Digital Twin and Trajectory Tracking for Articulated Dump Trucks

Digitaler Zwilling und Trajektorienfolgeregelung für Knickgelenkte Muldenkipper

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen
Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Arash Shahirpour

Berichter/in: Univ.-Prof. Dr.-Ing. Heike Vallery
Univ.-Prof. Dr.-Ing. Elisabeth Clausen

Tag der mündlichen Prüfung: 29.10.2025

Abstract

As the demand for automation for mining industries increases, the need for autonomous transport vehicles becomes increasingly important. Articulated dump trucks (ADT), one of the main transport vehicles in mining operation, pose challenges due to their articulated steering mechanism. While considerable progress has been made in autonomous vehicle research, the dynamic modeling and specific control methods for ADTs remain underexplored. This thesis investigates two main aspects in the deployment of ADT in mining environments: simulation and control. The simulation aspect focuses on the challenges of dynamic modeling, especially due to the unique steering mechanism in ADTs. Therefore, a dynamic model is initially developed that emphasizes the indirect incorporation of steering dynamics to preserve model differentiability. This indirect approach allows the model to be used in model-based algorithms, extending its application beyond simulation alone.

Following the successful implementation of the simulation environment, this thesis investigates Model-Based Predictive Controller (MPC) to achieve trajectory following for ADTs. The goal is to develop control strategies that support the full operational routine of an ADT. Given that phase delays and system constraints are inherent in ADTs, the MPC approach offers a viable solution due to its ability to incorporate system models, including their delays.

Furthermore, this thesis presents the models required for the MPC, covering various operational scenarios and accounting for system delays. These delays have been identified through extensive testing and system identification. However, the system identification process in this work relies on offline methods, meaning it does not adapt to changing conditions in real time.

In addition, this thesis investigates the integration of sideslip angle estimation and compensation as an important factor influencing control performance, especially while cornering. The results show that sideslip compensation improves control performance in such maneuvers. This thesis also differentiates between full-sized and compact ADTs, emphasizing that the faster steering dynamics of compact ADTs require tailored system models. All proposed methods have been validated through simulations and experimental setups, confirming the effectiveness and adaptability of the developed control strategies for real-world mining applications.

Kurzfassung

Mit steigender Automatisierungsnachfrage in der Bergbauindustrie gewinnen autonome Transportfahrzeuge an Relevanz. Als eines der wichtigsten Transportfahrzeuge im Bergbau stellen knickgelenkte Muldenkipper (ADT) jedoch aufgrund ihres speziellen Lenkmechanismus eine regelungstechnische Herausforderung dar. Während in der Forschung im Bereich der autonomen Fahrzeuge bereits erhebliche Fortschritte erzielt wurden, sind die dynamische Modellierung und die Regelung von ADTs noch nicht ausreichend erforscht. In dieser Arbeit werden zwei Hauptaspekte des Einsatzes von ADTs untersucht: Simulation und Regelung. Der Simulationsaspekt konzentriert sich auf die durch den Lenkmechanismus hervorgerufenen Herausforderungen. Es wird zunächst ein dynamisches Modell entwickelt, bei dem schwerpunktmäßig die Lenkdynamik indirekt einbezogen wird, um die Differenzierbarkeit des Modells zu gewährleisten. Dieser indirekte Ansatz erlaubt die Verwendung des Modells in modellbasierten Algorithmen und ermöglicht seine Anwendbarkeit über die reine Simulation hinaus.

Anschließend wird in dieser Arbeit die modellbasierte prädiktive Regelung (MPC) zur Trajektorienfolgeregelung untersucht. Ziel ist es, Regler zu entwickeln, die den gesamten Betriebsablauf eines ADTs unterstützen. In Anbetracht der Tatsache, dass Verzögerungen und Beschränkungen den ADTs inhärent sind, bietet der MPC-Ansatz eine praktikable Lösung, da er in der Lage ist, Systemmodelle, einschließlich ihrer Verzögerungen, zu integrieren. Außerdem werden in dieser Arbeit die für die MPC erforderlichen Modelle vorgestellt, die verschiedene Betriebsszenarien abdecken und die Systemverzögerungen berücksichtigen. Diese Verzögerungen wurden durch umfangreiche Tests und Systemidentifikation ermittelt. Die Systemidentifikation in dieser Arbeit beruht jedoch auf Offline-Methoden, sodass diese sich nicht an veränderte Bedingungen in Echtzeit anpassen lässt.

Darüber hinaus wird in dieser Arbeit die Integration einer Schwimmwinkelschätzung und -kompensation als wichtiger Einflussfaktor auf die Regelgüte – insbesondere bei Kurvenfahrten – untersucht. Bei diesen Manövern zeigen die Ergebnisse, dass die Kompensation des Schwimmwinkels die Regelgüte verbessert. In dieser Arbeit wird weiterhin zwischen großen und kompakten ADTs unterschieden, da die schnellere Lenkdynamik von kompakten ADTs maßgeschneiderte Systemmodelle erfordert. Alle vorgeschlagenen Methoden werden durch Simulationen und Versuchsaufbauten validiert sowie die Effektivität und Anpassungsfähigkeit der entwickelten Regelungsstrategien für reale Bergbauanwendungen bestätigt.

Contents

Own Publications

This thesis is based in parts on the following own publications:

[M1] Shahirpour, A. ; Weseloh, J. ; Abel, D.: Sideslip Angle and Parameter Estimation of an Articulated Dump Truck Using a Joint Extended Kalman Filter. In: *IFAC-PapersOnLine*. 22nd IFAC World Congress 56. no. 2 (Jan. 2023), pp. 10953–10958. DOI: `10.1016/j.ifacol.2023.10.787`.

[M2] Shahirpour, A. ; Abel, D.: Simulation and Successive Sideslip-Compensating Model Predictive Control for Articulated Dump Trucks. In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. Macau, China: IEEE, Oct. 2022, pp. 3907–3913. DOI: `10.1109/ITSC55140.2022.9922465`.

[M3] Shahirpour, A. ; Framing, C.-E. ; Zweigel, R. ; Abel, D.: Systemidentifikation und modellbasierte prädiktive Pfadfolgeregelung eines automatisierten knickgelenkten Muldenkippers; 2. Auflage. de. In: *Mechatronik 2021 : Darmstadt*. 2021, pages 140–145.

[M4] Shahirpour, A. ; Reuscher, T.: Design and Experimental Evaluation of Model Predictive Control for Autonomous Articulated Dump Trucks. In: *Not Published Under Review* (2025).

[M5] Varga, B. ; Shahirpour, A. ; Burkhardt, Y. ; Schwab, S. ; Hohmann, S.: Validation of Cooperative Shared-Control Concepts for Large Vehicle-Manipulators. en. In: *2020 IEEE Conference on Control Technology and Applications (CCTA)*. Montreal, QC, Canada: IEEE, Aug. 2020, pp. 542–548. DOI: `10.1109/CCTA41146.2020.9206372`.

The following provides an overview of these publications, their contributions to this thesis, and the author's specific role in each.

Publication [M3] was written by the author and reviewed by the second and third authors. The core ideas and implementations were developed by the author, with the other authors contributing to the establishment of the first iteration of the kinematic simulation and the first identification tests. The content of this paper is translated and rewritten for use in Chapters 3 and 4.

Publication [M2], was written, implemented, designed and executed by the author. Other authors contributed by reviewing the manuscript. This content of this paper is rewritten for use in Chapters 3 and 4.

In publication [M1], the author was responsible for writing, conceptualizing, and designing. The implementation of the filters, however, was carried out as part of a master's thesis supervised by the author. Furthermore, the other authors contributed through manuscript review. This paper was rewritten and used in Chapter 7.

Publication [M4] was written, designed, implemented, and executed solely by the author. The manuscript was reviewed by the other authors. This paper is to a large extent reproduced in Chapter 5.

Publication [M5] was conceptualized and written by Balint Varga, based on student theses, including the author's own Master thesis. This paper uses a modeling concept, which was also used in this work in Chapter 3.

List of Figures

List of Tables

List of Abbreviations

ADT	Articulated Dump Truck
ARTUS	Autonomes robustes Transportsystem für hybride umweltschonende Rohstoffgewinnung auf Basis knickgelenkter Sonderfahrzeuge
BPA	Best Practice Analysis
c-ADT	Compact Articulated Dump Truck
CoG	Center of Gravity
DBC	CAN Database Container File
EKF	Extended Kalman Filter
f-ADT	Full-Sized Articulated Dump Truck
FAR	Front Axle Referencing
GPR	Gaussian Process Regression
HIL	Hardware in the Loop
IMU	Inertial Measurement Unit
IPC	Industrial Personal Computer
JEKF	Joint Extended Kalman Filter
LQR	Linear Quadratic Regulator
MAE	Mean Absolute Error
MPC	Model-Based Predictive Controller
NMPC	Nonlinear Model Predictive Control
ODE	Ordinary Differential Equations
PLC	Programmable Logic Controller
RAR	Rear Axle Referencing
RMSE	Root Mean Square Error
RTK	Real-Time Kinematic
SIL	Software in the Loop
2D	Two Dimensional

List of Symbols

Standards

x	scalar
$\boldsymbol{x}$	vector
$\boldsymbol{X}$	matrix
$\boldsymbol{f}(\cdot)$	vector function

General Symbols

$\boldsymbol{x}$	system state vector
$\boldsymbol{w}$	process noise
$\boldsymbol{u}$	system input vector
$\boldsymbol{v}$	measurement noise
$\boldsymbol{y}$	system output vector
$\boldsymbol{Q}$	Kalman filter process noise covariance matrix
$\boldsymbol{R}$	Kalman filter measurement noise covariance matrix
t	time
$\mathcal{N}(\mu, \sigma^2)$	normal (Gaussian) distribution with mean μ and variance σ^2
$\boldsymbol{A}$	system matrix
$\boldsymbol{B}$	input matrix
$\boldsymbol{C}$	output matrix
T_s	sampling time
K	gain
x, y	position vector inertial coordinates
ϕ	articulation angle
ω	articulation angle rate
α	sideslip angle at the front axle
β	sideslip angle at the rear axle
$\psi_{1/2}$	yaw angle of the front/rear vehicle body
$L_{1/2}$	distance between the front/rear axle and the articulation hinge

$v_{x_{1/2}}$	longitudinal element of the velocity of the front/rear axle
$v_{y_{1/2}}$	lateral element of the velocity of the front/rear axle
$(x_{1/2}, y_{1/2})$	position of the front/rear axle in inertial coordinates
$v_{1/2}$	speed of the front/rear axle
$a_{x_{1/2}}$	longitudinal element of the acceleration of the front/rear axle
$a_{y_{1/2}}$	lateral element of the acceleration of the front/rear axle
$tw_{1/2}$	the distance between the wheel centers of the front/rear axle divided by two
C_α	cornering stiffness (lateral slip coefficient) of the tires
C_σ	longitudinal slip coefficient of the tires
Ω	tire rotational speed
$(v_{\mathrm{t},x}, v_{\mathrm{t},y})$	longitudinal and lateral speed elements at the tires in tire-fixed coordinates
r_e	effective tire radius
T_ω, ω_T	articulation angle rate time constant
$(x_{\mathrm{f/r}}, y_{\mathrm{f/r}})$	position of the front/rear center of gravity in inertial coordinates
$(u_{\mathrm{f/r}}, v_{\mathrm{f/r}})$	longitudinal and lateral element of the velocity of the front/rear center of gravity in body-fixed coordinates
$m_{\mathrm{f/r}}$	mass of the front/rear vehicle body
$I_{\mathrm{f/r}}$	moment of inertia of the front/rear vehicle body
(F_x, F_y)	longitudinal and lateral tire forces in tire-fixed coordinates
s	curvilinear abscissa
$\Delta\psi$	orientation error
ψ_Γ	orientation of the path's tangent
κ	curvature
T_d	dead time

Subscripts

ref	reference value
se-base	state estimation using the baseline version of the kinematic-slip model
se-ext	state estimation using the extended version of the kinematic-slip model
se-dyn	state estimation using the dynamic model
FAR	front axle referencing
RAR	rear axle referencing
c	referring to the c-ADT
f	referring to the f-ADT
op	operation point
dyn	referring to the dynamic model
red	referring to the reduced (simplified) dynamic model
kin	referring to the kinematic model

base	referring to the baseline kinematic-slip model
ext	referring to the extended kinematic-slip model
pf	path-following
tf	trajectory-following
se	state estimation
Fr	referring to the Frenet model
d	time discrete
bl	baseline
pf-Fr	referring to the path-following MPC using the model in Frenet coordinates
tf-kin$_\mathrm{f}$	referring to the trajectory-following MPC using the FAR$_\mathrm{f}$ model
tf-slip	referring to the trajectory-following MPC using the kinematic-slip model

1 Introduction

1.1 Motivation

The rising demand for finite raw materials and natural resources will force humanity to face new challenges in sustaining the quality of life on Earth. The concept of sustainable development, popularized in 1987 after the publication of the Brundtland Report by the United Nations [18], is one of the concepts proposed and widely recognized to deal with these challenges. This concept is essential to not only combat the depletion of resources but also to conserve them [29]. In the broadest definition, the pillars of sustainable development are efficient economic growth, protection of natural resources, and social responsibility.

In the context of mining, too, the sustainable development should be realized. The social responsibility aspect is particularly critical, as mining operations inherently involve hazardous working conditions, such as risks of accidents, exposure to harmful materials, and physically and mentally demanding labor [10, 29]. Therefore, it is essential to implement strategies that enable mining industries to comply with the principles of sustainable development.

One of the viable approaches promising to support sustainable development in mining is automation. Defined as the use of technology to manage systems and reduce or eliminate human involvement in all or parts of the process [51], automation in mining brings both promising advances and potential challenges. These challenges may include inadvertently heightening environmental impacts and introducing social challenges, both within and outside of the mine, affecting nearby communities and ecosystems [71, 73]. Nevertheless, as the mining sector moves toward sustainable development, automation remains a critical path forward, with its potential to enhance planning, monitoring, and the overall efficiency of extraction and transportation processes [42].

The primary contribution of automation in mining is its potential to create safer working conditions by reducing human presence in hazardous areas, as mining sites frequently present high-risk situations. Examples of automation include using drones or satellites for process monitoring, employing big data and predictive analytics for insights, implementing Internet of Things systems for real-time updates on equipment and processes, and utilizing autonomous or remotely operated vehicles for extraction and transport [41]. Additionally, automation promises to reduce the required and especially the lower-skilled labor while the operators are removed from the site and are instead responsible for monitoring the whole process remotely [97].

One of the promising areas of automation in mining industries is the autonomous transportation of bulk materials. A transportation cycle generally includes parking a mining truck in a probably tight loading area, driving to an unloading site by navigating through mostly rough and unmarked terrains or tunnels, and reversing into an unloading area. Drivers have to constantly monitor the surroundings and be careful and mindful of others in a repetitive task in a poorly visible environment. The combination of such monotonous operations with long hours and shift schedules leads to exhaustion and fatigue in operators [10]. This is especially challenging since no traffic regulations apply when no marking is present on the surface and no zone is strictly dedicated to heavy vehicles, smaller trucks, or pedestrians. Figure 1.1 shows the unmarked ground of the mining fields where the experiments of this work were conducted.

Autonomous transportation of mining bulk materials has the potential to enhance productivity, reliability, and flexibility by reducing the on-site workforce needed for transport and extending productive operational hours [11]. Additionally, depending on the specific sensors and actuators implemented, automation can deliver more consistent results by minimizing the impact of varying operating conditions, such as weather, lighting, and temperature.

Besides the economical benefits, an autonomous transportation process contributes to better and safer working conditions for employees, especially for the workers in underground mining, by reducing their involvement and human errors. Furthermore, considering that trucks in the transport process in mines are involved in more fatal

Fig. 1.1: Bird eye's view of the testing mining fields. The left picture is a limestone mine in Steyrling, Austria [30], and the right picture is a gravel pit in Buir, Germany [52].

accidents compared to all other mining equipment [97], such automation has the potential to reduce fatal accidents [94].

In this context, project ARTUS (Autonomes robustes Transportsystem für hybride umweltschonende Rohstoffgewinnung auf Basis knickgelenkter Sonderfahrzeuge) was introduced. The project was supported by the German Federal Ministry of Education and Research (BMBF), project number: FKZ 033R126DN. The goal in this project is to create an autonomous fleet of articulated dump trucks (ADT) for the transportation of bulk materials in mining areas. The reason for choosing ADTs is that they are one of the primary vehicle types widely used in mining operations. Furthermore, in recent years, smaller-sized or compact ADTs have gained popularity in urban construction sites. ADTs' popularity and widespread use are due to the vehicles' capability to carry heavy loads relative to their own weight while maintaining a high maneuverability over rough terrain. This high maneuverability is due to the special steering in these vehicles. ADTs do not steer via steering the front wheels but by changing the orientation of the front body relative to the rear body using cylinders that are mounted on the sides of the pivoting point or the hinge [4]. Figure 1.2 illustrates an ADT with hydraulic cylinders exaggerated in size for better visibility. This specific type of steering makes the high maneuverability of ADTs possible despite their considerable weight but poses challenges to control design, which will be explained in detail later.

The ARTUS project aimed to increase the efficiency and productivity of transportation and at the same time to improve safety for employees. In this project, both compact ADTs (c-ADTs) and full-sized ADTs (f-ADTs) are included in the autonomous fleet. Figure 1.3 shows the ADTs used in this project.

For the objective of this project to come true, five categories of work packages needed to be fulfilled. These are perception and mapping, localization, collision avoidance and

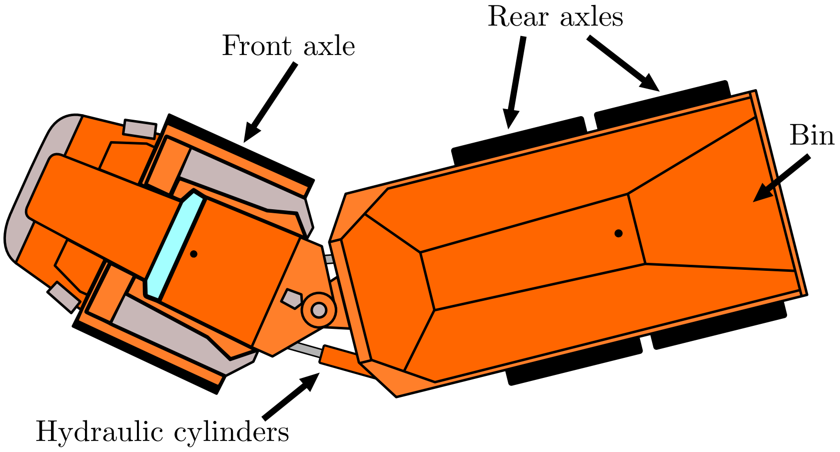


Fig. 1.2: Bird eye's view of an ADT with the hydraulic cylinders.

Fig. 1.3: Full-sized and compact-sized articulated dump truck (f-ADT and c-ADT) in a surface mine.

trajectory planning, trajectory-following controller, and fleet management. Figure 1.4 demonstrates these packages and how they communicate.

- **Fleet management:** This module is responsible for making mission strategy decisions. This includes assigning tasks to different vehicles by allocating them destinations, giving the respective signal to the other modules about the start or end of a transportation task, and to generally manage the movements of all of the vehicles.
- **Perception and mapping:** To create a map of derivable areas and obstacles, methods for environment perception need to be developed, which shall later help plan the trajectory. This is especially a challenge in a mine since the common perception methods that rely on vision sensors cannot be used with impaired vision in mines due to dust and bad weather conditions. Alternatively, radar-based environment perception is proposed for mines and is used in this project [69].
- **Localization:** The performance of the trajectory-following algorithms relies heavily

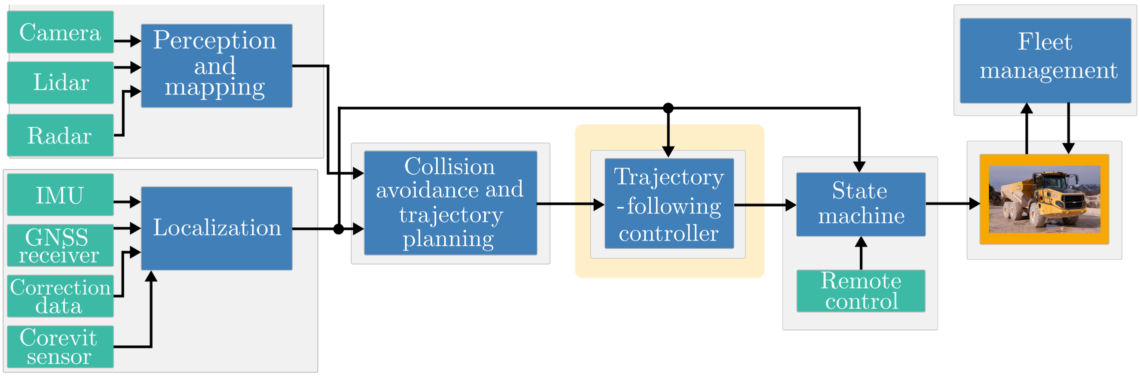


Fig. 1.4: Work packages for the project ARTUS.

on the localization data. The localization module gives information about the current pose of the vehicle.

- **Collision avoidance and trajectory planning :** Based on the data from the previous two modules, an obstacle-free and drivable trajectory needs to be planned from the vehicle's current position to its destination.
- **Trajectory-following controller:** Finally, this module is responsible for giving steering and speed commands to each vehicle, so that the vehicle follows the reference trajectory.

In this work, the trajectory-following for ADTs will be studied, and the other work packages were accomplished by other project partners.

1.2 State of the Art and Requirements

This Section presents an analysis of the state of the art in three categories. The first category is the state of the art in terms of trajectory/path-following controllers, which is the main goal of this work. The second category is the state of the art in terms of modeling the vehicle either for model-based controllers or for simulations. The third category is for developing estimation methods to estimate states that can be used in the controller but are not available. In the following, these aspects are discussed in the same order they are presented in the rest of this work.

1.2.1 ADT Modeling for Simulation: Requirements

In this work, vehicle models are needed for both simulation purposes and implementation in model-based algorithms. A simulation environment is especially valuable to this work since ADTs are not easily accessible for testing purposes. In the context of simulation, model accuracy is the primary concern, as the complexity and computational requirements are less restrictive. Time and computational resources are typically available, allowing for more complex parameterization and tuning processes without significant constraints. However, when models are intended for real-time applications, such as in model-based control, efficiency becomes crucial. Models in this context must demand less computational power to ensure they can run on hardware in real time. While there is no strict statistical requirement, the objective here is to balance accuracy with computational efficiency. In this Section, while the primary focus is on modeling for simulation purposes, consideration is also given to ensuring that the resulting model can be useful for model-based applications, such as controllers or estimation methods. Furthermore, to ensure that the model accurately represents the vehicle, it is essential to incorporate the unique steering mechanism of the ADT, as this is the primary difference between these vehicles and standard passenger cars.

Based on this discussion, the project requirements, and consultations with mine runners and drivers, the following requirements for the ADT model in the simulation have been established.

1. The simulation should use a dynamic model.
2. This model must be differentiable and in the form of ordinary differential equations (ODE) since model-based algorithms rely on using the derivative of the model.
3. The model must define forces, accelerations, and velocities in the vehicle's body coordinates, expressed in the same direction that they affect the vehicle body and are measured, to eliminate the need for further transformations when utilized in model-based algorithms.
4. The simulation environment must have the same inputs as the actual vehicle.
5. The specific dynamics of the steering system must be incorporated into the model to ensure that its effects on the vehicle are fully replicated without compromise.
6. The resulting model must provide improved accuracy in predicting the vehicle's position and moving angle compared to existing kinematic models.

1.2.2 ADT Modeling for Simulation: State of the Art

The following discussion presents the state of the art, highlighting the limitations of existing methods that make them unsuitable for this project. At the end of this Section, a summary of these shortcomings is provided, outlining the research gap that the proposed modeling approach should fulfill.

In the context of autonomous driving and vehicle dynamics, the modeling of an ADT can be generally categorized into two major types: kinematic and dynamic modeling [68]. As the names also suggest, a kinematic model provides a mathematical description of the vehicle's movement, focusing solely on geometric and kinematic relationships without considering the forces acting on the vehicle. Dynamic modeling, however, accounts for forces, torques, mass, and inertia that influence the vehicle's motion, providing a more holistic representation of the vehicle's behavior [68].

A baseline kinematic model for an ADT is presented in [25]. However, this model does not include any of the system phase delays or dead times. The kinematic nature of this model has the benefit that it is mathematically simpler and is easier to derive but also the disadvantage that it neglects the dynamic aspects of the system. This makes this model useful when developing a controller since it does not require considerable computation power or complex tuning. This model will be discussed in detail with the controller in the upcoming Section. However, for simulation purposes, kinematic models are not sufficient, and dynamic models are required.

For the sole purpose of simulation, using physics simulators such as ADAMS [61] or Gazebo [34] is a popular option [37, 98]. In such works, the ADT is simulated in the physics simulator by designing the front and rear bodies of the vehicle and connecting them using hydraulic cylinders. However, these methods do not provide a deep understanding of the system since the simulation is conducted in a physics engine and does not rely on a detailed analytical framework. Another popular option for the simulation is using a kinematic vehicle model in works such as [59]. As discussed earlier, kinematic models are insufficient for the simulation.

