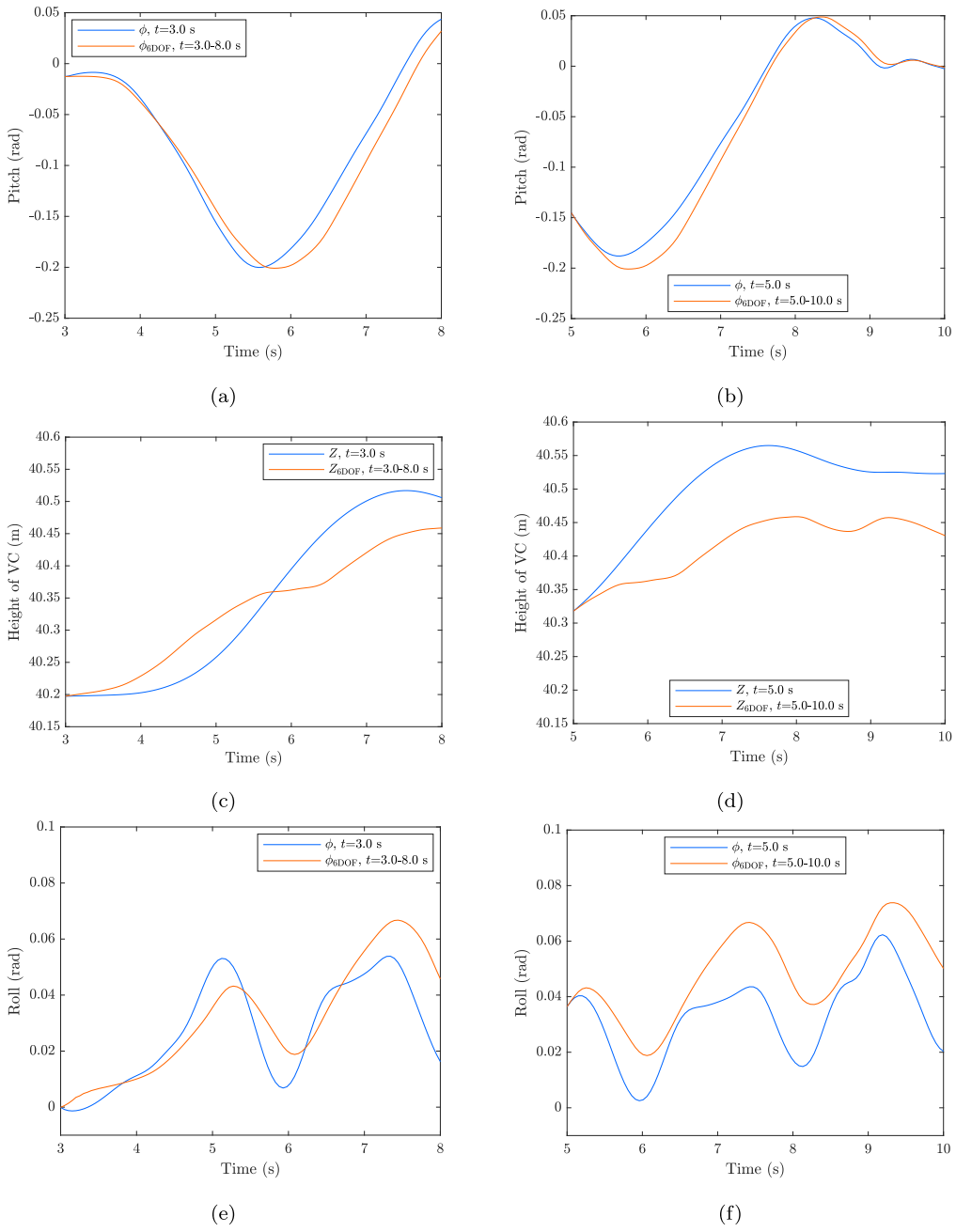


Fig. 18. Simulation results from the Vakola concrete path. (a)–(b) show pitch predictions at 3 s and 5 s and the corresponding 6-DOF pitch dynamics over 3–8 s and 5–10 s. (c)–(f) show the same data for the Z position and the roll angle.