Inspiration for ADT models can also come from semi-trailer truck models, despite the differences in vehicle steering. Semi-trailer trucks consist of a tractor unit and a semi-trailer, featuring a combination of a towing unit and a rear load unit similar to ADTs. However, they differ from ADTs in their steering mechanism. While ADTs utilize hydraulic cylinders at the hinge for steering, semi-trailer trucks employ Ackermann steering, allowing the front and rear bodies to move freely without hydraulic cylinders governing their relative rotation. Due to the dynamic similarity between these two vehicles, studying methods of modeling semi-trailer trucks provided the author with valuable information about the dynamics of ADTs. In [82], a model of a semi-trailer is presented, and in [21, 50, 80], the model is derived by using the Euler-Lagrange equation with the generalized forces in the inertial coordinates. However, it is desired that the model utilizes forces, accelerations, and velocities in the vehicle's body coordinates. This is important because, in this framework, the forces acting on the vehicle's body, along with the associated kinematic parameters such as speed, acceleration, and angular velocity, can be represented without the need for transformation to inertial coordinates. This allows for a consistent representation of the system's dynamics, as all relevant quantities remain within the same coordinate system in which they are defined and applied. Therefore, in the following, the focus is set on modeling in vehicle body's coordinates.

Another area of research in the dynamic modeling for ADTs is with explicit modeling of the cylinders. This is especially useful if the goal of the modeling is having access to the cylinders' states, developing controllers for the cylinders, or analyzing the lateral movement and oscillation due to the cylinders ([27, 32, 33, 54, 65]). This type of modeling adds a layer of complexity to the modeling and to the parameterizing of the final model. Furthermore, the nonlinearity added to the system by the cylinders is not directly differentiable due to sign functions in cylinder models. This makes it unnecessarily difficult to use these types of models in model-based algorithms since these methods rely on differentiable models. Therefore, a more suitable approach involves modeling the cylinders implicitly, capturing their dynamic effect on the vehicle bodies without directly employing the mathematical formulation of a cylinder.

To avoid the associated issues when modeling the cylinders explicitly, several approaches have been explored in the literature. In [36], a timberjack grapple skidder, which is an articulated vehicle with an identical steering system to the ADTs, is modeled. The cylinders are once modeled explicitly and once replaced with a controller that finds the torque that the cylinders would generate. While using this controller solves the issues that the explicit modeling of the cylinders presents, discussed above, it fails to exactly behave like the cylinders. For example, it cannot instantaneously and continuously determine the appropriate torque value required to maintain a constant articulation angle when aiming for an articulation angle rate of $0\,\mathrm{rad\,s^{-1}}$. In scenarios where a constant articulation angle or an angle rate of zero is desired, the necessary torque to achieve this varies as the vehicle moves. The effective torque on the hinge needed to maintain a constant angle changes dynamically with the vehicle's motion, making it challenging for a controller to instantaneously calculate and apply the precise torque required to counteract all other forces. In contrast, real vehicle cylinders can maintain an articulation angle by locking the angle through closing the hydraulic valves and making the vehicle behave similarly to a single rigid body. This is not possible when using a controller as suggested by [36].

In [38], the problem with the explicit modeling of the cylinders is addressed, and an idea is discussed to avoid this, but only a simplified dynamic error model is presented, and a slippage-free motion is assumed. This simplification and the slippage-free assumption tailor the model to the special use case of the paper, which is to design a controller for the cylinders. The resulting model is not suitable for simulation, since it cannot represent the dynamic aspects of the vehicle, which is one of the main reasons for establishing a dynamic model. In [26], the cylinders are not explicitly modeled either. However, the model also assumes a slippage-free motion and neglects the dynamic effect of the cylinders on the rear body. This means that the rear yaw angle is calculated by only using the kinematics of the system and not by considering the dynamic effects of the cylinders on the rear body.

In the following, the research gap, in accordance with the requirements outlined at the beginning of this Section, is summarized. Several approaches have been proposed to model an articulated dump truck (ADT) dynamically. Some of these approaches explicitly model the cylinders, which means to establish a vehicle model that is not in the form of ODEs and is not differentiable. Other attempts that model the cylinders implicitly only consider their kinematic effects, neglecting dynamic aspects and failing to capture the cylinders' influence on the vehicle's dynamics. Additionally, these approaches often overlook the essential functionality of the cylinders: maintaining a constant articulation angle to make the vehicle behave as a rigid body when no steering command is available and articulating at a specified angle rate when a command is issued. These gaps must be addressed by the proposed dynamic model.

1.2.3 Control Concepts: Requirements

The primary goal of control in autonomous articulated vehicles is to follow a given track while minimizing lateral error. However, the specific steering of articulated vehicles poses challenges to the task of the controller design. These challenges arise from two main factors: firstly, the relatively limited established approaches available in this area compared to Ackermann-steered vehicles, and secondly, the considerable transport and phase delay in the steering cylinders and the drivetrain of these vehicles. The mechanics of ADT steering and the associated challenges will be explained in detail in Chapter 3, and are assumed to be accepted here without further elaboration.

For the autonomous approach to be considered successful in this project, it has to be able to do the maneuvers that a human driver is able to with at least the same accuracy and the same speed. The required precision depends on the context, including the driving scenario and vehicle size, and is discussed in detail throughout this Section. Additionally, the trajectory-following controller should be able to drive with the same or higher speeds as the human driver, be able to park the vehicle, be able to navigate through narrow passages at lower speeds, and do all the mentioned tasks both when driving forward and backwards [95]. Furthermore, similar to a human driver, the controller must be able to determine both control inputs of the vehicle: articulation and speed command. Lastly, as two sizes of ADTs were involved in this project, the control approach must be able to function on both vehicle sizes.

To better understand the routine operation of an ADT, a best practice analysis (BPA) is performed on the vehicle. The BPA, visualized in Figure 1.5, shows the routine operation of an ADT over 118 h. The Figure illustrates the frequency of occurrences for specific combinations of steering angle and speed. The data were sampled every second, and each grid cell in the plot represents how many of these sampled points fall within a given range of steering angles and speeds. The color intensity within each cell reflects the frequency of sampled points in that range: warmer colors indicate higher frequencies (more sampled points).

According to the BPA, for the autonomous system to match human driver performance, it must be capable of operating the vehicle at speeds between $0\,\mathrm{m\,s^{-1}}$ and $3\,\mathrm{m\,s^{-1}}$ driving forward using the full articulation angle range from $-43\,\mathrm{deg}$ to $43\,\mathrm{deg}$. At speeds above $3\,\mathrm{m\,s^{-1}}$, the articulation angle range decreases progressively, falling within the range of $-10\,\mathrm{deg}$ to $10\,\mathrm{deg}$ by $5\,\mathrm{m\,s^{-1}}$. As shown in the figure, speeds exceeding $5\,\mathrm{m\,s^{-1}}$ are only achieved with minimal articulation angles.

With this brief introduction to the challenges and expectations of the autonomous approach, and with the help of the discussions held with industry partners and mining operators, the requirements of the final control approach can be summarized as follows:

1. The control approach should be suitable for the routine operation scenarios of an ADT, which include many stop-and-go situations, parking, navigation through narrow passages, and sharp turns.
2. The control approach should also be able to drive backwards into narrow passages and navigate through them by using a vehicle model with the reference on the rear axle.
3. The control approach should be self-contained and find both the optimal articulation and speed command, so that the vehicle drives slowly when required and faster when possible to increase accuracy and efficiency.
4. The control approach must accommodate different ADT sizes.

To evaluate the successful fulfillment of these requirements, two metrics are considered. The first is the maximum lateral tracking error, which is important when navigating through narrow passages such as a crusher, passing obstacles, or reaching the end of a trajectory, i.e. the destination. However, this metric is not as important when traveling at higher speeds on the open, unmarked, and unpaved ground of a mine, as there are no obstacles to avoid, and ground irregularities cause instances of high lateral errors. In this case, the key metric is not the lateral tracking error itself but the mean absolute lateral error (MAE) and the vehicle's consistent and predictable behavior. Specifically, the MPC should achieve a maximum lateral tracking error of 30 cm for the c-ADT and 50 cm for f-ADT in narrow passages and an overall MAE less than 40 cm and 60 cm for the c-ADT and f-ADT, respectively. Exceeding these requirements poses risks to other vehicles, the physical surroundings, and the infrastructure of the testing mines, as these limits are

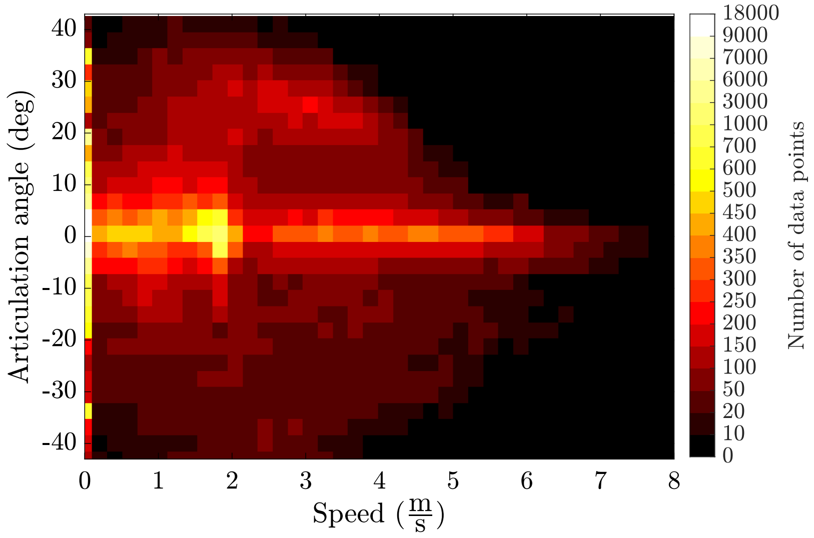


Fig. 1.5: Visualization of the BPA. Adapted from [M2] ©2022 IEEE.

defined with consideration for the physical constraints of the mining fields. Furthermore, while the speed in the state-of-the-art methods was limited to $4\,\mathrm{m\,s^{-1}}$, according to the BPA, it is required that the control concepts can achieve these accuracies while reaching over $4\,\mathrm{m\,s^{-1}}$ in the f-ADT on open grounds.

1.2.4 Control Concepts: State of the Art

The following presents the state of the art, highlighting its shortcomings. The challenges and critical aspects on the path to realizing autonomous ADTs are addressed one by one, and where applicable, existing approaches are reviewed. Each aspect is mentioned here for brevity, with further detailed discussions provided throughout the rest of this work. The gap between the literature and the requirements is presented at the end of this Section.

Autonomous articulated vehicles are not a new area of science, and there have been efforts as old as [79] in 1993 to create a simple PID control concept to steer the vehicle autonomously while driving at a constant speed and to keep it at a reference distance from the walls of an underground mine. With the rise of localization methods, the idea became more general and the goal was to keep the vehicle on a reference path. Therefore, models of articulated vehicles were presented that had a state describing distance error to a reference path. For example, in [5], an ADT is modeled in Frenet coordinates where the distance from a reference path is a state for both the front and the rear axle of the vehicle. The benefit of modeling in Frenet coordinates compared to modeling in inertial coordinates, is that it decouples the lateral and the longitudinal movement of the vehicle. As a result, the control task can be divided into two: lateral and longitudinal movement. This is of considerable benefit when using classical control theory for the path-following task since it poses two relatively simple control tasks and not a single complex one. In the mentioned work, the model is linearized, and a state feedback control is developed for the path-following. Another type of modeling of the vehicle with the distance error as a state is using error dynamics modeling, which is still a kinematic model not to be mistaken with a dynamic model. In [24, 25, 39, 66, 70], vehicle error dynamics modeling is presented and is followed by state feedback or feedback linearization control. Other control concepts such as fuzzy control or linear quadratic regulator (LQR) have also been employed on error dynamic models in works such as [3, 55, 83].

While the mentioned methods are relatively easy to implement and require less computation power than the model-based methods, their performance and applicability are limited. This is evident in the speed of the vehicles, which is limited to a maximum of $4\,\mathrm{m\,s^{-1}}$ in simulations and to $2\,\mathrm{m\,s^{-1}}$ or $3\,\mathrm{m\,s^{-1}}$ in experiments, and in the testing scenarios, which are low-dynamic paths and have limited or no changes in their directions. This setup is inadequate especially when driving on the surface, where the controller must be able to drive at higher speeds and more challenging turns with higher curvatures, as opposed to the simpler conditions of straight tunnels, so that the automation can match human drivers.

The underlying cause for the limitations in these approaches can be attributed to two key factors. The first factor is the non-predictive nature of these controllers. This means that these controllers react to a curve in the path only after it has already caused a distance error. The other problem is that the considerable transport and phase delays of the vehicle are not considered. Neglecting dead time and phase delay amounts to assuming that the control outputs are realized by the system actuators instantaneously. While this problem does not pose an issue when dead time and phase delay are negligible compared to the dynamics of the system and the reference, their large values in ADTs can cause oscillations around the reference path. For example, if the ADT is on the right side of a straight reference path, the controller will steer the vehicle towards the path to minimize the lateral deviation. As the vehicle approaches the path, the controller will reduce the steering angle so that the vehicle maintains the straight reference path. However, as the controller's outputs take time to be realized by the system, the vehicle may traverse to the left side of the path during dead time and phase delay. At lower speeds, this oscillation may gradually decay over time; however, as speed increases, the oscillation amplifies with each pass, resulting in an unstable system.

As a result, model predictive control (MPC) is proposed due to its ability to integrate a predictive aspect and to consider the noticeable delays and constraints of ADTs in steering and speed.

In [7], a nonlinear MPC (NMPC), and in [9, 88], linear MPCs are implemented for path-following, showing promising results at low speeds and in steady-state trajectories, such as circles. Notably, the simulation environment in these studies uses the same model as the controller, which likely contributes to the results. However, these methods do not account for phase delay and dead times, which may contribute to oscillations in the driven paths and limit achievable speeds in practical scenarios. The successful simulations with circular reference paths likely reflect the stability of steady-state conditions, where the effect of delays is less pronounced.

For non-constant speeds, the existing path-following controllers do not apply, because path-following controllers in general find only the optimal articulation angle by assuming a predefined and constant speed. When it is desired that the vehicle does not drive with a constant speed, a secondary approach should be employed that controls the speed in parallel. One approach involves establishing speed profiles for different sections of the path based on the intensity of the curvatures. This means to instruct lower reference speeds to a curve, the more intense its curvatures are. This can improve the results by instructing lower speeds at sharp curves. An example of this implementation can be found in [59]. Another method is presented in [9], where the optimization problem of the MPC is solved for multiple constant speeds in parallel, and the control policy with the lowest cost function value is selected. The goal of the cost functions is to minimize the distance to a reference trajectory while maintaining proximity to a set speed. The MPC chosen in each decision cycle is the MPC that can best minimize the distance to the reference trajectory while also driving at a high speed. The state weights in the cost function regulate the importance of each objective. However, this introduces new problems and challenges as the speed set point must be adjusted for each new path, and the proposed approach is not fully self-contained.

The issue of delays persists in both proposed approaches, and neither method achieves the optimal speed for a given reference track. While these methods broaden the application of path-following controllers, a path-following controller remains most practical when a constant speed is desired. As a result, path-following controllers are generally unsuitable for scenarios that require a dynamic or frequently changing speed profile. In application, this means that path-following controllers are not suitable for scenarios where a variety of driving scenarios such as parking or stop-and-go situations are involved. For these scenarios, trajectory-following MPCs are required, where the controller is self-contained and finds the optimal articulation and speed command for a given trajectory simultaneously. It is however important to mention that creating a suitable trajectory for a trajectory-following MPC requires computation power and effort to implement. This is due to the substantial difference between a path and a trajectory and the inclusion of time in the latter. This is a major advantage for path-following MPCs that makes them easier to implement and use.

Using trajectory-following MPCs for ADTs is limited in the state of the art. A trajectory-following MPC is implemented in [88] and compared to a path-following one, where both controllers are limited to $2\,\mathrm{m\,s^{-1}}$. A trajectory-following MPC is also introduced in [43], where the vehicle's phase delays are also included in the model[1]. However, dead times still remain an issue in these works, which explains the limited speed profile.

Another important aspect of the control task is the consideration of the sideslip angle, which is the difference between the orientation of the vehicle's front body or the yaw angle of the front body and the orientation of the true movement of the vehicle. Sideslip angle is explained in detail in Chapter 3. While ADTs are drift-less vehicles due to their all-wheel drive and special steering leading to the sideslip being limited [4], sideslip angle grows at higher speeds in curves. This means that sideslip plays an important role in the vehicle's movement at higher speeds. Since kinematic models cannot inherently consider sideslip in their equations, using kinematic models in their original form in the control means neglecting the sideslip angle. Therefore, in works such as [57, 58, 96] a kinematic model is augmented with the sideslip angles as parameters in its equations. Consequently, a path-following MPC is developed for these models. Since the inclusion of the sideslip angle in the kinematic nature of these models is only as a parameter and

---

[1] This paper introduces an ADT model with phase delays. This paper was published after publications [M3] and [M2], where, as far as the author knows, the delay was introduced to an ADT model for the first time.

not as a state, the MPC cannot consider the effect of the control variables on the value of the sideslip angles. Furthermore, In these works, only the current sideslip angle is added as a parameter to the MPC matrices, which raises two issues: firstly, the sideslip angle is assumed to be zero or constant for the rest of the prediction horizon. Secondly and more importantly, by doing so, the fact that the sideslip angle after the dead time might be different is ignored. These issues are not addressed in these works. Additionally, the testing is limited to simulations on circular tracks, where the constant sideslip inherent to the track prevents these problems from becoming apparent.

Another important aspect of autonomous driving for ADTs (and all types of front-steered vehicles) is driving backwards, as a crucial part of the routine work cycle of ADTs is to reverse into narrow passages, such as a crusher. However, the position of the front axle of these types of vehicles shows a non-minimum phase behavior in terms of the direction of its movement when driving backwards. Specifically, the front axle initially moves in the opposite direction of the steering before moving in the intended direction. This issue is explained in detail in Chapter 5.

This phenomenon poses challenges even with an MPC with a sufficiently long prediction horizon, or even an NMPC. The delay caused by the initial movement in the wrong direction means that by the time the vehicle starts moving in the intended direction, linearization and integration errors within the MPC's prediction can accumulate. This not only degrades the accuracy of the results but, in some cases, can lead to infeasible control solutions, as simulations and experimental tests prove in Chapter 5.

In [78], this problem is resolved by replacing the front axle's equations in the MPC with the ones of the rear axle. The rest of the equations, however, remain the same and are not adjusted to the new reference point. Similar to the rest of the state-of-the-art methods, the system delays are not considered and the testing speed is limited to $2\,\mathrm{m\,s^{-1}}$.

For compact ADTs (c-ADT), which have faster steering dynamics, no specialized models or controllers exist. Current ADT models are designed for the slow articulation process of full-sized ADTs (f-ADT), where the articulation angle is the integral value of the physically constrained articulation angle rate. In c-ADTs, however, this integral is not required. To take advantage of this faster steering, a specialized model for the c-ADTs should be presented. This problem is currently not addressed in the state of the art.

Given limitations in the state of the art, the gap between current approaches and the requirements outlined at the beginning of this Section are summarized as follows. Most existing methods focus primarily on path-following controllers, where the controller only provides a reference articulation angle and assumes a fixed, predefined speed. However, to meet the requirements, the controller must independently manage both articulation angle and speed, allowing it to reduce speed in curves or narrow areas to minimize lateral deviation from the reference trajectory. Additionally, current approaches have given limited attention to backward driving and often overlook dead times and phase delays in vehicle actuation. Lastly, existing studies mainly address f-ADTs, which have slower steering dynamics, and do not provide a dedicated approach for c-ADTs, which have faster steering dynamics. These limitations require developing new ADT models and controllers to address these challenges.

1.2.5 State Estimation Methods in ADTs: Requirements

As mentioned in the previous Section, vehicle models and respective MPCs will be developed in the course of this work that consider the sideslip angle of the vehicle and try to compensate for it. Therefore, the value of the sideslip angle is crucial for these MPCs at all times. A global navigation satellite system (GNSS) is used in this project to estimate the pose of the vehicle, and the pose can be used to calculate the sideslip angle directly [13]. However, GNSS data and the resulting pose estimation lose accuracy in mining fields since the direct line of sight to satellites is often lost or shadowed by the knolls in the mine especially when the mining surface is deeper within an excavated area [12]. Another possible method to directly measure the sideslip angle is to use optical sensors. These sensors are expensive and are not suitable for the harsh environment conditions encountered in mining environments, including dust, mud, and debris. As a result, indirect estimation methods are interesting for calculating the sideslip angle.

The goal of the estimation in this work is to provide an accurate sideslip angle for the controller. For this specific application, it is crucial that the estimated sideslip angle has minimal phase delay to ensure that the value used by the controller remains current. Additionally, simulations indicate that excessive error in the estimated sideslip angle can negatively impact controller performance and lead to instability in control results.

1.2.6 State Estimation Methods in ADTs: State of the Art

There is a limited number of efforts in the state of the art for ADT state estimation. In [74], a kinematic-slip model, which is similar to the models used for the MPCs that also consider sideslip angle, is used in an extended Kalman filter (EKF) for sideslip angle estimation. As also mentioned earlier, due to the kinematic nature of these equations, the inclusion of the sideslip angle is solely as parameters. Therefore, for the estimation of the sideslip angle in [74], the model is expanded to include the sideslip angle as a state that is modeled as random walk. With the same modeling, the radius of the wheels is also added as a state to the model. While using random walk in estimation is the common practice for parameters or states with a slow dynamic, the estimated sideslip with random walk cannot keep up with the very dynamic and fast-changing sideslip angles and, as will be shown in this work in Chapter 6, experiences considerable phase delays and poor accuracy. Therefore, it is recommended to use model knowledge for such states instead of relying on the random walk model.

In contrast to ADTs, state and parameter estimation for conventional front Ackermann-steered vehicles is studied extensively and comprehensively in the state of the art. These efforts fall into two categories for the model that is used in the estimation filter: The first category uses kinematic point mass systems while the second category is based on dynamic vehicle models. The specific use case for the filters is the determining factor in deciding which category to use. The first category offers the advantage that the estimation does not rely on the model parameters as heavily as the second one, which is due to the relatively smaller number of parameters, which also remain constant during driving. This helps the estimation to robustly deal with model uncertainties and makes the setup process simpler. However, the second category offers the possibility to employ different sensors in the estimation, which can increase the quality of the estimation and the number of states and parameters that can be estimated. In the following, the state of the art in utilizing these two types of models for estimation in Ackermann-steered vehicles is presented, with a focus on the most prominent approaches.

In [20], [46], and [77], the point mass kinematic model is employed to estimate the sideslip angle. In the first two works, this is done via an EKF, and in the last work via a heuristic scheduling algorithm. An advantage of methods using point mass models is that they do not require the position of the vehicle to be known since the vehicle position is not a state of the system. However, IMU data is used as input in all of these works.

Turning to the second category, a two-track dynamic vehicle model is used in [28], where the model employs suspension deflections as input. As mentioned earlier, using dynamic models in estimation allows different inputs to be utilized for different use-cases and sensor setups. A Kalman filter then estimates the tire normal forces, which are used together with inertial measurement unit (IMU) data in another EKF and an unscented Kalman filter (UKF) to find the sideslip angle. Another setup of a two-track dynamic model is used in [6], where the inputs to the model are the wheel rotational speed, IMU data, and propulsion and braking torque. An EKF and a UKF use the model and are compared in performance. A similar setup of a two-track dynamic model is used in [93] in a dual extended Kalman filter (DEKF) for state and parameter estimation. The estimated parameters are the moment of inertia, mass, and the position of the center of mass.

As mentioned earlier, the pose of the front body of the ADT is available in this project. Therefore, the methods that are based on a kinematic point mass, which do not rely on the current position of the vehicle, do not have an advantage compared to the other approaches and are not studied further. Furthermore, the state-of-the-art methods that utilize dynamic vehicle models in the estimation require information such as wheel rotational speed, tire normal forces, etc. that are not available on the ADT in this project and many commercial ADTs. This issue and the challenges associated with this topic are discussed in detail in Chapter 6. Furthermore, at the time this project started, there was no dynamic model of an ADT available that could be employed in an estimation algorithm.

Given the limitations in the state-of-the-art methods, at first, a dynamic model for an ADT needs to be developed. This model has to first fulfill the requirements defined in 1.2.2. Then, the model needs to be simplified and adapted for the sensor setups of commercial ADTs, which include the ADTs of this work as well. This model should be employed in an EKF for state and parameter estimation. For use cases where it is desired to have a simple and fast solution to the state estimation problem for the sideslip angle, a kinematic model and the respective Kalman filter need to be developed. This kinematic model, however, in contrast to the state-of-the-art methods, should utilize model knowledge in the equation of the sideslip angle by using a differential equation for describing the sideslip angle.

1.2.7 Research Objectives and Questions

For autonomous fleets of ADTs to be actually applicable, it is important that they are able to complete a transportation cycle. Considering the shortcoming in the state-of-the-art methods, and considering the requirement that were defined for a suitable development chain to achieve the controller, the following questions need to be answered or addressed in this work:

1. How precisely can a dynamic model of an ADT be developed that accounts for the effects of steering cylinders on overall system dynamics without explicitly modeling them?
2. How well can an MPC control the vehicle?
3. What estimation models can be implemented to accurately determine the sideslip angle in ADTs, and how do these techniques compare in terms of phase delay and accuracy?
4. What is the impact of incorporating the sideslip angle on the vehicle's control performance?

In order for these questions to be answered in the course of this work, the structure of this thesis is designed as follows and as illustrated in Figure 1.6. In Chapter 2, the hardware setup of the ADTs is presented. This includes both an introduction to ADTs in general and an overview of the ADTs in this project with their sensors and actuators. Furthermore, the computation platforms, where the developed algorithms run in real time for the experiments, are presented. In Chapter 3, a digital twin developed for both ADTs of this work is presented. This digital twin contains different simulations of the

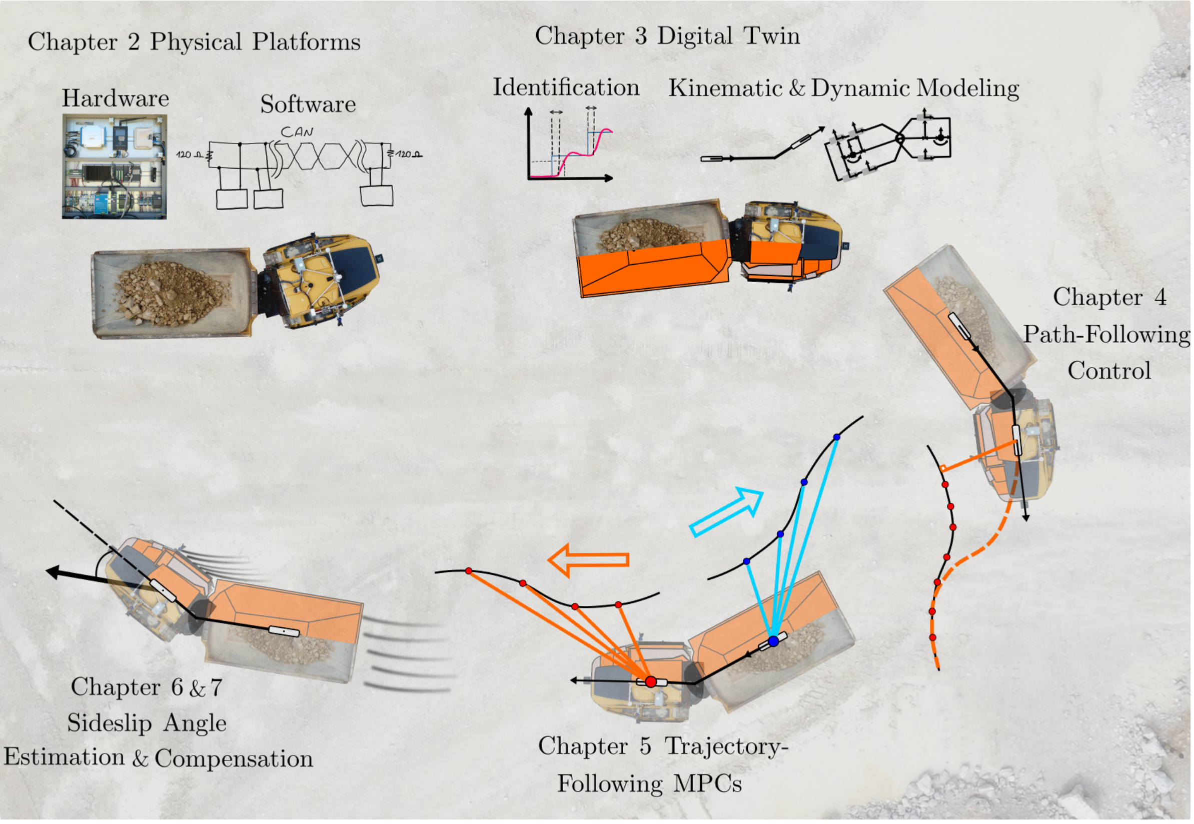


Fig. 1.6: Sketch of the Chapters of this work with their content.

vehicles and the interfaces between a controller and other modules of the system so that software in the loop and hardware in the loop can be achieved. In this Chapter, the first research question is addressed. Chapter 4 presents a classic feedback control method and a path-following MPC so that a direct comparison between model-based and classic control methods is achieved. The path-following controller will also later be used in comparison with trajectory-following controllers. In Chapter 5, trajectory-following controllers for a routine operation of the ADTs, which includes driving forward and backwards, will be presented. In this Chapter, the experimental results are also presented and discussed. The content of this Chapter contribute to answering the second research question. Chapter 6 presents the state and parameter estimation for these vehicles by presenting a set of estimation techniques that employ different vehicle models. By doing so, this Chapter addresses the third research question. Chapter 7 presents MPCs that utilize the sideslip angle and try to compensate for it in the control process. This Chapter discusses the fourth research question. Finally, in Chapter 8, the summary and outlook of this work are presented, and it is discussed, what the shortcomings of the presented methods are and how the results can be improved in future research.

2 Physical Platforms

This Chapter introduces the hardware used in this project. It begins with an introduction to ADTs in general, followed by a detailed technical explanation of the c-ADT and f-ADT of this work. This discussion includes the sensors and actuators available in these vehicles, their communication interfaces, and protocols. Furthermore, the modules responsible for executing the developed algorithms in real-time and on the physical vehicles are introduced.

2.1 Introduction to Articulated Dump Trucks

Articulated dump trucks (ADT) also known as articulated haulers are a class of large, heavy, and heavy-duty dump trucks designed for transporting heavy or massive and bulky loads over rough unpaved terrain. These vehicles consist of a front and a rear body connected at the articulation point or hinge. In contrast to Ackermann-steered vehicles, in ADTs, the front wheels cannot change their angle, and the steering is achieved by changing the orientation between the front and rear body, which is called the articulation angle. This is done by the steering cylinders that are mounted to the sides of the hinge and can create the necessary torque for articulation.

The purpose of these steering cylinders is twofold. As mentioned briefly in Chapter 1.2, they are responsible for steering the vehicle when a steering command is given. Secondly, in the absence of a steering command (in other words, when the steering commands the articulation angle to be constant), the cylinders are responsible for maintaining a constant articulation angle, making the vehicle behave similarly to one rigid body. This specific steering mechanism gives ADTs a high level of maneuverability even when they are transporting heavy loads. Figure 1.2 illustrates a bird's eye view of an ADT with its cylinders.

This high maneuverability, however, comes at a cost. When there is no steering command, the cylinders have closed valves and keep the articulation angle constant. When there is a steering command, oil pressure has to reach a certain level before the vehicle can actually articulate. This leads to considerable phase delay and dead times in the steering mechanism of ADTs.

To enable carrying their own and the load's heavy weight, these vehicles are equipped with powerful engines. While the engine provides the ADT with the required torque and power to carry out its tasks, it introduces phase delay and dead time to the longitudinal movement due to its mechanism, which is common for internal combustion engines [85].

ADTs are categorized into multiple classes depending on application and size. These are for example underground mines, surface mines, construction sites, etc. Figure 1.3 shows a full-sized and a compact ADT (f-ADT and c-ADT). In recent years, c-ADTs have gained popularity since they are suitable for urban construction sites where heavy loads need to be transported but there is not enough space for bigger transport vehicles.

The following Sections discuss the sensors and actuators that were available on the ADTs and those that were added for the project. The remainder of this Chapter is presented in two categories: the f-ADT and the c-ADT.

2.2 Compact ADT

2.2.1 Vehicle Introduction

The c-ADT of this project is a Wacker Neuson 1501, shown in Figure 2.1. Table 2.1 presents the vehicle dimensions. This vehicle weighs 1320 kg, can carry up to 1500 kg, has a diesel motor with 18.9 kW (25.3 hp), and benefits from all-wheel drive. The maximum allowed speed by the manufacturer is $4.5\,\mathrm{m\,s^{-1}}$. The ADT is a commercially available vehicle, originally designed for a human driver, featuring a physical steering wheel and a

Fig. 2.1: c-ADT after the rebuild.

Table 2.1: Dimensions of the c-ADT.

Parameter	**Value**
Vehicle total length	3.3 m
Front axle to hinge	0.805 m
Rear axle to hinge	0.845 m
Vehicle width	1.305 m
Vehicle track width	1.07 m

driver's seat. To adapt the c-ADT for autonomous driving, it had to be modified and equipped with necessary sensors and actuators, enabling the sending of reference signals to vehicle actuators and the receiving of the current vehicle states from the sensors. Fritz Rensmann GmbH was tasked with this modification process, which involved removing the seat and steering wheel to create space for additional components. The rebuild also included adding an industrial personal computer (IPC) to the vehicle, which functioned as the driver for the vehicle actuation and sensors. The vehicle IPC is connected to the communication interface, allowing the sensor data to be transmitted and reference values for the actuators to be set via the same interface. The details of the communication interface will be discussed in Section 2.2.4.

In the following, the hardware and software components of the vehicle after the rebuild are presented.

2.2.2 Sensors and Actuators

The c-ADT, following its rebuild, was equipped with various actuators and sensors. On the actuation side, it was possible to send reference values for both the articulation angle ϕ_{ref} and engine speed $N_{\text{eng,ref}}$. One key requirement for the ADT platform according to the project design was to also include the reference speed as an input. However, after the rebuild, it became evident that the vehicle was not able to track a given speed and showed unpredictable behavior. To better understand this issue, the mechanism of the new actuation of the vehicle must be explained.

The vehicle uses a motor that drives a pump, which generates hydraulic pressure. The high-pressure oil then flows to the all-wheel drive system and rotates the wheels so that the vehicle moves. This design is useful for the compact size of the vehicle since it removes the need for a traditional gearbox and differential leading to a high-power and low-weight vehicle. In order to add autonomous driving to the vehicle, a linear motor was added to the vehicle by a project partner, which replaced the gas pedal. Unfortunately, the linear motor did not have a feedback option and its position was unknown at all times other than when reaching its zero or maximum length position. Furthermore, it was not possible to close the feedback loop using the current motor RPM or vehicle speed since a change in the direction of the linear motor caused an unpredictable and additional dead time of up to 2.5 s. Therefore, implementing a speed controller for the vehicle was not possible.

The actuation behavior of the vehicle regarding articulation angle ϕ_{ref} and engine speed $N_{\mathrm{eng,ref}}$ will be explained in the following Chapter, Section 3.3. Figure 2.2 shows an overview of the signal flow in the c-ADT.

A further available actuator was the break percentage. However, due to the vehicle's special hydraulic drivetrain, unlike in normal passenger vehicles, the vehicle would stop as soon as the engine speed N_{eng} dropped to idle (at 1100 rpm). An increase in engine speed would cause the vehicle to move. As a result, active braking via the vehicle's brake system was generally unnecessary, as the vehicle's speed could be controlled solely by adjusting the engine speed. Active braking was only required in emergencies when a rapid stop was needed. The brake system of the vehicle did not provide feedback about the current state of the brake.

On the sensor side, the engine speed (N_{eng} with a resolution of 100 rpm) and the vehicle's articulation angle (ϕ with a resolution of 2 deg) were available. The Manufacturer does not provide information about the accuracy of the sensors.

Furthermore, the pose of the vehicle was provided by the localization module. The localization module, which is one of the work packages of this project according to the introduction in Chapter 1.1, was implemented for the c-ADT by a project partner also at IRT. This module is documented in [14]. The localization module provided the vehicle's position in inertial coordinates (x_1, y_1), its speed v_{x_1}, v_{y_1}, which are the longitudinal and lateral elements of the velocity vector at the front axle, and the yaw angle ψ_1 with an error standard deviation of 0.1 m, $0.1\,\mathrm{m\,s^{-1}}$, and $3{\cdot}10^{-4}$ rad, respectively [14].

Figure 2.2 illustrates the structure of the actuator and sensor signals and the vehicle's control loops. Since the brake system of the vehicle did not provide feedback about the current state of the brake, it is illustrated as an open-loop control.

2.2.3 Computation Platform

The IRT's computation platform for the c-ADT in this project is a Bachmann programmable logic controller (PLC), which is also an IPC. This module will be referred to as the IRT IPC in this work. The IRT IPC was chosen due to its real-time capabilities, which guarantee the execution of the algorithms with a defined rate. This unit is responsible for running the software framework in real time. The IRT IPC consists of the following elements:

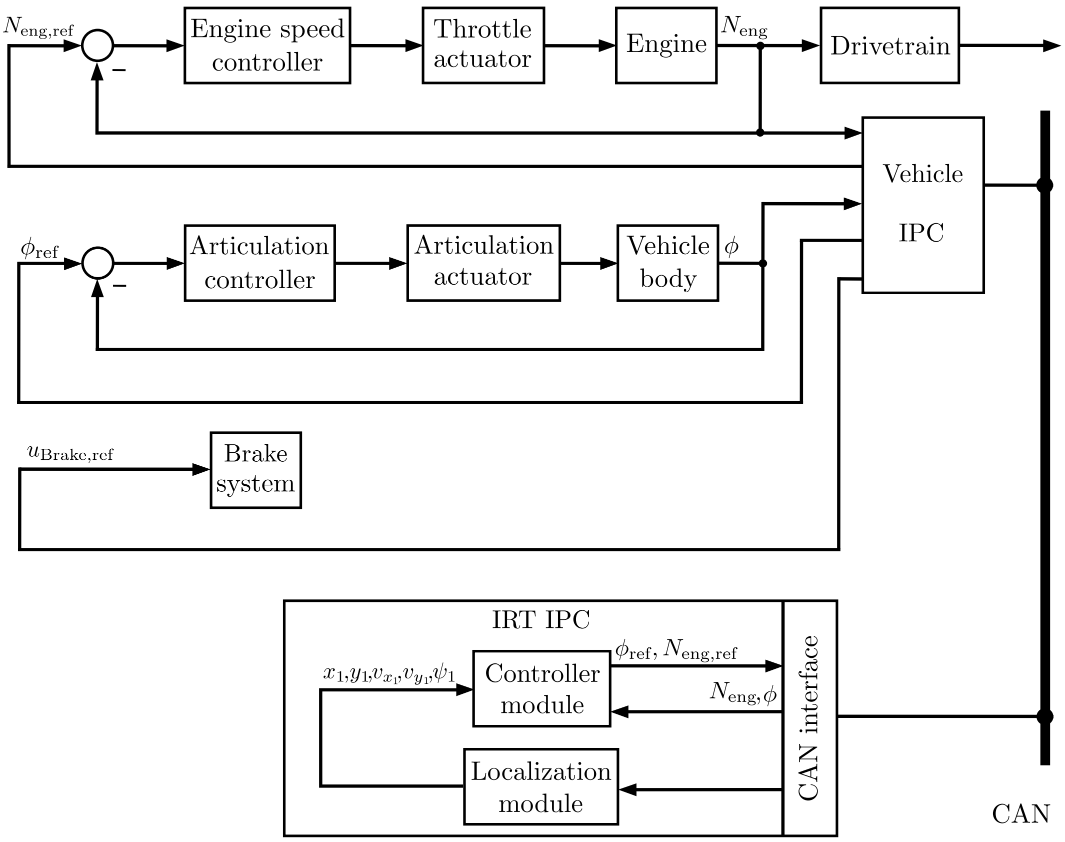


Fig. 2.2: Structure of the signal flow in the c-ADT after the rebuild.

- Processor module (MH230): The processor module is a dual-core processor with 2.3 GHz.
- Digital input/output (IO) unit (DIO232): This module has 32 channels that can be programmed as digital inputs or outputs.
- Controller area network (CAN) Module (CM202): This module is responsible for communications via CAN and can manage up to two different networks independently.

The Bachmann PLC can run C++ code in real-time and also offers the possibility to compile MATLAB/Simulink code directly from the MATLAB environment on it.

2.2.4 Software Framework and Communication Interface

The software framework inside the IRT IPC was given the task of running the localization and control modules and also the communication interface. The communication interface for this vehicle was designed in cooperation with the Fachhochschule Aachen (FH Aachen) and was designed as a controller area network (CAN) bus as is common in the automobile industry. Using CAN bus for communication reduces wiring, enables distributed control, and increases reliability for sending and receiving messages [89]. The CAN bus had the task of transferring sensor and actuator signals and also signals regarding safety and security features between the vehicle IPC and the IRT IPC.

For the safety aspect, two features were developed and implemented as a part of the software framework. The first aspect was a heartbeat signal, which ensured that the IPC in charge of driving the vehicle was operating properly and sending deliberate signals to the vehicle's IPC via the CAN bus. The heartbeat functioned by sending a series of expected counting numbers together with reference actuation signals to the vehicle's IPC as a sign of the IPC in charge being still alive. The vehicle's IPC would activate an emergency stop if the heartbeat were interrupted or mistaken.

The second safety feature was realized by a signal from a remote dead man's switch connected directly to the vehicle's IPC. This switch had to be in the vicinity of the vehicle at all times.

The collaborative tasks of the controller and localization modules and the CAN bus can be summarized in the following:

- CAN interface, bus management: Receive CAN messages and save them inside a memory unit.
- CAN interface, decoding module: Decode the received CAN message using a CAN database (DBC) file into a decimal number and save it inside another memory unit.
- Controller module: Run the algorithms such as the controller, the state estimation, and the planner, and save the control signals into a memory unit. Furthermore, this module was given the task of preparing the heartbeat signal. Therefore, at each successful iteration of the controller, the heartbeat signal should be also sent out.
- Localization module: Run the localization algorithms. More information about this module can be found in [13].
- CAN interface, encoding module: Encode the control signals into can messages using the DBC file.
- CAN interface, bus management: Send the CAN messages to the sending queue of the IRT IPC.

The DBC file, which is used for decoding and encoding CAN messages and is responsible for allocating CAN memory and priority to each signal, was developed by the author and the partners from FH Aachen.

When designing the software architecture, it is also important to assign a running rate to each software module that guarantees the functionality of that module and aligns with the computation power of the computer. In order to avoid a CAN message overflow when receiving the CAN messages in the receive queue, the bus management runs with 2000 Hz. In order to ensure that the data used by the controller and localization modules are not outdated, it was decided that the decoding module runs at 1000 Hz, adding only 1 ms to the latency of the incoming signal. The controller module runs at 20 Hz. The encoding module runs at 1000 Hz and the sending module at 2000 Hz, similar to their receiving counterparts. In order to avoid over-trafficking the CAN bus, the sending module only sends a message to the network when there is a change in the value of the heartbeat indicating the existence of a new control message. Figure 2.3 presents a graphical overview of the software framework in the Bachmann PLC.

The dead time introduced by this software framewrok is 50 ms, which complies with the fact that the controller module runs with 20 Hz. The dead time of the entire actuation cycle is discussed in the following Chapter, in Section 3.3.

2.2.5 Hardware Integration on the Vehicle

For the hardware integration on the vehicle, it was decided to use a switch box, where the hardware components could be installed. The chosen switch box had to fulfill the following requirements:

- House the power electronics required for the other modules,
- house the IRT IPC,
- house the localization module computers and the router,

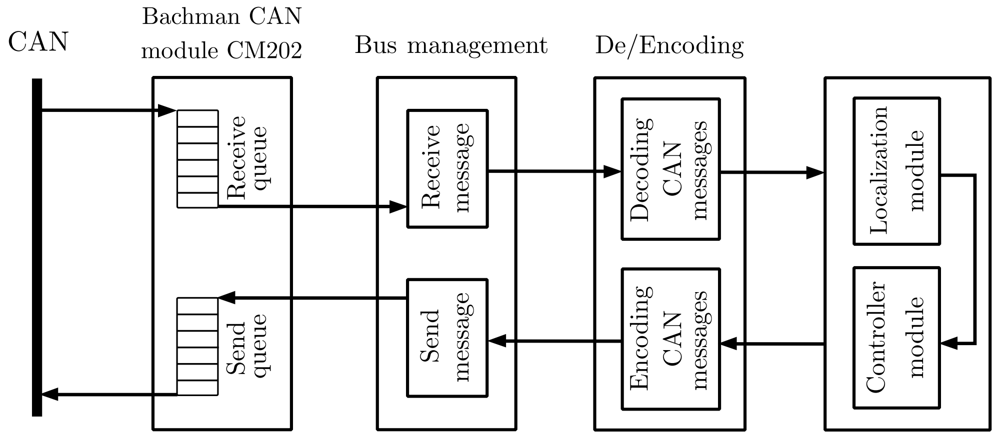


Fig. 2.3: c-ADT software framework inside the IRT IPC.

- be able to carry the weight of the mentioned components since it was to be mounted vertically,
- have sufficient room for air circulation after the components were installed.

For these requirements, a switch box from the manufacturer Rital was chosen with the dimensions 760×760×300 mm. The switch box and the elements inside weigh approximately 50 kg. For mounting the switch box on the vehicle, damping elements had to be added to the contact points to reduce the shocks given to the switch box and consequently to the equipment when driving on the unpaved and rough grounds in a mine. For this purpose, four damper elements from the manufacturer CAVOFLEX were chosen. Figure 2.4 shows the mounted switch box and the elements inside. Different elements of the switch box are marked with letters A, B, and C. Box A marks the hardware that is required for the navigation and localization module, and box C marks the power electronics. The Bachmann PLC is marked with B.

2.3 Full-Sized ADT

2.3.1 Vehicle Introduction

The f-ADT in this project is a Bell B30E, which weighs over 22 t and can carry up to 28 t. This vehicle has a diesel motor with 260 kW (348 hp) and an all-wheel drive. Similar

Fig. 2.4: c-ADT and the switch box. Box A marks the modules that are parts of the navigation and localization system. Box B marks the Bachmann PLC. Box C marks the power electronics. Box D marks the GNSS antennas.

to the c-ADT, this vehicle is also commercial and is available for purchase. However, unlike the c-ADT, this vehicle comes equipped with a CAN interface and does not require additional sensors and actuators for automation. Table 2.2 lists the dimensions of the vehicle.

2.3.2 Sensors and Actuators

Figure 2.5 illustrates the structure of the signal flow in the f-ADT. On this vehicle, the articulation angle and the wheel-based speed were available on the CAN. Thanks to the implemented localization algorithm by Indurad GmbH, the position of the vehicle, its yaw angle, and GNSS-based speed were also available, utilizing real-time kinematic (RTK) corrections for enhanced accuracy. Neither of the manufacturers present the accuracy of the sensors. Figure 2.6 shows the f-ADT equipped with the GNSS antennas.

The vehicle was able to follow a given reference for the speed, articulation angle, and

Table 2.2: Dimensions of the f-ADT.

Parameter	**Value**
Vehicle total length	9.95 m
Front axle to hinge	1.36 m
Rear axle to hinge	4.48 m
Middle axle to hinge	2.81 m
Vehicle width	2.99 m
Vehicle track width	2.35 m

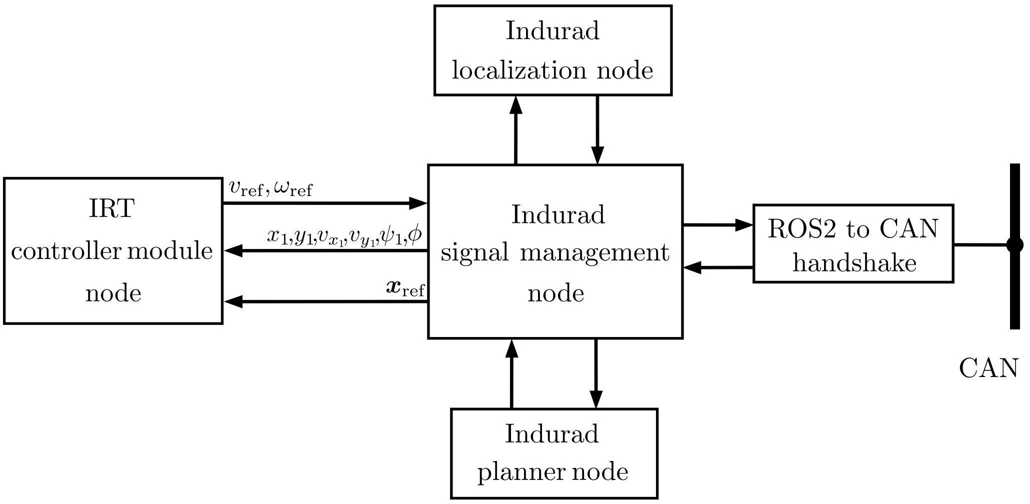


Fig. 2.5: Structure of the signal flow in the f-ADT.

Fig. 2.6: f-ADT in the mining field.

articulation angle rate. The actuation behavior of the vehicle regarding these variables will be explained in the following Chapter, Section 3.3.

The planner module for the f-ADT, which is one the work packages of this project according to Chapter 1.1, was developed externally by Indurad GmbH. As shown in Figure 2.5, the planner module communicates with the IRT node externally, providing the controller with the reference trajectory $\boldsymbol{x}_{\text{ref}}$.

2.3.3 Software Framework and Communication Interface

The f-ADT had an entirely different software framework than the c-ADT, which was developed externally by Indurad GmbH. The software framework was tasked with running the Indurad localization module, Indurad planner module, IRT controller module, and the communication with the vehicle's CAN bus. Due to safety and security reasons, the vehicle manufacturer has implemented a negotiation protocol on the CAN interface, so that only authorized parties can access the information on the CAN and send reference signals to the vehicle. Indurad GmbH developed the software framework in ROS2 and was authorized to communicate with the vehicle. As a result, the sensor information was available in the ROS2 environment as well. The control algorithms developed for this work were compiled and executed in this environment and no further development on the author's side was necessary.