References

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. http://dx.doi.org/10.1016/j.compag.2011.12.009.

Badar, T. (2024). *Enabling sustainable and cost-efficient semi-autonomous forest machine chain: modeling, estimation and control for autonomous driving in terrain* (Ph.D. thesis), (Publication No. 217/2024), Espoo, Finland: Aalto University, School of Electrical Engineering, URL https://aaltodoc.aalto.fi/handle/123456789/131416. Available via Aalto Thesis Database.

Badar, T., Backman, J., & Visala, A. (2024). Vehicle modeling and state estimation for autonomous driving in terrain. *Control Engineering Practice*, *152*, Article 106046. http://dx.doi.org/10.1016/j.conengprac.2024.106046.

Badar, T., Knuutinen, J., Backman, J., & Visala, A. (2024). On NMPC-based rollover avoidance methods for semi-autonomous forest machines. *IFAC-PapersOnLine*, *58*(28), 318–323. http://dx.doi.org/10.1016/j.ifacol.2025.01.014.

Badar, T., Ouattara, I., Backman, J., & Visala, A. (2023). Estimation of 3D form of the path for autonomous driving in terrain. *IFAC-PapersOnLine*, *56*(2), 4916–4921. http://dx.doi.org/10.1016/j.ifacol.2023.10.1264.

Badar, T., Ouattara, I., Backman, J., & Visala, A. (2024). Estimation of the height profile of the path for autonomous driving in terrain. *Computer and Electronics in Agriculture*, *219*(C), Article 108806. http://dx.doi.org/10.1016/j.compag.2024.108806.

Chen, S., & Chen, H. (2020). MPC-based path tracking with PID speed control for autonomous vehicles. *IOP Conference Series: Materials Science and Engineering*, *892*(1), Article 012034. http://dx.doi.org/10.1088/1757-899X/892/1/012034.

Chen, S., Chen, H., & Negrut, D. (2020). Implementation of MPC-based path tracking for autonomous vehicles considering three vehicle dynamics models with different fidelities. *Automotive Innovation*, *66*(3), 386–399. http://dx.doi.org/10.1007/s42154-020-00118-w.

Etkin, B., & Reid, L. D. (1995). *Dynamics of flight: Stability and control* (3rd ed.). USA: John Wiley and Sons.

Hu, C., Wang, R., Yan, F., & Chen, N. (2016). Output constraint control on path following of four-wheel independently actuated autonomous ground vehicles. *IEEE*

Transactions on Vehicular Technology, *65*(6), 4033–4043. http://dx.doi.org/10.1109/TVT.2015.2472975.

Hyyti, H., & Visala, A. (2013). Feature based modeling and mapping of tree trunks and natural terrain using 3D laser scanner measurement system. *IFAC Proceedings Volumes*, *46*(10), 248–255. http://dx.doi.org/10.3182/20130626-3-AU-2035.00065, 8th IFAC Symposium on Intelligent Autonomous Vehicles.

Imine, H., Fridman, L. M., & Madani, T. (2012). Steering control for rollover avoidance of heavy vehicles. *IEEE Transactions on Vehicular Technology*, *61*(8), 3499–3509. http://dx.doi.org/10.1109/TVT.2012.2206837.

Kalmari, J., Backman, J., & Visala, A. (2014). Nonlinear model predictive control of hydraulic forestry crane with automatic sway damping. *Computers and Electronics in Agriculture*, *109*, 36–45. http://dx.doi.org/10.1016/j.compag.2014.09.006.

Kalmari, J., Backman, J., & Visala, A. (2015). A toolkit for nonlinear model predictive control using gradient projection and code generation. *Control Engineering Practice*, *39*, 56–66. http://dx.doi.org/10.1016/j.conengprac.2015.01.002.

Kalmari, J., Backman, J., & Visala, A. (2017). Coordinated motion of a hydraulic forestry crane and a vehicle using nonlinear model predictive control. *Computers and Electronics in Agriculture*, *133*, 119–127. http://dx.doi.org/10.1016/j.compag.2016.12.013.