The precise dead time introduced to the entire control loop by this framework cannot be measured, as it operates as a closed black box not developed in-house, unlike in the c-ADT. However, this does not pose an issue, since the overall system dead time remains measurable, which is discussed in the following Chapter, in Section 3.3.

2.4 Summary

In this Chapter, the hardware setup for the ADTs was presented, which included an introduction to ADTs in general and a more detailed discussion about the ADTs in this project. Furthermore, the setup of the sensors, actuators, and how they communicate were explained. In the next Chapter, the concept of the digital twin for the ADTs will be presented.

3 Digital Twin

3.1 Introduction to a Digital Twin

This Chapter presents a comprehensive overview of the steps involved in the development of a digital twin for the ADTs. Digital twin is in simpler terms a virtual mapping from the real world that allows the exploration of a complex system in a digital environment [15]. The development of a digital twin was of great importance in this project due to the limited, expensive, and time-consuming access to the ADTs. The digital twin is implemented both as software in the loop (SIL) and hardware in the loop (HIL).

In order for the digital twin to be practical for this project, certain requirements had to be met. Firstly, the different modules of the digital twin and how their interfaces communicate had to be identical to the real vehicles. A successful implementation with this requirement met would mean that the algorithms could be executed on real vehicles without any adaptation or change in their structure. Furthermore, the implementation had to be modular, so that it was possible to replace the existing modules with a plug-and-play concept when a newer and more accurate implementation of that module became available. In the following, the required modules of the digital to fulfill these requirements are presented.

3.2 Modules of the Digital Twin

3.2.1 Modules Architecture

In this project, the digital twin serves the purpose of replicating real-world setups and conditions specifically from the perspective of control algorithms. Through this perspective, four crucial modules interact with these algorithms: the planner, the sensor, the vehicle, and the state estimation module. The control algorithms are embodied by a fifth module: the controller module. This module is the module that is later compiled and used in real experiments. In the following, these modules are explained. Figure 3.1 illustrates an overview of the entire system. In this Figure, the modules are highlighted by thicker box lines.

3.2.2 Planner Module

The first module is the planner, which has the task of providing the controllers with a reference path or trajectory. As described in Chapter 1.1, the planner module is also a stand-alone work package of this project. The task of this module is to generate the reference trajectory while taking all the obstacles into consideration. The planner for the f-ADT was developed by Indurad GmbH and could only be accessed on-site and not beforehand. As a result, it had to be treated as a black box. Furthermore, the c-ADT did not have a dedicated planner.

Given these circumstances, particularly because the planner module of the f-ADT was only accessible on-site, the digital twin's planner module was designed to generate pre-recorded paths or trajectories rather than function as a full trajectory planner. It was also decided that a version of this planner should also be used on-site for the c-ADT as this vehicle lacks its own planner. This module is therefore implemented twice in the digital twin, as also illustrated in Figure 3.1. The functions of these implementations are as follows:

- The first implementation, located outside the controller module, corresponds to the planner of the f-ADT. In reality, this is an external module that communicates with the control algorithms.
- The second implementation, situated within the controller module, corresponds to the planner for the c-ADT. It is integrated into the control algorithms to ensure later inclusion in the compiled code.

In the digital twin, parameter P_1 determines, which planner is utilized for the controller.

For the pre-recorded trajectories, different data recording methods were employed depending on the specific objectives of each test. These included simple tracks such as a circle or a rectangle for proof-of-concept tests or more complex tracks such as sine curves with increasing frequency. Further details regarding each chosen path/trajectory, how they were created, and a discussion about their specifics are presented accordingly throughout this work prior to each test scenario.

3.2.3 Controller Module

The controller module of the digital twin corresponds to the controller module of the work packages presented in Chapter 1.1 and is the main goal of this work. This module in the digital twin primarily embodies all the control algorithms developed in this work. As mentioned in Section 3.2.1, this module will ultimately be compiled and deployed in both the ROS2 environment for the f-ADT and the CAN environment for the c-ADT. As a result, its implementation in the digital twin includes additional components so

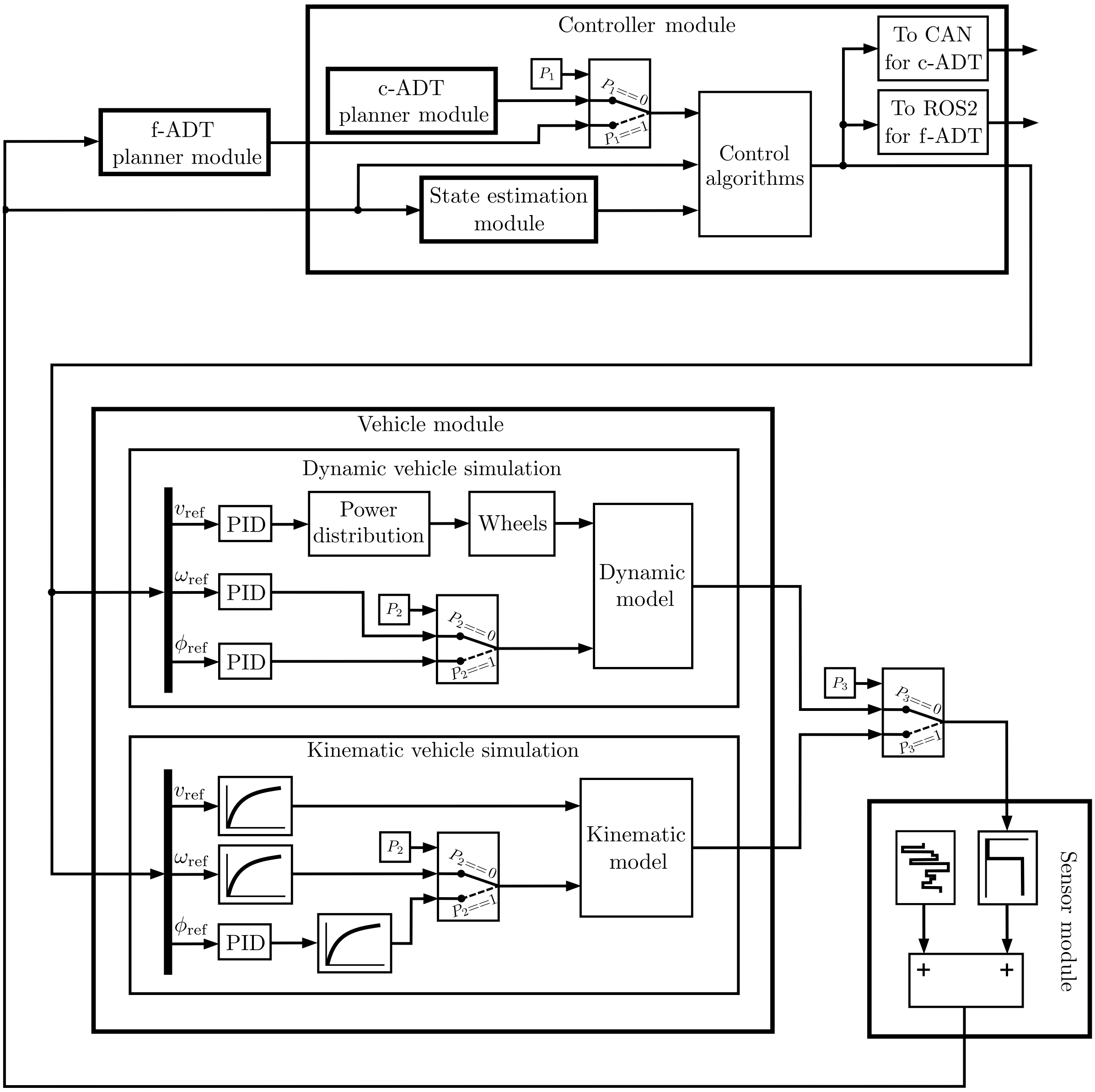


Fig. 3.1: Block diagram of the digital twin. The modules of the digital twin are marked with thicker boxes and are the controller module, the planner module, the state estimation module, the vehicle module, and the sensor module.

that it can function as a stand-alone plug-and-play unit. As illustrated in Figure 3.1, these components are the planner for the c-ADT, the interfaces needed to transmit the controller output to both the c-ADT and f-ADT, and the state estimation module.

3.2.4 Sensor Module

The objective of the sensor module is twofold. It should provide vehicle information in data types that match those of the real vehicles and send them to the controller. Furthermore, it should also add dead time to the signals so that the data flow in the simulation is similar to the real vehicles. For testing the state estimation algorithms in the digital twin, the sensor module needs also to provide the state estimation module with the required data. In this case, noise needs to be added to the signals so that the state estimation is tested under realistic conditions.

3.2.5 State Estimation Module

The next module is the state estimation module. This module receives the sensor information with added noise and sends filtered and estimated states to the controller.

3.2.6 Vehicle Module

The vehicle module is responsible for simulating the physical behavior of an ADT by receiving reference values for lateral and longitudinal movement from the controller $(\phi_{\text{ref}}, \omega_{\text{ref}}, v_{\text{ref}})$ and responding to them. The parameter P_2 in Figure 3.1 is symbolically responsible for switching between articulation angle ϕ_{ref} or rate ω_{ref} as the input to the system. To ensure that this simulation is close to the real vehicles in its physical dynamic behavior and is identical to them in the interface setup, multiple sub-modules need to be developed. As shown in Figure 3.1, these sub-modules are vehicle models with different levels of complexity and fidelity, low-level PID controllers, wheel models, and power distribution, which are explained below.

For different complexity and fidelity, kinematic and dynamic vehicle models are utilized in the vehicle model. In Figure 3.1, parameter P_3 is symbolically responsible for choosing between the dynamic or kinematic vehicle model for the simulation.

The low-level controllers and power distribution in this module are designed to ensure that the inputs of the real vehicles and the simulated vehicles in the vehicle module are identical. These controllers serve the purpose of bridging the gap between the mathematical model inputs and the real vehicle inputs. For the tuning of these controllers, identification tests for both lateral and longitudinal movement are designed and conducted on the real ADTs.

The rest of this Chapter presents the comprehensive development process of the vehicle module. To do so, it first begins with vehicle identification. This includes test design, real-world execution on the vehicles, and evaluation of the results. Following this, the vehicle models that are used in the simulation are presented, followed by the wheel models and power distribution. Afterwards, the implementation of the vehicle module and the digital twin in MATLAB/Simulink is presented, which is followed by the validation with real-world data. This concludes the SIL of this work. Finally, a HIL system developed for this project is also presented that is designed to test the compatibility of the controller module with the IRT IPC where it is executed.

3.3 Identification

3.3.1 Identification Objective

In order to display the necessity of identification for the simulation and for the controllers, the objective and function of identification must be discussed. As explained before in Section 2.1, ADTs' high maneuverability while transporting heavy loads is what distinguishes them and makes them better suited for their use cases. This high maneuverability is made possible by the special steering in these vehicles. However, as a result of the special steering and the ability to carry heavy loads, there is a dead time and a phase delay between receiving the steering command and realizing it. In the context of autonomous vehicles, phase delays and dead times are not exclusive to ADTs; however, their value is in this case considerable. A controller that functions properly on a delay-free system, might not function similarly well on a delayed system or might become infeasible. Therefore, it is crucial that these delays are present in the digital twin and are also considered in the controllers.

The objective of the identification process is to design, implement, and evaluate experimental tests that determine the vehicle's dead times and phase delays, and then fit mathematical models to these findings for integration into the controller. The ultimate goal of this Section is not to necessarily reach a perfect match between the identified function and the vehicle behavior but to represent the dead time and phase delay, which are the features that are important to the controller.

In this Section, the design of the identification tests, experimental tests on the vehicle, and evaluation and analysis of the results are presented. All the identification functions were obtained using the MATLAB identification toolbox utilizing the TFEST function with 20 iterations. The contents of this Section are based on the publication [M3].

3.3.2 Full-Sized ADT: Lateral Identification

The aim of this Section is to identify the lateral movement of the vehicle for the articulation angle rate ω by finding a transfer function describing $\frac{\omega}{\omega_{\mathrm{ref}}}$, where ω_{ref} is the reference value for the articulation angle rate. In order to identify the lateral movement of the f-ADT, a series of experiments were carried out on the vehicle. The aim of these experiments was to collect data on lateral movement in different operating scenarios, such as at a standstill, while driving with a range of higher and lower speed, and with or without a load. In these tests, different reference set-points were given to the vehicle in the form of step functions, and the vehicle response was recorded. The results of the experiments were used to fit a first-order low-pass element using MATLAB's Identification Toolbox.

These experiments were performed on the vehicle in four iterations. After each experiment, the results of the identification were discussed with the vehicle manufacturer (Bell), and the vehicle's software was adapted to make the lateral movement as consistent as possible in most operating scenarios. This consistency made it possible to use a single first-order low-pass element to describe the articulation behavior. The results of one of the final tests on the vehicle with the final version of the software are shown below. The following function has been fitted to the final tests of the vehicle:

$$\frac{\omega(s)}{\omega_{\mathrm{ref}}} = \frac{1}{Ts+1} e^{-T_t s}. \tag{3.1}$$

Here, s is the complex frequency variable or the Laplace variable, $T = 0.5\,\mathrm{s}$, and $T_t = 0.5\,\mathrm{s}$. As these are the fitting results, the identified model might not perform as well in all scenarios. In this specific test, the vehicle drives with $5\,\mathrm{km\,h^{-1}}$ and steers. Figure 3.2 shows the experiment and its results. The first steering command occurs at $t = 1.5\,\mathrm{s}$. After a dead time of 0.5 s, the vehicle reacts to the reference steering command. At $t = 5.8\,\mathrm{s}$, the vehicle reaches its end angle and cannot steer further, and the articulation angle rate goes back to zero, regardless of the reference value. A new articulation command is given to the vehicle at $t = 8.6\,\mathrm{s}$ in the opposite direction, and the vehicle shows similar behavior. At $t = 16.6\,\mathrm{s}$, the vehicle reaches its other end angle, and the articulation angle rate drops to zero before the reference value does. The fitted function describes the lateral movement of the vehicle and is consistent in the aforementioned operating scenarios.

A direct identification of the articulation angle with a first-order low-pass element is not possible due to the characteristics of the hydraulic articulation. In this steering mechanism, oil pressure has to accumulate in the presence of an articulation command, which causes an articulation angle rate. The maximum achievable articulation angle rate of the vehicle is 12 deg/s. Considering the full range of articulation, which is -43 deg to 43 deg, depending on the desired $\Delta\phi$, the time required to reach the reference articulation angle varies. This implies that representing the articulation angle itself with a first-order low-pass element is not possible. In other words, when given an articulation command, the articulation angle rate is built up and the integral of this rate yields the articulation angle. Therefore, two equations are required to describe the articulation angle itself: one is the equation for the articulation angle rate (3.1), and the other equation is the integral.

3.3.3 Full-Sized ADT: Longitudinal Identification

The aim of this identification procedure was to also find a function to describe the longitudinal vehicle behavior when given a reference speed. For this experiment, similar to the experiments for lateral identification, a step function was given to the vehicle as a reference and the speed was recorded. The vehicle is able to follow the reference speed by using the acceleration and brake systems. Similar to the lateral behavior, multiple tests were conducted on the vehicle in cooperation with the manufacturer to achieve consistency in the vehicle's longitudinal behavior. Figure 3.3 shows segments of the results of the final acceleration and deceleration tests.

The identification tests proved, that the vehicle's speed shows behaviors that cannot be modeled with first or second-order delay elements. Instances of this behavior are shown in Figure 3.3 at $t = 12\,\mathrm{s}$ while decelerating and at $t = 2\,\mathrm{s}$ while accelerating in forms of unpredictable undershoots and overshoots, respectively. According to the manufacturer, this behavior occurs due to the gear shifts in the gearbox, leading to overshoots and undershoots in the speed. Since the gearbox is treated as a black box, more information about the exact operation points where the gear shifts occur is not available. As a result, they cannot be directly modeled.

Nevertheless, the goal of identification in this Section is not to reach a perfect match but to primarily identify dead time and phase delay, which are the important features for the controller. This goal is achievable without making the model, and consequently, the resulting control concepts too complex to be applicable to this problem, given the limited resources such as the computing power on board. Furthermore, these deviations happen only occasionally and are quickly corrected by the system, as also shown in the Figure, which makes the use of a first-order low-pass element for the modeling sufficient.

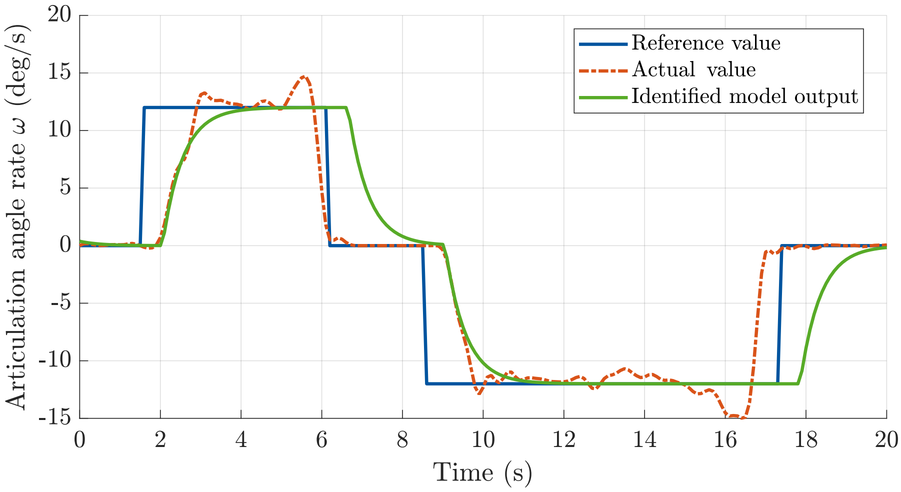


Fig. 3.2: Lateral identification experiments and results for the f-ADT.

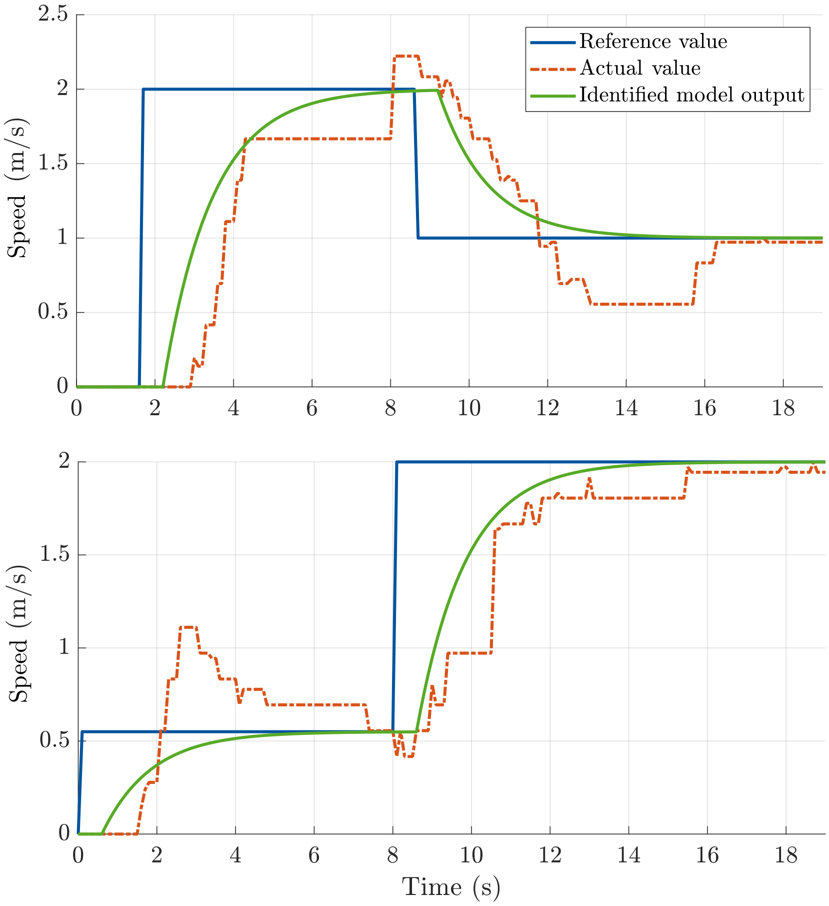


Fig. 3.3: Longitudinal identification experiments and results for the f-ADT.

By using MATLAB's identification toolbox, the following function has been fitted to the first-order low-pass element with dead time:

$$\frac{v(s)}{v_{\text{ref}}} = \frac{1}{Ts+1}e^{-T_t s}, \tag{3.2}$$

where s is the Laplace variable, T =1.25 s, and T_t =0.5 s. As these are the fitting results, the identified model might not perform as well in all scenarios.

3.3.4 Compact ADT: Lateral Identification

The lateral movement of the c-ADT is similar to the f-ADT in that they both are articulated vehicles. However, the c-ADT is considerably lighter and requires less torque to steer. Therefore, the maximum achievable articulation angle rate is not limited by the actuators but by the articulation angle reaching its end of the range (−30 deg to

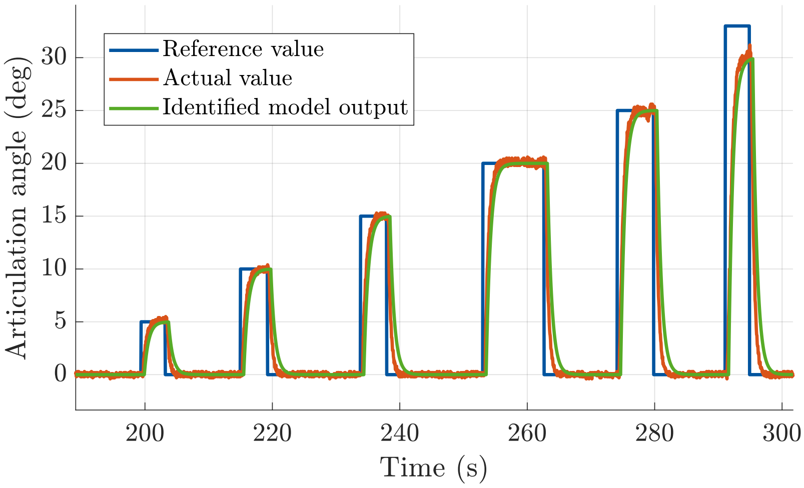


Fig. 3.4: Lateral identification experiments and results for the c-ADT.

30 deg). Consequently, it is possible to identify the articulation angle directly, without using the articulation angle rate. This is especially of interest since the rate is not a possible control input for this vehicle. Figure 3.4 shows a segment of the identification tests for the articulation angle and its results. The following is the identified equation:

$$\frac{\phi(s)}{\phi_{\text{ref}}} = \frac{1}{Ts+1} e^{-T_t s}, \tag{3.3}$$

where s is the Laplace variable, T =0.67 s, and T_t =0.5 s. As these are the fitting results, the identified model might not perform as well in all scenarios.

3.3.5 Compact ADT: Longitudinal Identification

Given the limitations of the vehicle discussed in Section 2.2.2, developing a speed controller was not possible for the vehicle. Consequently, the speed behavior could not be directly identified. As a result, it was decided that the best approach was to model and identify the motor RPM values starting from the linear motor's zero position. Furthermore, a lookup table was created that mapped each motor RPM to a speed making it possible to reach a reference speed. The experiment trajectories for testing the control concepts were later designed with ideally constant speed so that the identified function matches the vehicle.

Figure 3.5 shows a segment of the identification experiments. These experiments start at the RPM corresponding to the linear motor's zero position, which is approximately 1100RPM. A reference motor RPM is given to the vehicle and the data is recorded. The

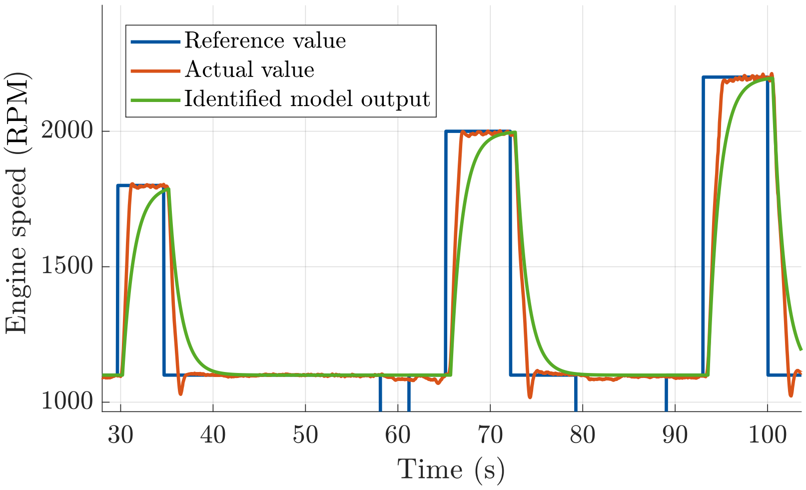


Fig. 3.5: Longitudinal identification experiments and results for the c-ADT.

identified first-order low-pass element is:

$$\frac{N_{\text{eng}}(s)}{N_{\text{eng,ref}}} = \frac{1}{Ts+1}e^{-T_t s}, \tag{3.4}$$

where s is the Laplace variable, N_{eng} is the engine RPM, $N_{\text{eng, ref}}$ is the reference engine RPM, T =1.25 s, and T_t =0.5 s. As these are the fitting results, the identified model might not perform as well in all scenarios.

3.4 Introduction to Vehicle Modeling

In this Section, the vehicle models used in the vehicle module of the digital twin are presented. In the following, a short introduction to wheel slip is presented since this physical property plays an important role in this work. This introduction is followed by the kinematic and dynamic modeling of the vehicle.

3.4.1 Wheel Dynamics

This Section is based on [63] and [68] and presents a practical yet short summary for slip, which is useful in the context of modeling an ADT dynamically. For more detailed information, please refer to [63].

Figure 3.6 is inspired by [63] and shows a side view of a tire with forces and torques affecting it and caused by it. In this context, the following variables must be defined. These variables will only be used in the context of this Chapter.

- Longitudinal and lateral speeds of the wheel are marked by V_x and V_y. These are elements of the total velocity vector V. Longitudinal here means at the direction where the wheel is pointed, and lateral here means perpendicular to the longitudinal direction.
- The longitudinal and lateral forces applied to the vehicle body from the wheel are marked by F_x and F_y. Longitudinal and lateral here have similar meanings as for the speeds. The normal load or vertical force acting on the wheel is denoted by F_z. Although the z axis is in the opposite direction of F_z, this force is defined to be positive in the upward direction for practical reasons and to comply with the literature [64].
- The rotational speed of the wheel around its rotational axis is marked by Ω.
- The effective radius of the wheel is denoted by r_e. Effective in this context means the radius of the wheel at free rolling.
- Effective rolling speed is marked by U and defined as $U = \Omega r_\mathrm{e}$.
- The self-aligning torque is denoted by M_z. It is the moment generated about the vertical axis of the tire, which causes the tire to align itself with the direction of travel. This is the moment exerted to the tire by the street.
- The torque applied to the wheel by the drivetrain is marked by T.

In the context of vehicle dynamics, slip refers to the relative difference between the actual velocity vector and the theoretically ideal velocity vector of an object. Theoretically ideal in this context indicates the velocity vector if the wheel were rolling without any sliding or skidding, rolling in the same direction the wheel is pointed at. In the following, firstly, the definition of slip in the wheels is presented and then the slip in the vehicle.

In the wheels, slip is possible due to the elasticity of a tire and manifests itself in the longitudinal and lateral elements, named slip ratio and slip angle, respectively. The longitudinal movement of a wheel is physically possible by the slip ratio, and a change in the direction of the movement is possible due to the slip angle of the wheel.

3.4.1.1 Longitudinal Wheel Slip

Longitudinal slip, which is associated with the slip ratio parameter σ, occurs, assuming ideal conditions, during acceleration and deceleration. When accelerating, a wheel's effective rolling speed U is faster than the actual longitudinal speed V_x due to tire deformation. When decelerating tire deformation leads to the effective rolling speed U

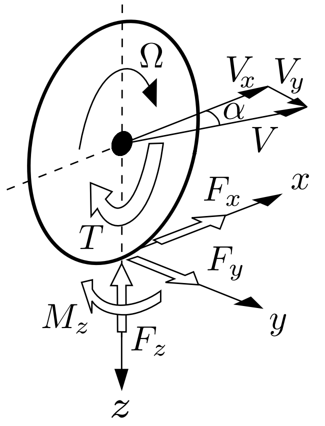


Fig. 3.6: Characteristic variables of a tire inspired from [64].

being smaller than the longitudinal speed V_x. Based on this phenomenon, the slip ratio is defined as follows:

$$\sigma = \begin{cases} \dfrac{r_{\mathrm{e}}\Omega - V_x}{r_{\mathrm{e}}\Omega} & \text{, during acceleration,} \\ \dfrac{r_{\mathrm{e}}\Omega - V_x}{V_x} & \text{, during deceleration.} \end{cases} \tag{3.5}$$

This ratio is defined in a way so that it is positive when accelerating and negative when decelerating.

3.4.1.2 Lateral Wheel Slip

Lateral wheel slip, which is associated with the wheel slip angle parameter α, happens when making a turn with the vehicle. In other words, it ideally does not occur when driving at free roll on a straight line. This quantity manifests itself as a difference between the direction of the actual wheel movement and the direction the wheel is pointing toward. Wheel lateral slip occurs when the wheel is trying to change the direction of the movement and keep the vehicle on a curve by applying centripetal forces to the vehicle chassis, and the vehicle body tries to maintain its longitudinal speed. This force transfer leads to tire deformation, especially at the contact point with the road. The wheel slip angle is therefore calculated by the following equation:

$$\alpha = -\arctan\frac{V_y}{V_x}. \tag{3.6}$$

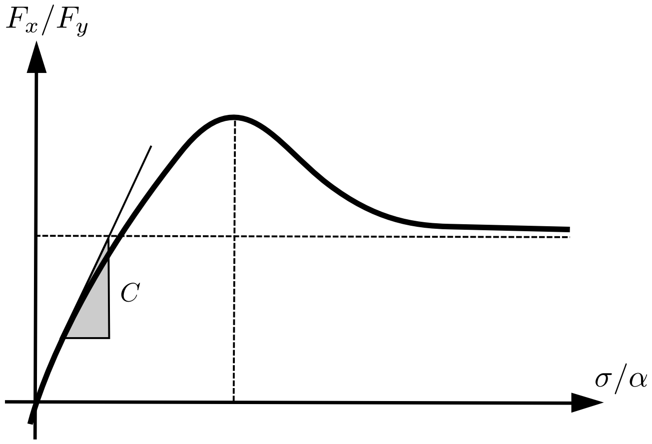


Fig. 3.7: Slip behavior of the tires inspired from [64].

3.4.1.3 Horizontal Wheel Forces

The importance of the wheel slip ratio and slip angle arises when calculating the wheel forces applied to the vehicle body. Figure 3.7 shows the relationship between the lateral force F_y and the wheel slip angle α, and the longitudinal force F_x and slip ratio σ. As this Figure also indicates, with the increasing slip angle or ratio, the resulting force increases at first as well. However, after a maximum is reached, where the grip is at its maximum, any further slip leads to less transferred force between the wheel and the vehicle. There have been many attempts in the literature to find and fit a function to the wheel slip behavior. The Magic Formula in [63] and the Dugoff model in [68] are amongst the best-known and used models. However, in cases where finding the tire parameters is not impossible or not easily possible, or when drifting does not occur, linear tire models can also be used, which are fitted to the curve at lower sideslip angles where the slope can be assumed to be linear, as shown in Figure 3.7. The linear tire models are the following for the longitudinal and lateral forces:

$$F_x = C_\sigma \sigma, \tag{3.7}$$

$$F_y = C_\alpha \alpha, \tag{3.8}$$

where C_α is the lateral and C_σ is the longitudinal slip coefficient. C_α is also called tire cornering stiffness.

3.4.2 Introduction to Kinematic Modeling

The modeling is generally referred to as kinematic when solely the kinematic characteristics and properties of the vehicle are used in the process of modeling. These include speed, and if desired, acceleration in both translational and rotational movement. These are translated into the vehicle's speed, acceleration, articulation angle ϕ, and rate ω in ADTs. Due to this limitation in choosing physical variables to model, not all the characteristics of a vehicle can be modeled kinematically due to their dynamic nature and the involvement of the dynamic properties, force and torque, in their behavior. One of the most important dynamics that are neglected in kinematic models is wheel dynamics, which leads to the assumption of no wheel slip angle in kinematic models. This will be explained in detail in the next Section. However, a kinematic model is easier to parameterize since it uses a limited number of parameters, which are generally easy to measure, such as vehicle length.

An important feature of the vehicle is the vehicle sideslip angle. The sideslip angle represents the difference between the vehicle's yaw angle and the actual direction of motion [76]. Figure 3.8 illustrates the sideslip angle in a vehicle. Kinematic models generally calculate the sideslip angle of the vehicle solely geometrically and based on the relative angle between the wheel and the vehicle body [76]. This is not a holistic representation for the sideslip angle since dynamic features such as wheel slip also contribute to it.

In the ADT kinematic models, since the angle between the wheels and the vehicle body is zero, the geometrical calculation of the sideslip angle obtains also a zero for the sideslip angle. In other words, in the kinematic modeling of an ADT, a sideslip-free motion is assumed. This assumption holds at lower speeds but decays with increasing speeds [4]. Dynamic modeling, however, considers sideslip angle and therefore can in theory better model the vehicle's motion especially in curves. Therefore, it is recommended to use dynamic models for the purpose of simulation. In the following, dynamic modeling is introduced.

3.4.3 Introduction to Dynamic Modeling

Dynamic modeling is the next category of modeling, which generally refers to models that include and consider forces, torques, mass, inertia, and, depending on the required level of complexity, other dynamic characteristics of a vehicle. As the list of dynamic properties also suggests, dynamic models can be developed arbitrarily complex for special purposes. For example, for modeling cabin vibrations, which are useful when analyzing the comfort in the cabin, or for modeling cylinder vehicle interactions when analyzing the durability of cylinders. Other examples of the dynamic characteristics that can be dynamically modeled include tire dynamics, suspension dynamics, mass transfer dynamics, and roll and pitch dynamics [76]. This does not mean that all of these dynamics are automatically included in all dynamic models, but that the possibility remains to add them to the model.

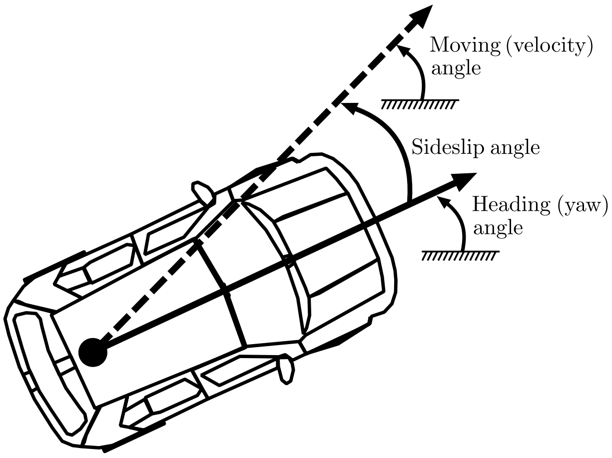


Fig. 3.8: Moving angle, yaw angle, and sideslip angle in a vehicle. In this context, an Ackermann vehicle has been chosen to define the concepts of sideslip angle for simplicity and better comparability with the existing literature. These concepts are applicable to other moving objects including ADTs.

One of the most important dynamic features in vehicles, which is included in vehicle dynamic models is wheel slip. Kinematic vehicle models neglect the wheel slip angle and assume it to be zero. As a result, dynamic models can more accurately describe a vehicle's cornering behavior since wheel slip is considerable when cornering.

In the course of this Chapter, both kinematic and dynamic models for an ADT are established, and the challenges in using either of them in control and estimation algorithms are discussed. Furthermore, methods are introduced to augment the sideslip angle into the kinematic model and to simplify the established dynamic model.

3.5 Kinematic Models of the Vehicle

3.5.1 Kinematic Vehicle Model in Inertial Coordinates (Baseline Model)

Figure 3.9 illustrates the kinematic bicycle configuration of an ADT. In this configuration, the front wheels of the ADT are replaced by a single wheel at the center of the front axle. If the configuration represents an ADT with two wheels on the rear body, the rear wheels are replaced by a single wheel placed at the center of the rear axle. For an ADT with four rear wheels, all four wheels are replaced by a single wheel located in the middle of the two rear axles. In this configuration, L_1 is the distance between the articulation hinge and the front axle, x_1 and y_1 are the Cartesian coordinates of the center of the

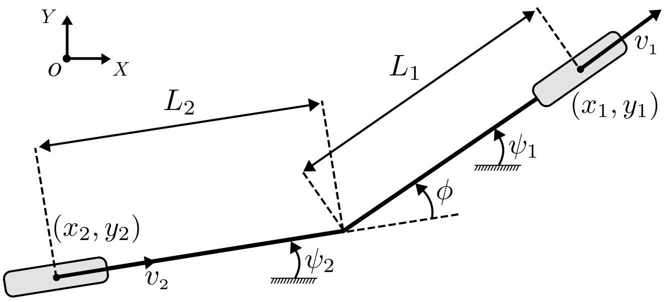


Fig. 3.9: Kinematic configuration of an ADT from [M2] ©2022 IEEE.

front axle in the inertial frame, v_1 is the speed at the center of the front axle, and ϕ is the articulation angle. Similar to the front body, L_2 is the distance between the articulation hinge and the center of the rear axles, x_2 and y_2 are the Cartesian coordinates of the center of the rear axles in the inertial frame, and v_2 is the speed at the center of the rear axles. The inertial frame is marked by $\mathcal{F}_O = \{O; \boldsymbol{i}, \boldsymbol{j}\}$. Note that the articulation angle is $\phi = \psi_1 - \psi_2$ and is therefore positive when steering left. The baseline kinematic model of an ADT can be expressed as follows [25]:

$$\dot{x}_1 = v_1 \cos \psi_1, \tag{3.9a}$$

$$\dot{y}_1 = v_1 \sin \psi_1, \tag{3.9b}$$

$$\dot{\psi}_1 = \frac{\sin \phi}{L_2 + L_1 \cos \phi} v_1 + \frac{L_2}{L_2 + L_1 \cos \phi} \omega, \tag{3.9c}$$

$$\dot{\phi} = \omega \quad (\phi = \psi_1 - \psi_2), \tag{3.9d}$$

where ω is the articulation angle rate. In this model, the inputs to the system are the current speed at the front axle v_1 and the current articulation angle rate $\omega = \dot{\phi}$. This model is the baseline kinematic model of an ADT and is established by assuming a sideslip-free motion of the vehicle. This is noticeable when observing the equations of motion for an ADT where $\dot{x}$ and $\dot{y}$ are functions of only the yaw angle ψ_1 and not the sideslip angle α. Furthermore, this model does not consider the actuator phase delay and dead times in the steering and speed equations. Using this model in a simulation or inside an MPC means to assume that the reference inputs for speed and articulation angle rate are instantly realized by the vehicle.

This problem must be addressed in both control and simulation contexts. In the control framework, it will be resolved by augmenting first-order low-pass elements into the existing equations, as detailed in upcoming Chapters 4, 5, and 7 prior to the controller setup. In contrast, the simulation will employ a similar approach with a different implementation, as illustrated in Figure 3.1. Here, first-order low-pass elements are introduced into the reference actuation signals before they enter the model, aligning with the identification functions established in Section 3.3.

3.5.2 Kinematic-Slip Vehicle Model in Inertial Coordinates (Kinematic-Slip Model)

As mentioned earlier, the baseline model assumes a sideslip-free motion and cannot represent the vehicle behavior accurately when compared to models that also include the sideslip angles. Based on [58], a kinematic-slip model of the vehicle is presented here that also includes the sideslip angles α and β in its equations while preserving the kinematic nature of the model. Figure 3.10 shows the kinematic-slip configuration of the vehicle, where v_1 and v_2 are the speed at the front and at the rear axle, respectively. The system equations are expressed as follows:

$$\dot{x}_1 = v_1 \cos(\psi_1 + \alpha), \tag{3.10a}$$

$$\dot{y}_1 = v_1 \sin(\psi_1 + \alpha), \tag{3.10b}$$

$$\dot{\psi}_1 = \frac{\sin(\phi + \alpha - \beta)}{L_2 \cos\beta + L_1 \cos(\phi - \beta)} v_1 + \frac{L_2 \cos\beta}{L_2 \cos\beta + L_1 \cos(\phi - \beta)} \omega, \tag{3.10c}$$

$$\dot{\phi} = \omega \quad (\phi = \psi_1 - \psi_2). \tag{3.10d}$$

3.5.3 Vehicle Model in Frenet Coordinates

The kinematic model of an ADT can also be expressed in Frenet coordinates so that the states of the system are not the Cartesian coordinates of the front axle in the inertial frame but the distance between the axle and a given path. This type of modeling is especially useful when designing a path-following controller. As mentioned earlier in Chapter 1.2, in [5], the vehicle model is presented in Frenet coordinates. Since there is a mistake in the model in that work, the derivation and the final model in Frenet coordinates are presented in this Chapter. This derivation is based on [56] and [M5] and is illustrated in Figure 3.11.

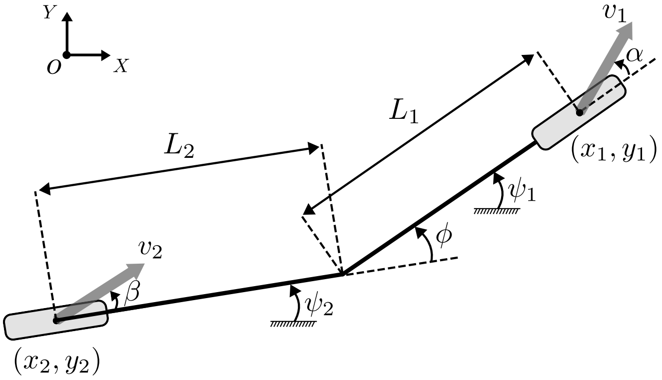


Fig. 3.10: Kinematic-slip configuration of an ADT from [M2] ©2022 IEEE.

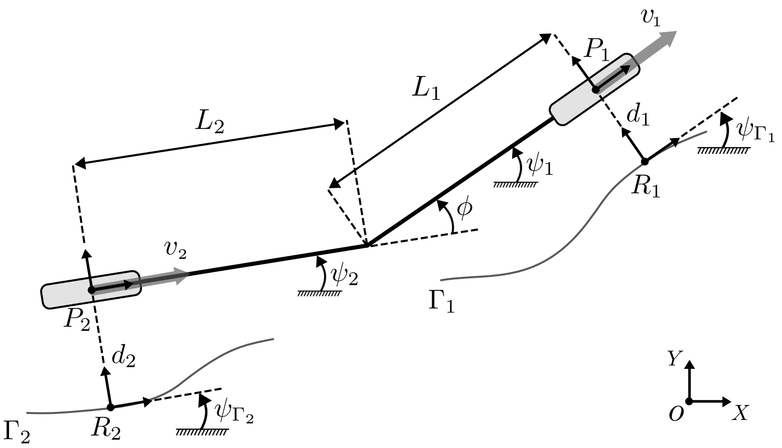


Fig. 3.11: Kinematic configuration of an ADT in Frenet coordinates from [M3].

Two reference curves Γ_1 and Γ_2 are assumed to be known for the front and rear axle, respectively. Five coordinate systems need to be defined, which are:

1. $\mathcal{F}_O = \{O; \boldsymbol{i}, \boldsymbol{j}\}$ the fixed inertial frame,
2. $\mathcal{F}_{P_1} = \{P_1; \boldsymbol{i}_{P_1}, \boldsymbol{j}_{P_1}\}$ the moving frame located on the front axle of the ADT,
3. $\mathcal{F}_{P_2} = \{P_2; \boldsymbol{i}_{P_2}, \boldsymbol{j}_{P_2}\}$ the moving frame located on the rear axle of the ADT,
4. $\mathcal{F}_{R_1} = \{R_1; \boldsymbol{i}_{R_1}, \boldsymbol{j}_{R_1}\}$ the moving frame located on Γ_1, where the reference point R_1 can be obtained by projecting P_1 on the path orthogonally,
5. $\mathcal{F}_{R_2} = \{R_2; \boldsymbol{i}_{R_2}, \boldsymbol{j}_{R_2}\}$ the moving frame located on Γ_2, where the reference point R_2 can be obtained by projecting P_2 on the path orthogonally. The reference points R_1 and R_2 exist and can be assumed to be unique when the distance between the ADT and the path is smaller than the momentary radius of the path.

Three sets of variables need to be defined:

- s_1, s_2: the curvilinear abscissa at the reference points R_1 and R_2.
- d_1, d_2: the distances between P_1 and R_1, and P_2 and R_2, respectively.
- $\Delta\psi_1 = \psi_1 - \psi_{\Gamma_1}$, $\Delta\psi_2 = \psi_2 - \psi_{\Gamma_2}$: the orientation errors of the front and rear body relative to the path. In other words, they are the difference between the front and rear yaw angles of the ADT and the tangent of the path at the reference points.

The derivation is continued for the front axle. The curvature at the path Γ_1 at the reference point R_1 is:

$$\kappa_{R_1} = \frac{\partial \psi_{R_1}}{\partial s_1}, \tag{3.11}$$

so that:

$$\dot{s}_1\kappa_{R_1} = \dot{\psi}_{\Gamma_1}. \tag{3.12}$$

The following equations are known for a sideslip-free motion:

$$\boldsymbol{r}_{R_1P_1} = d_1\boldsymbol{j}_{R_1}, \tag{3.13}$$

$$\frac{\mathrm{d}\boldsymbol{r}_{OR_1}}{\mathrm{d}t} = \dot{s}_1\boldsymbol{i}_{R_1}, \tag{3.14}$$

$$\frac{\mathrm{d}\boldsymbol{r}_{OP_1}}{\mathrm{d}t} = v_1\boldsymbol{i}_{P_1} \tag{3.15}$$

$$= v_1\cos\Delta\psi_1\boldsymbol{i}_{R_1} + v_1\sin\Delta\psi_1\boldsymbol{j}_{R_1}. \tag{3.16}$$

where $\boldsymbol{r}_{R_1P_1}$ is the position vector connecting R_1 and P_1. Furthermore, it is known that:

$$\frac{\mathrm{d}\boldsymbol{j}_{R_1}}{\mathrm{d}t} = -\dot{\psi}_{\Gamma_1}\boldsymbol{i}_{R_1} \tag{3.17}$$

$$= -\dot{s}_1\kappa_{R_1}\boldsymbol{i}_{R_1}. \tag{3.18}$$

Finally, since $\boldsymbol{r}_{OP_1} = \boldsymbol{r}_{OR_1} + \boldsymbol{r}_{R_1P_1}$, the velocity at P_1 can also be expressed as:

$$\frac{\mathrm{d}\boldsymbol{r}_{OP_1}}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{r}_{OR_1}}{\mathrm{d}t} + \frac{\mathrm{d}d_1\boldsymbol{j}_{R_1}}{\mathrm{d}t} \tag{3.19}$$

$$= \dot{s}_1\boldsymbol{i}_{R_1} + \dot{d}_1\boldsymbol{j}_{R_1} - \dot{s}_1\kappa_{R_1}\boldsymbol{i}_{R_1} \tag{3.20}$$

$$= \dot{s}_1(1 - d_1\kappa_{R_1})\boldsymbol{i}_{R_1} + \dot{d}_1\boldsymbol{j}_{R_1}. \tag{3.21}$$

Setting 3.16 equal to 3.21 yields the following:

$$\dot{s}_1 = v_1\frac{\cos\Delta\psi_1}{1 - d_1\kappa_{R_1}}, \tag{3.22}$$

$$\dot{d}_1 = v_1\sin\Delta\psi_1. \tag{3.23}$$

Now that two states of the system have been established, the differential equation for $\Delta\psi_1$ can be introduced. According to its definition, the following holds:

$$\Delta\dot{\psi}_1 = \dot{\psi}_1 - \dot{\psi}_{R_1}. \tag{3.24}$$

Substituting 3.9c and 3.12 in this equation yields the following for $\Delta\dot{\psi}_1$:

$$\Delta\dot{\psi}_1 = \frac{\sin\phi\ v_1 + L_2\omega}{L_2 + L_1\cos\phi} - \dot{s}_1\kappa_{R_1}. \tag{3.25}$$

For $\Delta\psi_2 = \dot{\psi}_2 - \dot{\psi}_{R_2}$, the following can be established:

$$\Delta\dot{\psi}_2 = \dot{\psi}_2 - \dot{\psi}_{R_2} \tag{3.26}$$

$$= \dot{\psi}_1 - \dot{\phi} - \dot{\psi}_{R_2} \tag{3.27}$$

$$= \frac{\sin\phi\ v_1 + L_2\omega}{L_2 + L_1\cos\phi} - \omega - \dot{s}_2\kappa_{R_2} \tag{3.28}$$

$$= \frac{\sin\phi v_1 - L_1\cos\phi\omega}{L_2 + L_1\cos\phi} - \dot{s}_2\kappa_{R_2}. \tag{3.29}$$

Now the state-space model of the system can be presented as follows:

$$\dot{s}_1 = v_1\frac{\cos\Delta\psi_1}{1 - d_1\kappa_{R_1}}, \tag{3.30}$$

$$\dot{d}_1 = v_1\sin\Delta\psi_1, \tag{3.31}$$

$$\Delta\dot{\psi}_1 = \frac{\sin\phi v_1 + L_2\omega}{L_2 + L_1\cos\phi} - \dot{s}_1\kappa_{R_1}, \tag{3.32}$$

$$\dot{s}_2 = v_{x2}\frac{\cos\Delta\psi_2}{1 - d_2\kappa_{R_2}}, \tag{3.33}$$

$$\dot{d}_2 = v_{x_2}\sin\Delta\psi_2, \tag{3.34}$$

$$\Delta\dot{\psi}_2 = \frac{\sin\phi v_1 - L_1\cos\phi\omega}{L_2 + L_1\cos\phi} - \dot{s}_2\kappa_{R_2}. \tag{3.35}$$

In this system, the equation for $\dot{s}_2$ and $\dot{d}_2$ were established similarly to those of $\dot{s}_1$ and $\dot{d}_1$ and were not discussed again.

This model can also potentially be presented with the inclusion of sideslip angles. However, since the resulting model is not further used in this work, the sideslip-kinematic model in Frenet coordinates will not be discussed here.