Lee, S., Wolberg, G., & Shin, S. (1997). Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, *3*(3), 228–244. http://dx.doi.org/10.1109/2945.620490.

Lee, S., Yakub, F., Kasahara, M., & Mori, Y. (2013). Rollover prevention with predictive control of differential braking and rear wheel steering. In *2013 6th IEEE conference on robotics, automation and mechatronics* (pp. 144–149). http://dx.doi.org/10.1109/RAM.2013.6758574.

Liu, J., Jayakumar, P., Stein, J. L., & Ersal, T. (2016). A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles. *Vehicle System Dynamics*, *54*(11), 1629–1650. http://dx.doi.org/10.1080/00423114.2016.1223863.

Moore, J. B., & Tam, P. K. (1973). Fixed-lag smoothing for nonlinear systems with discrete measurements. *Information Sciences*, *6*, 151–160. http://dx.doi.org/10.1016/0020-0255(73)90032-7.

National Land Survey of Finland (2024). Maps and spatial data. http://www.maanmittauslaitos.fi/en/maps-and-spatial-data. (Accessed 30 July 2024).

Ouattara, I., Korhonen, V., & Visala, A. (2022). Lidar-odometry based UAV pose estimation in young forest environment. *IFAC-PapersOnLine*, *55*(32), 95–100. http://dx.doi.org/10.1016/j.ifacol.2022.11.121.

Rosen, J. B. (1960). The gradient projection method for nonlinear programming. Part I. Linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, *8*(1), 181–217. http://dx.doi.org/10.1137/0108011.

Särkkä, S., & Solin, A. (2019). Applied stochastic differential equations. In *Institute of mathematical statistics textbooks*, Cambridge University Press.

Schofield, B. (2006). *Vehicle dynamics control for rollover prevention* (Ph.D. thesis), Department of Automatic Control, Lund Institute of Technology, Lund University.

Shim, T., & Ghike, C. (2007). Understanding the limitations of different vehicle models for roll dynamics studies. *Vehicle System Dynamics*, *45*(3), 191–216. http://dx.doi.org/10.1080/00423110600882449.

Sun, Z., Wang, R., Meng, X., et al. (2024). A novel path tracking system for autonomous vehicle based on model predictive control. *Journal of Mechanical Science and Technology*, *38*, 365–378. http://dx.doi.org/10.1007/s12206-023-1230-y.

Tan, Q., Qiu, C., Huang, J., Yin, Y., Zhang, X., & Liu, H. (2022). Path tracking control strategy for off-road 4WS4WD vehicle based on robust model predictive control. *Robotics and Autonomous Systems*, *158*, Article 104267. http://dx.doi.org/10.1016/j.robot.2022.104267.

Wang, L., Chen, S., & Ren, H. (2024). An accurate trajectory tracking method for low-speed unmanned vehicles based on model predictive control. *Scientific Reports*, *14*, 10739. http://dx.doi.org/10.1038/s41598-024-60290-5.

Ye, B.-L., Niu, S., Li, L., & Wu, W. (2023). A comparison study of kinematic and dynamic models for trajectory tracking of autonomous vehicles using model predictive control. *International Journal of Control, Automation, and Systems*, *21*(9), 3006–3021. http://dx.doi.org/10.1007/s12555-022-0337-8.

Yin, C., Xu, B., Chen, X., Qin, Z., Bian, Y., & Sun, N. (2020). Nonlinear model predictive control for path tracking using discrete previewed points. In *2020 IEEE 23rd international conference on intelligent transportation systems* (pp. 1–6). http://dx.doi.org/10.1109/ITSC45102.2020.9294173.

Yu, S., Shen, C., & Ersal, T. (2021). Nonlinear model predictive planning and control for high-speed autonomous vehicles on 3D terrains. *IFAC-PapersOnLine*, *54*(20), 412–417. http://dx.doi.org/10.1016/j.ifacol.2021.11.208, Modeling, Estimation and Control Conference MECC 2021.