3.6 Dynamic Models of the Vehicle

In this Section, the dynamic modeling of an ADT is presented. At first, the dynamic model of an ADT without the inclusion of wheels and hydraulic cylinders is introduced. Then, a methodology is established that avoids the explicit modeling of the cylinders while taking their dynamic effect into consideration. The simulation environment is then complete when wheel models, power distribution, and low-level controllers for the speed and articulation angle are added. In addition to the full dynamic model, a simplified version is also presented that suits model-based algorithms better such as MPC or state estimation. Finally, the presented models are validated using real driving data.

The following dynamic models are established for the f-ADT of this project without loss of generality since the methodology can be adapted for all setups of ADTs, including the c-ADT of this project.

The contents of this Section are based on the publication [M2].

3.6.1 Full Dynamic Model of an ADT

3.6.1.1 Dynamic Model of an ADT Body

Before establishing the equations of motion for the vehicle, the required coordinate systems need to be presented. Figure 3.12 shows the dynamic configuration of an ADT. In this Figure, $(u_f,\, v_f)$ and $(u_r,\, v_r)$ are the front and rear speed elements, F_{x_i} and F_{y_i} are the wheel forces, T is the torque caused by the cylinders, R_x and R_y are the reaction forces at the pivoting joint, and ψ_1 and ψ_2 are the front and rear yaw angle.

Furthermore, a_1 and a_2 are the distance between the articulation hinge and the front and middle axle, respectively. The parameter b_1 is the distance between the front CoG and the front axle, while b_2 is the distance between the rear CoG and the middle axle. The distance between the rear CoG and rear axle is c_2. The track width in a vehicle is the distance between the centers of the wheels of the same axle. To simplify the derivation of the vehicle model, variables tw_1 and tw_2 have been defined as half of the front and rear track widths of the vehicle, respectively. Finally, the front body's mass and moment of inertia are marked by m_f and I_f while those of the rear body are marked by m_r and I_r, respectively.

- **The wheel-based frames**: Each wheel ($i \in [1, 6]$) has its own frame, which is fixed at the wheel.
- **Front body frame**: This is a vehicle-fixed frame at the front center of gravity (CoG).
- **Rear body frame**: This is a vehicle-fixed frame at the rear CoG.
- **Inertial frame**: This is the locally fixed frame to describe the position of the vehicle in Cartesian coordinates.

Using Newtonian mechanics, the equations of motion for a rotating system can be established for the front and rear body of the vehicle in two dimensions (2D) with four

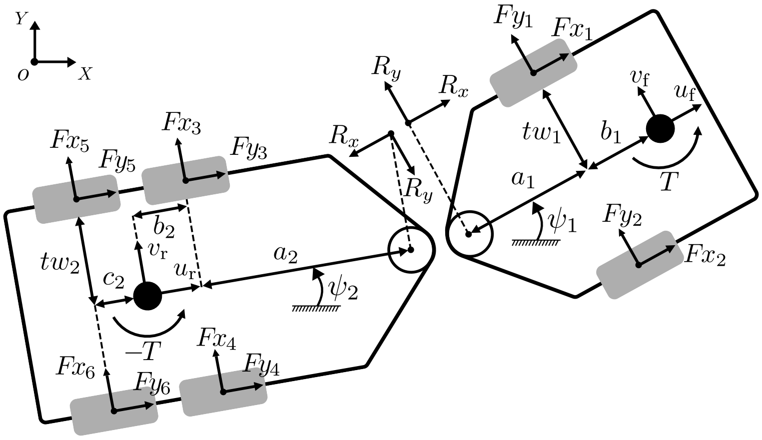


Fig. 3.12: Dynamic configuration of an ADT from [M2] ©2022 IEEE.

degrees of freedom. The following sets of variables are defined for the better readability of the equations:

$$\begin{aligned} c &= \cos\phi, & s &= \sin\phi, \\ d_1 &= a_1 + b_1, & d_2 &= a_2 + b_2, \\ F_{x_\mathrm{f}} &= F_{x_1} + F_{x_2}, & F_{y_\mathrm{f}} &= F_{y_1} + F_{y_2}, \\ F_{x_\mathrm{m}} &= F_{x_3} + F_{x_4}, & F_{y_\mathrm{m}} &= F_{y_3} + F_{y_4}, \\ F_{x_\mathrm{r}} &= F_{x_5} + F_{x_6}, & F_{y_\mathrm{r}} &= F_{y_5} + F_{y_6}, \\ \Delta F_{x_\mathrm{f}} &= F_{x_2} - F_{x_1}, & \Delta F_{x_\mathrm{m}} &= F_{x_4} - F_{x_3}, \\ \Delta F_{x_\mathrm{r}} &= F_{x_6} - F_{x_5}. \end{aligned} \tag{3.36}$$

The articulation angle in this system is defined similarly to the kinematic model and is:

$$\phi_\mathrm{dyn} = \psi_{1_\mathrm{dyn}} - \psi_{2_\mathrm{dyn}}. \tag{3.37}$$

Equations of motion for the front body are the following:

$$m_\mathrm{f}\,(\dot{u}_\mathrm{f} - v_\mathrm{f}\dot{\psi}_1) = F_{x_\mathrm{f}} + R_x, \tag{3.38}$$

$$m_\mathrm{f}\,(\dot{v}_\mathrm{f} + u_\mathrm{f}\dot{\psi}_1) = F_{y_\mathrm{f}} + R_y, \tag{3.39}$$

$$I_\mathrm{f}\,\ddot{\psi}_\mathrm{f} = T - b_1 F_{y_\mathrm{f}} - d_1 R_y + \Delta F_{x_\mathrm{f}}\, tw_1. \tag{3.40}$$

For the rear body, similar equations can be established:

$$m_\mathrm{r}\,(\dot{u}_\mathrm{r} - \dot{\psi}_2 v_\mathrm{r}) = -R_x\, c + R_y\, s + F_{x_\mathrm{r}} + F_{x_\mathrm{m}}, \tag{3.41}$$

$$m_\mathrm{r}\,(\dot{v}_\mathrm{r} + \dot{\psi}_2 u_\mathrm{r}) = -R_y\, c - R_x\, s + F_{y_\mathrm{r}} + F_{y_\mathrm{m}}, \tag{3.42}$$

$$\begin{aligned} I_\mathrm{r}\,\ddot{\psi}_2 = {} & -T + F_{y_\mathrm{m}}\, b_2 - F_{y_\mathrm{r}}\, c_2 \\ & + \Delta F_{x_\mathrm{m}}\, tw_2 + \Delta F_{x_\mathrm{r}}\, tw_2 - d_2\, R_y\, c - d_2\, R_x\, s. \end{aligned} \tag{3.43}$$

The goal here is to establish ordinary differential equations of the four degrees of freedom of the system, which are: u_f, v_f, ψ_1, and ψ_2. To achieve this, the reaction forces R_x and R_y, the rear body speed u_r and v_r, and accelerations $\dot{u}_\mathrm{r}$ and $\dot{v}_\mathrm{r}$ need to be eliminated from the equations. In the following, the relative speed and accelerations of the front and rear body are presented, which remove the rear body speed and accelerations from the equations when inserted in equations (3.38) and (3.41):

$$u_\mathrm{r} = u_\mathrm{f}\, c - s\, (v_\mathrm{f} - d_1\, \dot{\psi}_1), \tag{3.44}$$

$$v_\mathrm{r} = c\, (v_\mathrm{f} - d_1\, \dot{\psi}_1) - d_2\, \dot{\psi}_2 + u_\mathrm{f}\, s. \tag{3.45}$$

Calculating the derivatives yields the following for the acceleration:

$$\dot{u}_\mathrm{r} = \dot{u}_\mathrm{f}\, c - s\, (\dot{v}_\mathrm{f} - d_1\, \ddot{\psi}_1) - u_\mathrm{f}\, s\, (\dot{\psi}_1 - \dot{\psi}_2) - c\, (v_\mathrm{f} - d_1\, \dot{\psi}_1)\, (\dot{\psi}_1 - \dot{\psi}_2), \tag{3.46}$$

$$\dot{v}_\mathrm{r} = c\, (\dot{v}_\mathrm{f} - d_1\, \ddot{\psi}_1) - d_2\, \ddot{\psi}_2 + \dot{u}_\mathrm{f}\, s + u_\mathrm{f}\, c\, (\dot{\psi}_1 - \dot{\psi}_2) - s\, (v_\mathrm{f} - d_1\, \dot{\psi}_1)\, (\dot{\psi}_1 - \dot{\psi}_2). \tag{3.47}$$

After substituting the rear speed and acceleration with (3.44) to (3.47), the reaction forces were eliminated by applying algebraic manipulations, including addition, subtraction, and multiplication. This process resulted in the following set of four final ODEs that govern the overall dynamics of the system:

$$\begin{aligned} &m_\mathrm{f}\, \dot{u}_\mathrm{f} - F_{x_\mathrm{f}} + m_\mathrm{r}\, \dot{u}_\mathrm{f} - F_{x_\mathrm{m}}\, c - F_{x_\mathrm{r}}\, c - F_{y_\mathrm{m}}\, s - F_{y_\mathrm{r}}\, s + d_1\, m_\mathrm{r}\, \dot{\psi}_1^2 \\ &\quad - m_\mathrm{f}\, \dot{\psi}_1\, v_y - m_\mathrm{r}\, \dot{\psi}_1\, v_y - d_2\, m_\mathrm{r}\, \ddot{\psi}_2\, s + d_2\, m_\mathrm{r}\, \dot{\psi}_2^2\, c = 0 \end{aligned} \tag{3.48}$$

$$\begin{aligned} &- d_2\, m_\mathrm{r}\, s\, \dot{\psi}_2^2 - F_{y_\mathrm{f}} + m_\mathrm{f}\, \dot{v}_\mathrm{f} + m_\mathrm{r}\, \dot{v}_\mathrm{f} - F_{y_\mathrm{m}}\, c - F_{y_\mathrm{r}}\, c + F_{x_\mathrm{m}}\, s + F_{x_\mathrm{r}}\, s \\ &\quad - d_1\, m_\mathrm{r}\, \ddot{\psi}_1 + m_\mathrm{f}\, \dot{\psi}_1\, v_x + m_\mathrm{r}\, \dot{\psi}_1\, v_x - d_2\, m_\mathrm{r}\, \ddot{\psi}_2\, c = 0 \end{aligned} \tag{3.49}$$

$$\begin{aligned} &F_{y_\mathrm{f}}\, b_1 - T - \Delta F_{x_\mathrm{f}}\, tw_1 + I_\mathrm{f}\, \ddot{\psi}_1 + {d_1}^2\, m_\mathrm{r}\, \ddot{\psi}_1 - d_1\, m_\mathrm{r}\, \dot{v}_\mathrm{f} + F_{y_\mathrm{m}}\, d_1\, c + F_{y_\mathrm{r}}\, d_1\, c \\ &\quad - F_{x_\mathrm{m}}\, d_1\, s - F_{x_\mathrm{r}}\, d_1\, s - d_1\, m_\mathrm{r}\, \dot{\psi}_1\, v_x + d_1\, d_2\, m_\mathrm{r}\, \dot{\psi}_2^2\, s + d_1\, d_2\, m_\mathrm{r}\, \ddot{\psi}_2\, c = 0 \end{aligned} \tag{3.50}$$

$$\begin{aligned} &T - F_{y_\mathrm{m}}\, b_2 + F_{y_\mathrm{r}}\, c_2 + F_{y_\mathrm{m}}\, d_2 + F_{y_\mathrm{r}}\, d_2 - \Delta F_{x_\mathrm{m}}\, tw_2 - \Delta F_{x_\mathrm{r}}\, tw_2 + I_\mathrm{r}\, \ddot{\psi}_2 \\ &\quad + {d_2}^2\, m_\mathrm{r}\, \ddot{\psi}_2 - d_2\, m_\mathrm{r}\, \dot{v}_\mathrm{f}\, c - d_2\, m_\mathrm{r}\, \dot{u}_\mathrm{f}\, s - d_1\, d_2\, m_\mathrm{r}\, \dot{\psi}_1^2\, s \\ &\quad + d_1\, d_2\, m_\mathrm{r}\, \ddot{\psi}_1\, c - d_2\, m_\mathrm{r}\, \dot{\psi}_1\, v_x\, c + d_2\, m_\mathrm{r}\, \dot{\psi}_1\, v_y\, s = 0 \end{aligned} \tag{3.51}$$

Solving this system for the degrees of freedom yields the differential equation for each of the degrees of freedom and has the following format:

$$\dot{\boldsymbol{x}}_\mathrm{dyn} := \begin{bmatrix} \dot{u}_{\mathrm{f}_\mathrm{dyn}} \\ \dot{v}_{\mathrm{f}_\mathrm{dyn}} \\ \ddot{\psi}_{1_\mathrm{dyn}} \\ \ddot{\psi}_{2_\mathrm{dyn}} \end{bmatrix} = \begin{bmatrix} g_{\mathrm{dyn}_1}(F_{x_i}, F_{y_i}, T, \boldsymbol{x}_\mathrm{dyn}, \cdot) \\ g_{\mathrm{dyn}_2}(F_{x_i}, F_{y_i}, T, \boldsymbol{x}_\mathrm{dyn}, \cdot) \\ g_{\mathrm{dyn}_3}(F_{x_i}, F_{y_i}, T, \boldsymbol{x}_\mathrm{dyn}, \cdot) \\ g_{\mathrm{dyn}_4}(F_{x_i}, F_{y_i}, T, \boldsymbol{x}_\mathrm{dyn}, \cdot) \end{bmatrix} =: \boldsymbol{f}_\mathrm{dyn}(F_{x_i}, F_{y_i}, T, \boldsymbol{x}_\mathrm{dyn}, \cdot), \tag{3.52}$$

In this model, if the torque T is set to zero, the model becomes a tractor-trailer system. In a tractor-trailer system, the articulation angle reaches zero when driving on a straight line, even when starting at a non-zero articulation angle, and it will reach a non-zero angle when taking a curve, even when starting at a zero articulation angle. The following Section presents a method for the implicit inclusion of the cylinders via the torque T in the vehicle.

3.6.1.2 Implicit Modeling of the Steering Cylinders

As mentioned earlier in Section 2.1, the task of the cylinders is twofold, which is:

- to maintain the articulation angle when there is no steering command (or when the command requires the articulation angle to stay constant) and make the vehicle behave like a rigid body, which from now on is called operating scenario one (OS1),
- and to steer the vehicle when the steering command is present and non-zero to a given articulation angle or with a given rate, which is referred to as operation scenario two (OS2) from now on.

The goal of this section is to introduce a method of incorporating the cylinders and their dynamic effect on the vehicle without actually adding their model to the system. As mentioned before in Chapter 1.2, this has the benefit, that the resulting model is simpler to parameterize and has linearizable differential equations. The idea here is to calculate the torque that actual cylinders would generate to fulfill OS1 and OS2.

In order to do so, torques T_1 and T_2 are first introduced, where $T = T_1 + T_2$. T_1 is responsible for the OS1 and realizes this by using the inverse dynamics of the vehicle to compensate for all the forces and torques aimed at the articulation angle. T_2 is responsible for OS2.

Calculation of T_1

The starting point is the equation for the articulation angle $\phi_{\text{dyn}} = \psi_{1_{\text{dyn}}} - \psi_{2_{\text{dyn}}}$ and its second order derivative, which is set equal to zero:

$$\ddot{\phi}_{\text{dyn}} = \ddot{\psi}_{1_{\text{dyn}}} - \ddot{\psi}_{2_{\text{dyn}}} = 0. \tag{3.53}$$

In this equation, the following steps are to be taken:

- Set $T_2 = 0$.
- Solve (3.53) for T_1. At this point, the required torque to set (3.53) to zero is calculated T_1.
- In the vehicle model (3.52), T_1 is replaced by its newly calculated value.

In the resulting system, articulation angle rate $\ddot{\phi}_{\text{dyn}}$ can now only be affected by T_2 and other torques and forces on the system have no effect since they are canceled out by T_1.

Calculation of T_2

T_2 can now be calculated with the following simple P control law to achieve the reference articulation angle rate.

$$T_2 = P_\omega(\omega_{\text{ref}} - (\dot{\psi}_{1_{\text{dyn}}} - \dot{\psi}_{2_{\text{dyn}}})) = P_\omega(\omega_{\text{ref}} - \dot{\phi}_{\text{dyn}}), \tag{3.54}$$

where P_ω has to be chosen according to the identification results for articulation angle rate so that the simulation's articulation behavior is similar to the real vehicle. Due to the nature of the relation between the control variable (articulation angle rate) and manipulated variable (torque) and the existence of a natural integrator between them, the control loop has no steady-state errors. Figure 3.13 illustrates the calculation of T for the simulation environment. The results of the validation will be presented at the end of this Chapter.

If it is desired, to have reference articulation angle ϕ_{ref} instead of the reference articulation rate ω_{ref}, another layer of P control can be added to the system and the value of T_2 can be calculated with the following P control law:

$$T_2 = P_\omega(P_\phi(\phi_{\mathrm{ref}} - \phi_{\mathrm{dyn}}) - \dot{\phi}_{\mathrm{dyn}}). \tag{3.55}$$

This control law will also exhibit no steady-state error due to the existence of a natural integrator between the articulation angle and articulation angle rate, which are the control and manipulated variables, respectively.

3.6.1.3 Tire Model

With the model of the ADT body established, the tire models for the vehicle can be discussed. A tire model represents the forces applied to the vehicle's body F_{x_i}, F_{y_i}, $i \in [1, 6]$ when an input torque is applied to the wheels. As discussed in Section 3.4.1, there are a variety of tire models with different complexity and accuracy levels to be used depending on the specific use case.

Nonlinear tire models, such as the Magic Formula, are highly effective for accurate simulation. However, they require numerous parameters that can only be determined through tire testing. Since such testing was not possible, complex tire models like the Magic Formula were excluded from the simulation. Furthermore, due to the very rigid structure of tires in ADTs, the heavy weight of the vehicle that prevents the vehicle from drifting, and the speed limited to $30\,\mathrm{km\,h^{-1}}$, using a linear tire model presented a plausible option. Therefore, the tire linear model was used in the simulation with the following equations ([64]):

$$F_{x_i} = C_\sigma \sigma_i, \tag{3.56}$$

$$F_{y_i} = -C_\alpha \arctan \frac{v_{\mathrm{t},y_i}}{v_{\mathrm{t},x_i}}, \tag{3.57}$$

where C_σ is the longitudinal and C_α is the lateral slip coefficient, v_{t,y_i} and v_{t,x_i} are the lateral and longitudinal speeds at each tire. The letter t in subscript is short for tire. The longitudinal slip ratio is denoted by σ and can be calculated with:

$$\sigma_i = -\frac{v_{\mathrm{t},x_i} - r_{\mathrm{e}}\,\Omega_i}{v_{\mathrm{t},x_i}}, \tag{3.58}$$

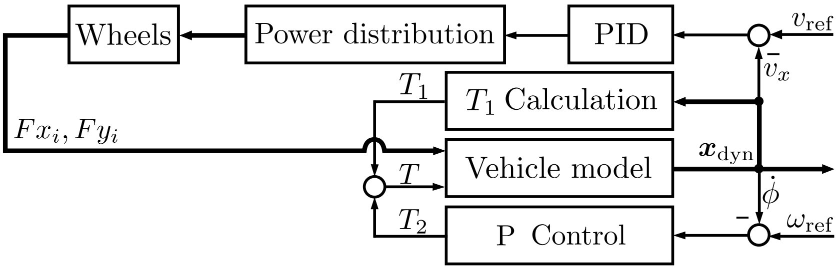


Fig. 3.13: Block diagram of the dynamic simulation environment from [M2] ©2022 IEEE.

where Ω is the tire rotational speed, and r_{e} is the effective radius of the tire.

3.6.1.4 Power Distribution

As mentioned before, ADTs normally have all-wheel drive, which is very suitable for their use case on uneven and unpaved mostly off-road terrain. Since the exact power distribution used for the ADT in this project is not known and falls into the patents of the manufacturer, and to maintain simulation generality, an alternative and simpler power distribution is acquired for the simulation. In [45] a simple power distribution is presented, where each wheel of a normal urban vehicle receives torque proportional to the inverse of its distance to the center of gravity in the car. This cannot be directly used for an ADT since there are two different centers of gravity. In this case, an interpretation of the power distribution can be utilized: each wheel receives torque proportional to the vertical load on it. This interpretation can be employed for the simulation since the weight of the front and rear bodies of the vehicle are known and are divided equally on their respective wheels. A PID control law controls the overall required torque for achieving a given reference speed. The P, I, and D elements are chosen in a way, that the speed behavior of the simulation is similar to the real vehicle, i.e., is similar to the identification results from Section 3.3.

With the dynamic model of the ADT body, steering cylinder representation, tire model, and power distribution established, the simulation environment can be set up in MATLAB/Simulink. The validation of the simulation environment is presented in Section 3.7.2.

3.6.2 Reduced Dynamic Model

The vehicle model presented in the previous Section has explicit models for the tires and power distribution and is suitable for simulation purposes. However, a simplified or reduced vehicle model is better suitable for online model-based algorithms such as model-based filters, controllers, and generally use cases where computationally less challenging equations are required. In the following, an approach is presented to achieve such a model in three steps:

Step one
In this reduced model, the vehicle does not have to be presented with two tracks and a bicycle presentation is sufficient. Therefore, it is assumed that the vehicle has a single track. To achieve this, the variables tw_1 and tw_2 are simply set to zero in (3.52).

Step two
Similar to the approach presented in Section 3.6.1.2, a method is introduced to calculate the required torque T for the two operating scenarios. However, in this new approach, it is not required to define T_1 and T_2. It is assumed that the articulation angle rate behavior of the ADT is known and can be described by a first-order low-pass element as follows:

$$\frac{\dot{\phi}}{\omega_{\mathrm{ref}}} = \frac{k_\omega}{T_\omega s + 1} \text{ which is equal to: } \ddot{\phi} = \frac{k_\omega \omega_{\mathrm{ref}} - \dot{\phi}}{T_\omega}, \tag{3.59}$$

where ω_{ref} is the reference, and $\dot{\phi}$ is the current articulation angle rate. Parametersk_ω and T_ω are the identified parameters. This equation describes the second-order derivative of the articulation angle and can therefore be set equal to (3.53), which also describes the same variable. This yields:

$$\frac{k_\omega \omega_{\mathrm{ref}} - \dot{\phi}}{T_\omega} = \ddot{\psi}_{1_{\mathrm{dyn}}} - \ddot{\psi}_{2_{\mathrm{dyn}}}. \tag{3.60}$$

When (3.60) is solved for the torque T, the required torque to achieve the identified articulation behavior from (3.59) can be found. When the resulting value for T is set in the model after **Step one**, the articulation input to the model changes from torque T to the reference articulation angle rate ω_{ref}.

Step three
The proposed power distribution from [45] can be used here as well. Since the weight of the vehicle in the front m_{f} and in the back m_{r} is known, the wheel forces $F_{x_i}, i \in [2, 6]$ can all be stated as functions of F_{x_1}:

$$\begin{aligned} F_{x_2} &= F_{x_1}, \\ F_{x_3} &= F_{x_4} = F_{x_5} = F_{x_6} = F_{x_1} \frac{m_{\mathrm{r}}}{2m_{\mathrm{f}}}. \end{aligned} \tag{3.61}$$

By using (3.61), wheel forces $F_{x_i}, i \in [2, 6]$ are removed from the equations and only F_{x_1} remains. Furthermore, the lateral forces of the tires $F_{y_i}, i \in [2, 6]$ can be replaced by the following equations, which is the simplified alternative to the linear tire model in

(3.56):

$$F_{y_i} = -C_\alpha \frac{v_{\mathrm{t},y_i}}{v_{\mathrm{t},x_i}}. \tag{3.62}$$

This simplification uses the small-angle approximation assuming that v_{t,y_i} is considerably smaller than v_{t,x_i}.

Step four
In this final step, similar to **Step two**, it is assumed that the longitudinal behavior of the vehicle is known and can be described using a first-order low-pass element as:

$$\frac{u_\mathrm{f}}{v_\mathrm{ref}} = \frac{k_v}{T_v s + 1} \text{ which is equal to: } \dot{u}_\mathrm{f} = \frac{k_v v_\mathrm{ref} - u_\mathrm{f}}{T_v}, \tag{3.63}$$

where v_ref is the reference speed, and u_f is the current longitudinal speed. Since (3.63) also describes the first-order derivative of the speed, it can be set equal to (3.52), which describes the same variable $\dot{u}_{\mathrm{f}_\mathrm{dyn}}$. This yields:

$$\frac{k_v v_\mathrm{ref} - u_{\mathrm{f}_\mathrm{dyn}}}{T_v} = \dot{u}_{\mathrm{f}_\mathrm{dyn}}. \tag{3.64}$$

When this equation is solved for F_{x_1}, and the result is put in the resulting equation from **Step three**, the longitudinal input of the system changes from F_{x_1} to the reference speed v_ref, and the reduced, bicycle model of an ADT is defined by the following equation:

$$\dot{\boldsymbol{x}}_\mathrm{red} = \boldsymbol{f}_\mathrm{red}(v_\mathrm{ref}, \omega_\mathrm{ref}, \boldsymbol{x}_\mathrm{red}, \cdot). \tag{3.65}$$

In summary, it was managed to establish a dynamic model for ADTs that is simpler to set up and requires less computation power when used in a program compared to the full dynamic model. This simplification was achieved by using a single track instead of two, removing the two-step calculation of the cylinder torque and introducing a simpler method to obtain it, and finally removing the explicit tire models and inserting implicit tire models inside the vehicle model instead.

3.7 Software in the Loop

In this Section, the setup of the SIL is presented by discussing the implementation of the vehicle module and its interfaces with other modules in the digital twin. Furthermore, the validation of the vehicle simulations using real-world data is presented.

3.7.1 Software in the Loop Setup

The SIL is implemented in MATLAB/Simulink. Figure 3.14 provides an overview of the top level of this environment. As the Figure illustrates, the Simulink blocks correspond to the modules of the digital twin, which are the controller, the planner, the vehicle, the sensor, and the state estimation module. In the vehicle module, the presented dynamic model, reduced dynamic model, and the baseline kinematic model are implemented for both the f-ADT and c-ADT. The setup of the dynamic models follows the implementations and equations presented in Sections 3.6.1 and 3.6.2 for the full and reduced dynamic model, respectively. The kinematic model setup is based on the model in Section 3.5.1. The remainder of the modules used in the SIL follows the discussion presented in Section 3.2.

The vehicle parameters are based on the data sheets of the vehicles [86] and [60] and are presented in Tables 3.1 and 3.2. For the tuning of the speed PID controller and the steering P controllers, the parameters were chosen with the help of the MATLAB PID toolbox in a way, that the resulting longitudinal and lateral behavior of the vehicle aligns with the identification results from Section 3.3.

The vehicle module runs with 1000 Hz with the fixed-step ODE45 setting.

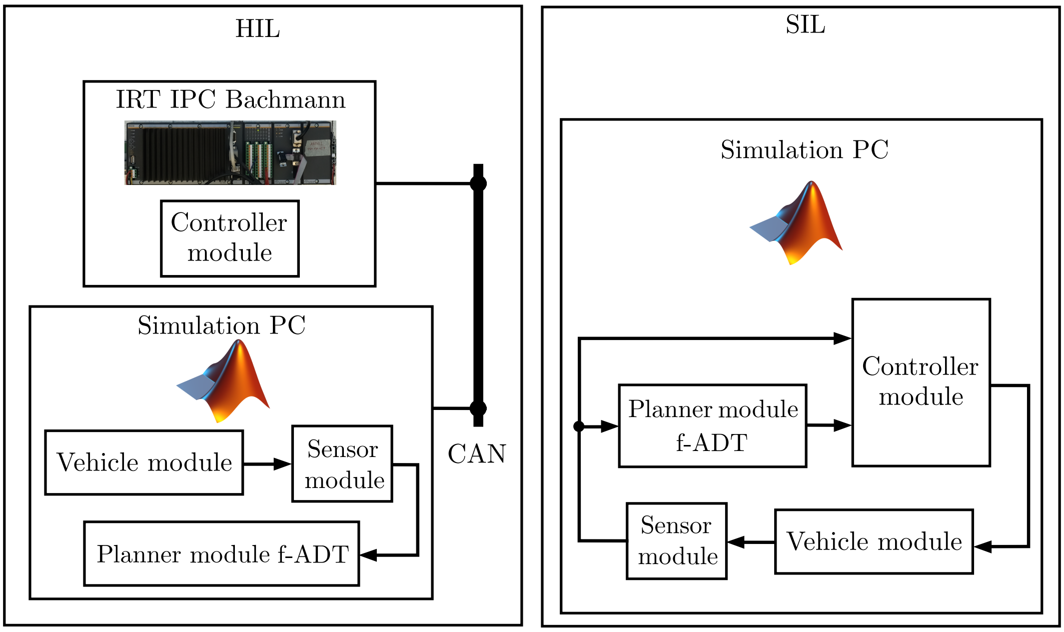


Fig. 3.14: Schematic realization of SIL and HIL in the digital twin.

Table 3.1: f-ADT dynamic model parameters

Parameter	Symbol	Value
Front body lengths	a_1, b_1, tw_1	1.36 m,0.9 m,1.2 m
Rear body lengths	a_2, b_2, c_2, tw_2	2.82 m,0.83 m,0.83 m,1.2 m
Tire stiffness	C_α, C_l	143 kN rad^{-1}, 127 kN
Vehicle mass	m_1, m_2	9750 kg, 9560 kg
Vehicle inertia	I_1, I_2	17 293 kg m^2,22 378 kg m^2
Tire radius and inertia	r_w, I_w	0.8 m, 184.6 kg m^2
Steering P control	P_ω, P_ϕ	25 000 Nms/deg, 2 s^{-1}
Speed PID control	P_v, I_v, D_v	23 000 N s^{-1}, 900 N s^{-1}, 800 N s^{-1}

Table 3.2: f-ADT kinematic model parameters

Parameter	Symbol	Value
Vehicle lengths	L_1, L_2	1.36 m,3.65 m
Steering P control	P_ϕ	2 s^{-1}

3.7.2 Software in the Loop Validation

3.7.2.1 Experimental Setup

In order to validate the simulation of the f-ADT, the results of a series of test drives in a mining field are compared with the results of the full dynamic, reduced dynamic, and kinematic simulation. During the real test drives, the speed, articulation angle, and position were recorded. To carry out the comparison, the speed and the articulation angle are given to the simulation as reference values, and the results are recorded. In Figure 3.15, this comparison is shown. In this test drive, the vehicle drives with a constant speed of 3 m s^{-1} and makes a left turn. The duration of this test drive is 35 s. In Figure 3.15, the comparison is in terms of the position of the vehicle, the articulation angle, and the moving angle.

The choice of moving angle for this comparison is based on the fact that it represents important dynamic aspects of the vehicle, which are lateral speed, yaw angle, and sideslip angle. This is evident from the equation of the moving angle, which is the sum of the yaw and sideslip angles. By defining θ to be the moving angle only in the context of this Section, it is given as:

$$\theta = \psi_1 + \alpha = \psi_1 + \arctan\frac{v_f}{u_f}, \tag{3.66}$$

where ψ_1 is the yaw angle of the front body, and α is the sideslip angle. This is shown in Figure 3.8 for a simple vehicle. While comparing positions provides a visual understanding, it can show a difference between the simulation and the real vehicle that does not necessarily indicate poor simulation behavior. This drift increases especially in longer simulations. In contrast, the moving angle provides a more immediate and meaningful comparison, which reflects the dynamic behavior of the vehicle better.

In the kinematic simulation, the moving angle is equal to the yaw angle, since a sideslip angle free ($\alpha = 0$) motion is assumed. In the dynamic simulations, the moving angle is calculated using (3.66). The moving angle of the vehicle in the experiments is measured with the localization data from the localization module as the ground truth.

3.7.2.2 Experimental Results

As Figure 3.15 illustrates, all simulations were able to follow the desired articulation angle similarly and with negligible deviation. This indicates the successful identification of the vehicle and tuning of the steering controller.

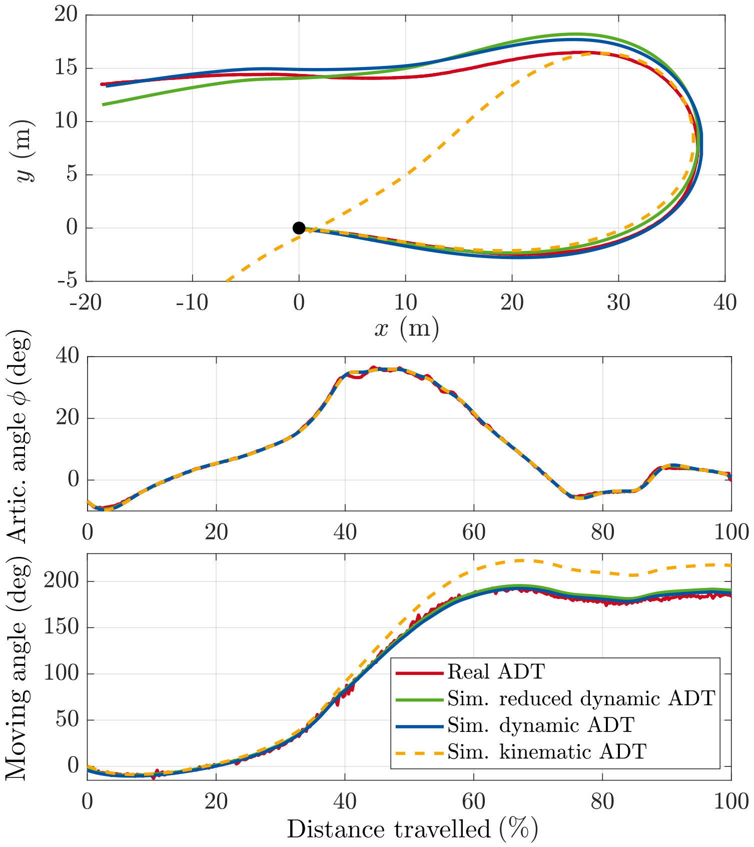


Fig. 3.15: Comparison between the real ADT in a mining field, the full dynamic, the reduced dynamic, and the kinematic simulation from [M2] ©2022 IEEE.

The results of the moving angle indicate a moving angle error of 3 deg, 5 deg, and 31 deg for the full dynamic, reduced dynamic, and kinematic simulations, respectively. The significant moving error in the case of the kinematic model can be explained by the kinematic nature of this model and that it assumes a sideslip angle-free motion. This error is responsible for the noticeable accumulated position error at the end of the simulation. The full dynamic simulation delivers a more accurate moving angle due to its more complex steering dynamic and its two-track model, in contrast to the reduced dynamic model.

Although the results of the experiment seem promising and can validate the full dynamic model simulation, it is important to mention that the experiment and t he models were designed for the requirements of this work that were defined in Section 1.2.2. These requirements are tailored for the purposes of this project. If, however, having a higher accuracy for the moving error in more dynamic scenarios is also desired, the following suggestions can be considered for the model:

- More accurate tire models such as Magic Formula [64].
- Using a three-dimensional model with nine degrees of freedom, which can better simulate roll and pitch angles and their effect on the overall behavior.
- Alternatively, methods to estimate wheel load shift and the roll angle can be used that replicate the roll and pitch angle effects on the vehicle in two-dimensional modeling [76].
- More accurate power distribution that is closer to the power distribution of the real vehicle, which is possible when a model of the vehicle's power distribution is available.

3.8 Hardware in the Loop

The HIL system was developed only for the c-ADT since the f-ADT's hardware was developed by Indurad GmbH.

The objective of the HIL system for the c-ADT was to create an environment where the compatibility of the different modules of the software framework (see Section 2.2.4) with the IRT IPC could be tested. Furthermore, the HIL could facilitate debugging scenarios so that the functionality of the system could be tested and assured prior to the field tests. The HIL setup consists of the IRT IPC with the software framework installed and a secondary PC where a simulation of the vehicle runs in MATLAB. The communication between the two systems is via CAN to replicate the real-world setup. The software framework on the IRT IPC was explained in detail in Chapter 2. Figure 3.14 provides an overview of the HIL.

3.9 Summary

In this chapter, the digital twin of the ADT was established by first defining the various modules of the digital twin, followed by presenting a novel approach to modeling the vehicle, which serves as the pivotal component of the digital twin. The other modules will be detailed throughout the remainder of this work.

To establish the vehicle module, the identification tests conducted on the vehicles were presented, along with an evaluation of the results. The objective of these tests was to model and parameterize the behaviors of the actuators by employing first-order low-pass elements and introducing them to the vehicle models.

The modeling process began with the presentation of different kinematic models, followed by the development of a novel dynamic model in accordance with the requirements defined in Section 1.2.2. While the established dynamic model is suitable for simulation purposes, a simplified version was also introduced for additional use cases with limited computational resources. Subsequently, the SIL environment was set up using the presented models and identification results, with validation conducted using real-world data. Finally, the HIL setup was established, facilitating early debugging and functionality testing.

4 Path-Following Control Concepts

This Chapter initiates the control design process for this work by first presenting two path-following controllers. One is based on classical feedback control methods while the other is an MPC. Subsequently, this Chapter presents the simulation results of the controllers, followed by field test results in a mining field with the f-ADT. Finally, a comparison discussion about these controllers is presented at the end of this Chapter. This comparison showcases the abilities of model-based controllers compared to classical feedback controllers and presents the benefits of employing actuator identification in the control concept. The contents of this Chapter correspond to [M3].

4.1 Path-Following MPC

This Section presents the components that the MPC setup requires.

4.1.1 MPC Model

The chosen model for the path-following MPC is a simplified variation of (3.30), which is presented in the following in its original equations again for ease of access:

$$\dot{s}_1 = v_1 \frac{\cos \Delta\psi_1}{1 - d_1 \kappa_{R_1}}, \tag{4.1}$$

$$\dot{d}_1 = v_1 \sin \Delta\psi_1, \tag{4.2}$$

$$\Delta\dot{\psi}_1 = \frac{\sin \phi v_1 + L_2 \omega}{L_2 + L_1 \cos \phi} - \dot{s}_1 \kappa_{R_1}, \tag{4.3}$$

$$\dot{s}_2 = v_2 \frac{\cos \Delta\psi_2}{1 - d_2 \kappa_{R_2}}, \tag{4.4}$$

$$\dot{d}_2 = v_2 \sin \Delta\psi_2, \tag{4.5}$$

$$\Delta\dot{\psi}_2 = \frac{\sin \phi v_1 - L_1 \cos \phi \omega}{L_2 + L_1 \cos \phi} - \dot{s}_2 \kappa_{R_2}. \tag{4.6}$$

The inputs to this model are the speed at the front axle v_1, articulation angle rate ω, and front and rear curvatures κ_{R_1} and κ_{R_2}. These inputs should not be mistaken with control inputs of the MPC, which will be discussed later. The rest of the variables and parameters were previously introduced in Section 3.5.3 and will not be reintroduced here again. As the equations of this model indicate, the model exhibits many nonlinearities specifically due to the existence of the input variables κ_{R_1} and κ_{R_2} in the denominator of (4.1) and (4.4). However, by making the following assumptions and the resulting adjustments in the model, the model and consequently the linearized model can be simplified. These simplifications are partly based on [M5].

Assuming that the orientation errors $\Delta\psi_1$ and $\Delta\psi_2$ are close to zero, $\cos\Delta\psi_1$ and $\cos\Delta\psi_2$ can be approximated to be equal to one. Furthermore, by assuming that the curvatures of the reference paths are also close to zero, indicating a path without sharp curves, the denominator of the equation for $\dot{s}_1$ and $\dot{s}_2$ can be replaced by one. This yields the following for $\dot{s}_1$ and $\dot{s}_2$:

$$\begin{aligned} \dot{s}_1 &= v_1 \frac{\cos\Delta\psi_1}{1 - d_1\kappa_{R_1}} \qquad (4.7) \\ &\approx v_1, \\ \dot{s}_2 &= v_2 \frac{\cos\Delta\psi_2}{1 - d_2\kappa_{R_2}} \qquad (4.8) \\ &\approx v_2. \end{aligned}$$

Now that the model is established, the identification equation from Section 3.3 from (3.1) should be added to the vehicle model. As discussed in Section 1.2.4, without the inclusion of the identification equations, the MPC assumes that the reference for the articulation angle can be realized by the vehicle instantly. However, this assumption is not valid for most real systems and specifically for ADTs considering their phase delays.

Using the discussed approximations and including the articulation identification equation yields the following kinematic ADT model in Frenet coordinates:

$$\begin{aligned} \dot{d}_1 &= v_1 \sin\Delta\psi_1, \qquad (4.9) \\ \Delta\dot{\psi}_1 &= \frac{\sin\phi v_1 + L_2\omega}{L_2 + L_1\cos\phi} - v_1\kappa_{R_1}, \qquad (4.10) \\ \dot{d}_2 &= v_2 \sin\Delta\psi_2, \qquad (4.11) \\ \Delta\dot{\psi}_2 &= \frac{\sin\phi v_1 - L_1\cos\phi\omega}{L_2 + L_1\cos\phi} - v_2\kappa_{R_2}, \qquad (4.12) \\ \dot{\phi} &= \omega, \qquad (4.13) \\ \dot{\omega} &= (-\omega + k_\omega\omega_{\text{ref}})/T_\omega. \qquad (4.14) \end{aligned}$$

The state vector of this system is $\boldsymbol{x}_{\text{pf-Fr}} = [d_1, \Delta\psi_1, d_2, \Delta\psi_2, \phi, \omega]^T$, with the subscript indicating path-following control using the Frenet model, and the input to this system is simply the reference articulation angle rate $u_{\text{pf-Fr}} = \omega_{\text{ref}}$.

Speed at the front axle v_1 is an input to the model but is not a control variable and is assumed to remain constant over the MPC's prediction horizon. This assumption is valid since the vehicle drives with a constant speed or only experiences instants of speed change. Therefore, the identification equation for speed was not added to this model. From the perspective of the MPC, the speed v_1 is a parameter. The value of this parameter is set equal to the current speed of the vehicle each time the MPC is called.

As one of the main factors for choosing an MPC is its predictive aspect, it is important to discuss how the MPC receives information about the future by analyzing the model. In many cases that involve a tracking MPC, the MPC attempts to follow a reference trajectory over time. This means, that the reference itself provides the MPC with information about the future of the environment. In this model, however, as the state vector indicates, the lateral distances and orientation errors are the states. Consequently, the future reference for these states is to maintain values as close to zero as possible. As a result, the future reference is constant and is equal to zero. Consequently, for this model, the reference cannot provide the MPC with information about the future. In this case, the future curvatures κ_{R_1} and κ_{R_2} are the elements to provide information about the future path. The curvatures of the upcoming path are considered as a known disturbance in this model and form the disturbance vector $\boldsymbol{z}_{\text{pf-Fr}} = [\kappa_1, \kappa_2]^T$.

While disturbance inputs do not typically need to be included in the MPC model, in the case of this modeling, their inclusion plays an integral role in the functionality of the MPC. This is because these curvatures are known in advance and provide the MPC with critical information about the future of the paths. Removing them from the model partly eliminates the predictive advantage that MPCs have over feedback controllers.

4.1.2 Model Linearization and Discretization

In this Section, the approach presented in [1] and [16] is utilized for setting up the MPC. This approach requires linearized and time-discrete system matrices. In the following, the linearization and discretization of a general nonlinear continuous system are presented.

The nonlinear continuous system in the following is assumed:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{4.15}$$

$$\boldsymbol{y}(t) = \boldsymbol{C} \cdot \boldsymbol{x}(t), \tag{4.16}$$

where $\boldsymbol{f}$ is the set of nonlinear system differential equations, and $\boldsymbol{x}$, $\boldsymbol{u}$, and $\boldsymbol{y}$ are the state, input, and output vectors. Matrix $\boldsymbol{C}$ is the output matrix. For simplicity and since it applies to all systems of this work, a linear output equation is assumed as also shown in (4.16).

Utilizing the Jacobian matrix at the current operation point, which is denoted by the letters op in the subscript, $(\boldsymbol{x}_{\mathrm{op}}, \boldsymbol{u}_{\mathrm{op}})$ yields the linearized system matrices as follows:

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_{\mathrm{op}},\boldsymbol{u}_{\mathrm{op}}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{\boldsymbol{x}_{\mathrm{op}},\boldsymbol{u}_{\mathrm{op}}}, \tag{4.17}$$

$$\boldsymbol{B} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}_{\mathrm{op}},\boldsymbol{u}_{\mathrm{op}}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}\right|_{\boldsymbol{x}_{\mathrm{op}},\boldsymbol{u}_{\mathrm{op}}}, \tag{4.18}$$

assuming that the system has n states and m inputs. Here, the index in $\square_i$ marks the $i-$th element of the associated vector. The Jacobian is executed every time the MPC is called to have the current linearized system at the current operation point $(\boldsymbol{x}_{\mathrm{op}}, \boldsymbol{u}_{\mathrm{op}})$. The next step is to discretize the system. There are multiple approaches to discretize the system model. According to [1], matrix exponential is chosen. With the sampling time T_{s} the time-discrete system matrices are as follows:

$$\boldsymbol{A}_{\mathrm{d}} = e^{\boldsymbol{A}T_{\mathrm{s}}}, \tag{4.19}$$

$$\boldsymbol{B}_{\mathrm{d}} = (e^{\boldsymbol{A}T_{\mathrm{s}}} - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{B}, \tag{4.20}$$

$$\boldsymbol{C}_{\mathrm{d}} = \boldsymbol{C}. \tag{4.21}$$

The subscript d indicates time-discrete elements. This yields the following linear and time-discrete system model:

$$\boldsymbol{x}_{\mathrm{d}}(k+1) = \boldsymbol{A}_{\mathrm{d}}(k)\boldsymbol{x}_{\mathrm{d}}(k) + \boldsymbol{B}_{\mathrm{d}}(k)\boldsymbol{u}_{\mathrm{d}}(k), \tag{4.22}$$

$$\boldsymbol{y}_{\mathrm{d}}(k) = \boldsymbol{C}_{\mathrm{d}}\boldsymbol{x}_{\mathrm{d}}(k). \tag{4.23}$$

What is presented in this Section so far, is the linearization and discretization approach for systems with the form presented in (4.15) and (4.16), which holds for the rest of the systems in this work. However, the system in this Chapter has an additional element, namely the input disturbance, and has the following form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{z}(t)), \tag{4.24}$$

$$\boldsymbol{y}(t) = \boldsymbol{C} \cdot \boldsymbol{x}(t), \tag{4.25}$$

where $\boldsymbol{z}(t)$ is the disturbance vector. For this system, the linearization has the following additional step:

$$\boldsymbol{E} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}\right|_{\boldsymbol{x}_{\mathrm{op}},\boldsymbol{u}_{\mathrm{op}}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \dots & \frac{\partial f_1}{\partial z_l} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z_1} & \dots & \frac{\partial f_n}{\partial z_l} \end{bmatrix}\right|_{\boldsymbol{x}_{\mathrm{op}},\boldsymbol{u}_{\mathrm{op}}}, \tag{4.26}$$

where l is the number of disturbance elements. Consequently, the discretization also has the following additional step:

$$\boldsymbol{E}_\mathrm{d} = (e^{\boldsymbol{A}T_\mathrm{s}} - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{E}, \tag{4.27}$$

yielding the following linear and time-discrete model:

$$\boldsymbol{x}_\mathrm{d}(k+1) = \boldsymbol{A}_\mathrm{d}(k)\boldsymbol{x}_\mathrm{d}(k) + \boldsymbol{B}_\mathrm{d}(k)\boldsymbol{u}_\mathrm{d}(k) + \boldsymbol{E}_\mathrm{d}(k)\boldsymbol{z}_\mathrm{d}(k), \tag{4.28}$$

$$\boldsymbol{y}_\mathrm{d}(k) = \boldsymbol{C}_\mathrm{d}\boldsymbol{x}_\mathrm{d}(k). \tag{4.29}$$

Following the aforementioned steps for the system of this Chapter yields the following system:

$$\boldsymbol{x}_\text{pf-Fr,d}(k+1) = \boldsymbol{A}_\text{pf-Fr,d}(k)\boldsymbol{x}_\text{pf-Fr,d}(k) + \boldsymbol{b}_\text{pf-Fr,d}(k)u_\text{pf-Fr,d}(k) + \boldsymbol{E}_\text{pf-Fr,d}(k)\boldsymbol{z}_\text{pf-Fr,d}(k). \tag{4.30}$$

The matrices of this system can now be utilized in the MPC approach.

4.1.3 Dead Time Compensation

As mentioned in Section 1.2.4, one of the important features of MPCs is the ability to consider actuator dead times in the control process. This can be achieved directly by including system dead times in the MPC model and in the formulation of the Hessian matrix [87]. While this method is a viable solution to the dead time problem, it relies on the linearized model to compensate for the dead time, which progressively loses accuracy as it moves away from the linearization point. As a result, this method may pose issues when the dead time is relatively long, which is the case in ADTs.

To address these concerns, another method can be employed where the predicted vehicle states after the dead time $\widehat{\boldsymbol{x}}$ are calculated externally before the MPC and are given to the MPC instead of the real current states. This is shown in Figure 4.1. The vector $\boldsymbol{x}_\mathrm{ref}$ refers to the reference path. The predicted state $\widehat{\boldsymbol{x}}$ can be obtained by using the nonlinear vehicle model and numerically integrating with buffered control variables, starting from the actual current state, to find the state after the dead time. For the

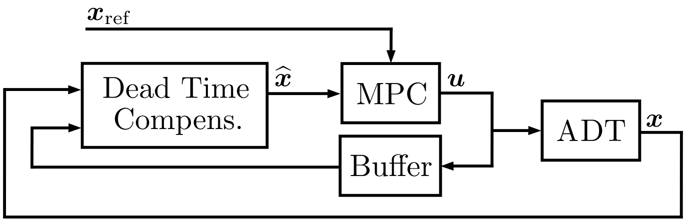


Fig. 4.1: Structure of the MPC and the dead time compensation.

integration, different methods such as Euler forward or Runge Kutta can be used. The basic idea behind this method is similar to the Smith predictor. The nonlinear prediction model is the baseline kinematic model (3.9) augmented with identification equations. This approach is used across this work for dead time compensation for all MPCs unless indicated otherwise.

4.1.4 Reference Path Preparation

The curvatures of the upcoming reference paths are considered as known disturbances for the MPC and must be calculated each time the MPC is called for all the upcoming path points within the MPC prediction horizon. Calculating the curvature at a point on a given path requires the coordinates of two adjacent points as well. Furthermore, the points where the curvature is determined must align with the MPC horizon and its discrete time step. This means that given that the MPC model is time-discrete with a discretization step of T_s, the curvatures must be computed at N points that are equidistant in time starting from the orthogonal projection of an axle's position onto the path. Figure 4.2 illustrates these points on a path.

Figure 4.2 shows not only the vehicle and the future points where the curvature should be calculated but also the predicted vehicle position after the dead time, illustrated by a transparent vehicle. As mentioned in Section 4.1.3, in order to compensate for the system dead time, the vehicle state after the dead time is calculated and given to the MPC as the current system state. With the same logic, the path curvatures given to

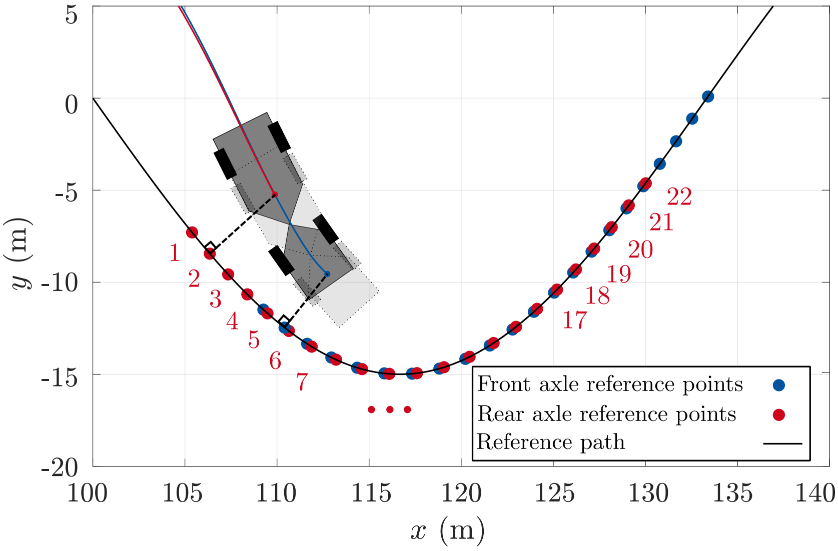


Fig. 4.2: Path points for the front and rear axle of the ADT. For better readability, the path points of the rear axle are numbered. In this scenario, $N = 20$.

the MPC also have to be the path curvatures that the vehicle encounters after the dead time. This is why the orthogonal projection of the predicted vehicle is used as the first point where the curvature is calculated and not the orthogonal projection of the actual current axle position. As the Figure also indicates, a total of $N + 2$ points are marked so that the curvature of the inner N points of interest can be calculated.

In order to find these time-equidistant points on a given path, the time distance between them T_s is translated into the physical distance, which is $T_\mathrm{s} \cdot v_1$, representing the distance the vehicle can travel in T_s. An algorithm is utilized here that given a path and an initial point and direction, locates the point on the path in a physical distance of $T_\mathrm{s} \cdot v_1$ from the initial point. The algorithm starts with the orthogonal projection of an axle on the path and then repeats this operation for another N times forward and one single time backward to obtain the total required $N + 2$ points.

After the coordinates of these points are established, the MATLAB File Exchange library "2D Line Curvature and Normals" [47] is utilized to calculate the curvatures.

4.1.5 Cost Function

For the path-following problem, a quadratic cost function is utilized, which aims to

- ensure good tracking behavior by minimizing the lateral tracking errors d_1 and d_2,
- enhance stability and maintain consistent and predictable vehicle dynamics by minimizing the orientation error $\Delta\psi_1$ and $\Delta\psi_2$ and aligning the vehicle with the direction of the path,
- reduce sideslip angle and skidding by keeping the articulation angle ϕ and rate ω at minimum.

The cost function achieves these goals by minimizing all state values and is given in the following:

$$\begin{aligned}
\min_{u_{\text{pf-Fr,d}}(\cdot|k)} \quad & \sum_{i=1}^{N_\mathrm{c}} \|\boldsymbol{x}_{\text{pf-Fr,d}}(k+i|k)\|_{\boldsymbol{Q}}^2 + && (4.31\text{a}) \\
& + \|u_{\text{pf-Fr,d}}(k+i-1|k) - u_{\text{pf-Fr,d}}(k+i-2|k)\|_r^2 && \\
\text{s.t.} \quad & \boldsymbol{x}_{\text{pf-Fr,d}}(\cdot|k) \in \mathbb{X}, \quad u_{\text{pf-Fr,d}}(\cdot|k) \in \mathbb{U}, && (4.31\text{b})
\end{aligned}$$

where $\boldsymbol{Q}$ and r are the weight elements, and $\mathbb{X}$ and $\mathbb{U}$ are the set of feasible states and control inputs, respectively.

The only constraints that the MPC must consider are regarding the physical constraints of the vehicle, which are a maximum articulation angle and a maximum articulation angle rate. These two constraints are added to the MPC formulation.

The chosen solver for the MPC is QP KWIK from MATLAB, which is based on [75]. The MPC runs at 20 Hz both in the simulation and in the experiment. Table 4.1 shows the rest of the MPC parameters.

4.2 Path-Following Feedback Control with Stanley Controller

Stanley is the name of an autonomous vehicle from Stanford University that took part in the DARPA Grand Challenge in 2005 and won first place [84]. This controller does not consider the kinematics, nor the delays of the vehicle and can therefore be implemented with relatively less effort compared to MPCs. In the following, this controller is presented.

The control law of the Stanley controller is:

$$\phi_{\mathrm{ref}} = (\psi_1 - \psi_{\Gamma_1}) + \arctan \frac{K_{\mathrm{S}} d_1}{v_1}. \tag{4.32}$$

In this method, K_{S} is the gain parameter that should be adjusted for the controller, and ϕ_{ref} is the articulation angle reference. The other variables are previously described in detail in Section 3.5.3 and shown in Figure 3.11.

While this controller is primarily created for Ackermann-steered vehicles, it is used for ADTs here without adjustment. This is because the essential goal of a controller in Ackermann vehicles and in ADTs is alike, which is to minimize the distance to a given reference path while keeping the relative orientation to the path at a minimum. This is indicated in (4.32) as well, as the reference articulation angle increases with the orientation error and the lateral error to the path. The gain parameter K_{S} is the factor that determines, which of the two goals has more priority. The inclusion of the vehicle speed in the equation works as a regulator to avoid high steering angles at high speed. This increases safety during the experiment.

Table 4.1: Path-following MPC parameters for the f-ADT.

Parameter	**Symbol**	**Value**
Vehicle lengths	L_1, L_2	1.36 m, 3.65 m
Steering constants	T_ω, k_ω	0.5 s, 1
Further parameters	$T_{\mathrm{d}}, T_{\mathrm{s}}, N$	0.5 s, 0.3 s, 20
MPC's constraints	$\lvert\omega_{\mathrm{ref}}\rvert_{\max}$	12 deg/s,
	$\lvert\phi\rvert_{\max}$	42 deg,

4.3 Path-Following Control Simulation and Experimental Evaluation

4.3.1 Simulation and Experimental Setup

The goal of the simulation and the experiment is to compare the performance of the path-following MPC with the Stanley controller as representatives of predictive and classical feedback controllers, respectively. To achieve this comparison, two paths are created and the objective for the controllers is to follow the path with as little lateral error as possible at a constant speed. The paths are designed to match the area that the mining field can provide for testing so that the same scenarios in the simulation can also be performed later in the mining field. In these tests, the vehicle drives with a constant speed of $2\,\mathrm{m\,s^{-1}}$.

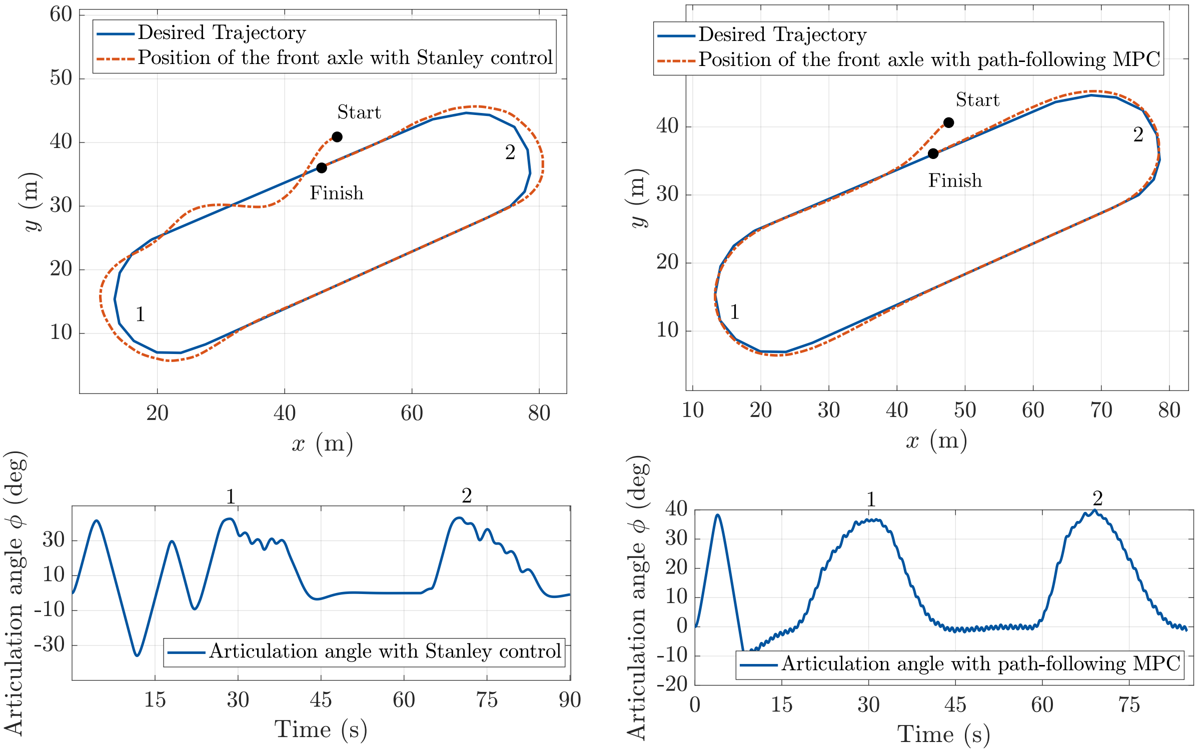


Fig. 4.3: Simulation results for the path-following MPC vs the Stanley controller at track 1. The numbers mark the curves on the driven track and the articulation angle plot.

4.3.2 Simulation and Experimental Results

Figures 4.3 and 4.4 illustrate the simulation results, and Figure 4.5 shows the experimental results. This discussion starts by analyzing the simulations first. Both the MPC and the Stanley controller start off-track so that their behavior in such situations is evaluated and compared. As the Figures show, the Stanley controller has difficulty stabilizing the vehicle over the reference path. The overshoots observed on the track increase in magnitude with increasing speed. Furthermore, the Stanley controller shows a higher lateral error to the reference path when facing a non-zero curvature as observed at all curves. The results also reveal that the vehicle experiences more frequent changes in articulation direction when driven by the Stanley controller compared to the MPC, as indicated by the frequent oscillations in the articulation angle plot.

It is also important to mention that the results of the Stanley controller suffer further at higher speed in terms of the magnitude of the overshoots when stabilizing the vehicle on the track and become unstable at speed above $4\,\mathrm{m\,s^{-1}}$, which is the reason for the limited speed of the tests.

Figure 4.5 shows the experimental results of the same scenario as discussed earlier in this Section. The results are similar to the simulation and prove the experimental comparison

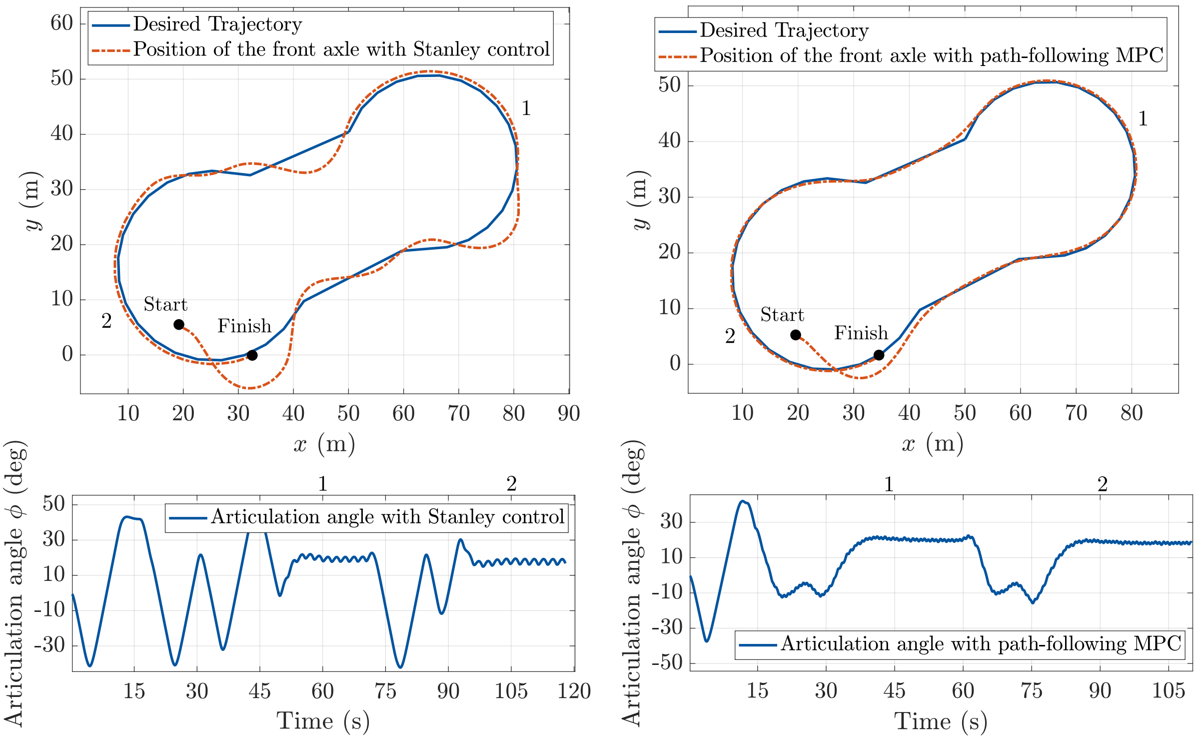


Fig. 4.4: Simulation results for the path-following MPC vs the Stanley controller at track 2. The numbers mark the curves on the driven track and the articulation angle plot.

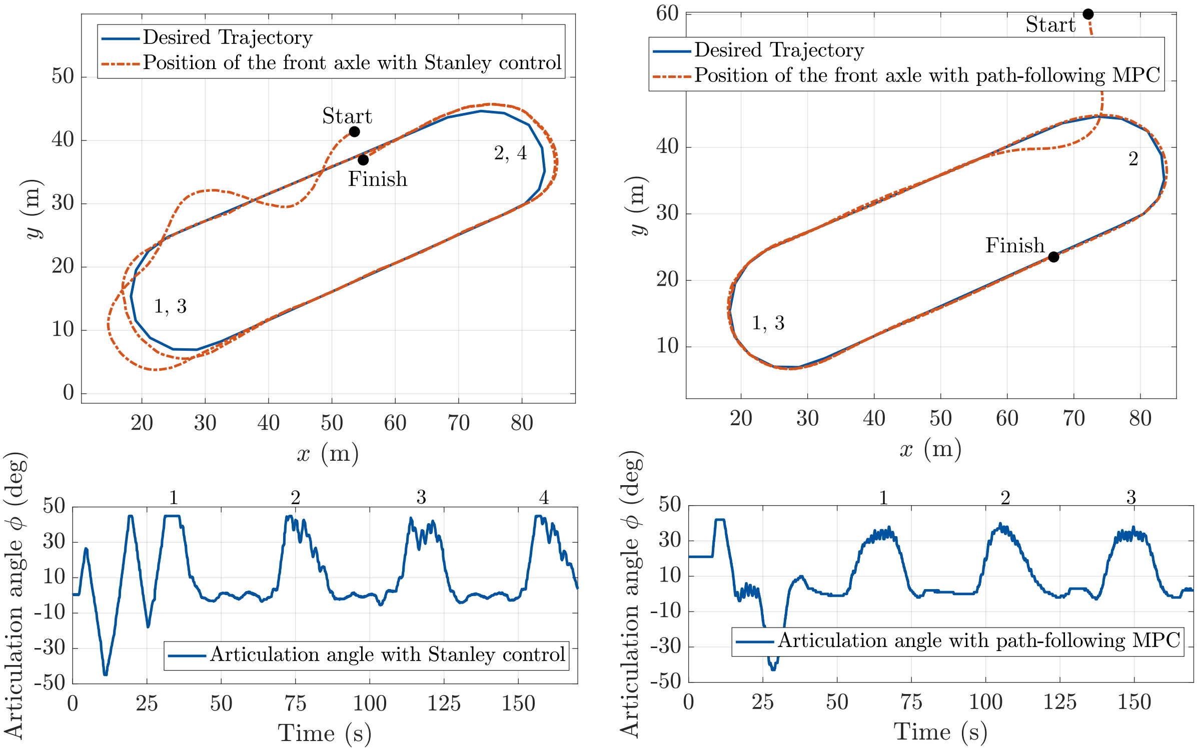


Fig. 4.5: Experimental results for the path-following MPC vs the Stanley controller at track 1. The numbers mark the curves on the driven track and the articulation angle plot.

between the two controllers.

The shortcomings of the Stanley controller are due to its non-predictive nature, which causes the controller to react to the path only after a lateral error has already happened. Furthermore, MPC's ability to count for system phase delays and dead times further improves its results.

As the results indicate, both approaches serve as viable solutions for the autonomous driving of ADTs, and, depending on the acceptable implementation difficulty, computational power on board, the desired operation speed, and the allowed maximal lateral error one can be chosen. However, the path-following MPC delivers noticeably better results and is regarded as the better solution when computational power and implementation difficulty are not an issue. These results also prove the general hypothesis that predictive methods deliver superior results in path tracking at ADTs.

Having presented the results of the simulations and experiments, the use case and limitations of path-following controllers in general can be discussed. For the path-following MPC to function, the future curvatures need to be known. These curvatures are obtained based on the upcoming speed since the position of the points where the curvature is calculated depends on the speed. As a result, changes in the speed after the curvatures have been calculated lead to inaccuracies in the obtained curvatures. This compromises the accuracy of the future path from the MPC's perspective and negates the advantages that a predictive method can provide. As a result, path-following MPCs are generally more suitable for scenarios where the vehicles drive on tracks without frequent variations in the speed profile or without stop-and-go situations.

For other scenarios involving stop-and-go situations, parking scenarios, and frequent speed variations, trajectory-following controllers are recommended. These controllers can control both the speed and the articulation angle simultaneously, automatically taking the effects of variations of the speed into consideration. In the next Chapter, the trajectory-following MPCs are presented.

4.4 Summary

In this Chapter, two path-following controllers were presented, one based on classical feedback control and the other based on an MPC approach. Furthermore, the performance of the controllers in the simulation environment and on a field test was investigated. Finally, a discussion about the strengths and weaknesses of each controller and path-following controllers in general was presented.

5 Trajectory-Following MPCs

In this Chapter, the trajectory-following MPC are presented that can handle the entire operational routine of an ADT defined in Section 1.2.4. To achieve this goal, this Chapter first presents vehicle models that are required for different operation scenarios. Afterwards, an MPC is developed for each model. Finally, the results of experiments conducted in a surface mine with the f-ADT and c-ADT carrying out a routine transportation scenario are presented at the end of this Chapter.

5.1 Kinematic Model Library for the Trajectory-Following MPCs

5.1.1 ADT Kinematic Model for Driving Forward

For this operation scenario, driving forward for the f-ADT, the baseline kinematic model from Section 3.5.1 is employed together with the actuation identification equations for speed (3.2) and articulation angle rate (3.1). This yields the following model:

$$\dot{x}_1 = v_1 \cos\psi_1, \tag{5.1}$$

$$\dot{y}_1 = v_1 \sin\psi_1, \tag{5.2}$$

$$\dot{\psi}_1 = \frac{\sin\phi}{L_2 + L_1\cos\phi} v_1 + \frac{L_2}{L_2 + L_1\cos\phi}\omega, \tag{5.3}$$

$$\dot{\phi} = \omega, \tag{5.4}$$

$$\dot{\omega} = (-\omega + k_\omega \omega_{\mathrm{ref}})/T_\omega, \tag{5.5}$$

$$\dot{v}_1 = (-v_1 + k_v v_{\mathrm{ref}})/T_v. \tag{5.6}$$

In this system, (x_1, y_1) is the position vector of the front axle in inertial coordinates, and v_1 is the speed at the front axle. Variable ϕ is the articulation angle, while ψ_1 is the front yaw angle. Additionally, ω is the articulation angle rate, and L_1 and L_2 denote the distance from the hinge to the front and the rear axle, respectively. Parameters (k_ω, T_ω) and (k_v, T_v) are the articulation angle rate and the speed gain and time constant, while

ω_{ref} and v_{ref} denote the reference value for articulation angle rate and speed, respectively. Figure 5.1 illustrating this setup is repeated here from Section 3.5.1 for ease of access.

The state vector of this model, which is designed with front axle referencing ($\square^{\mathrm{FAR}}$) for the f-ADT ($\square_{\mathrm{f}}$), is given as $\boldsymbol{x}_{\mathrm{f}}^{\mathrm{FAR}} = [x_1, y_1, \psi_1, \phi, v_1]^\top$. The reference articulation angle rate ω_{ref} and the reference speed v_{ref} define the control input vector $\boldsymbol{u}_{\mathrm{f}}^{\mathrm{FAR}} = [v_{\mathrm{ref}}, \omega_{\mathrm{ref}}]$. This model is called $\mathrm{FAR}_{\mathrm{f}}$.

5.1.2 ADT Kinematic Model for Driving Backwards

When driving forward, the front axle of the vehicle, which is the reference point in the models so far, moves in the same direction as the steering. However, when driving backwards, the system shows non-minimum phase behavior at the front axle's position, and steering in one direction causes the front axle to move in the opposite direction for a period of time before moving in the targeted direction. The duration of the opposite movement T_{o} and the displacement of the front axle in the opposite direction increase at lower speed or when the lengths of the front and rear bodies of the vehicle differ more strongly. Fig. 5.2 shows this effect in a simulation when the vehicle starts with $\psi_1 = \phi = 0\,\mathrm{deg}$, and the articulation angle is set to $\phi = 40\,\mathrm{deg}$ at $t = 5\,\mathrm{s}$. When T_{o} becomes longer than the prediction horizon of an MPC, the MPC mistakenly expects the ADT to generally move in the opposite direction of the steering. This leads to infeasibility in the MPC. Even if the prediction horizon of the MPC is longer than T_{o}, the accuracy of the control will be lost until the end of T_{o} due to linearization errors.

To avoid using a nonlinear MPC or an unnecessarily long prediction horizon, this Chapter proposes to eliminate this problem by switching the reference point of the vehicle to the rear axle. In order to achieve this and to create a model that depends only on the rear speed as the speed variable, the relationship between the front and the rear speed and

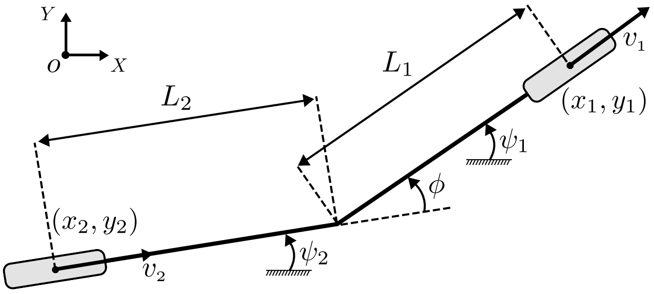


Fig. 5.1: Kinematic configuration of an ADT from [M2] ©2022 IEEE.

yaw angles must be formulated, as can be shown based on the kinematics

$$v_1 = v_2 \cos\phi + L_2 \dot{\psi}_2 \sin\phi, \tag{5.7}$$

$$\dot{\psi}_1 = \dot{\psi}_2 + \omega, \tag{5.8}$$

where v_2 is the speed of the rear axle, and ψ_2 is the rear yaw angle. The reference point of the ADT can be moved to the rear axle by inserting (5.7) and (5.8) in (5.3). The state vector of the resulting model with rear-axle referencing (RAR) is given as $\boldsymbol{x}_\mathrm{f}^\mathrm{RAR} = [x_2, y_2, \psi_2, \phi, \omega, v_2]^\top$, where (x_2, y_2) is the position of the rear axle in the inertial coordinates. This model is called the $\mathrm{RAR_f}$ model. The control input vector is $\boldsymbol{u}_\mathrm{f}^\mathrm{RAR} = [v_\mathrm{ref}, \omega_\mathrm{ref}]$, and the state equations are given as:

$$\dot{x}_2 = v_2 \cos\psi_2, \tag{5.9}$$

$$\dot{y}_2 = v_2 \sin\psi_2, \tag{5.10}$$

$$\dot{\psi}_2 = \frac{\sin\phi}{L_2\cos\phi + L_1} v_2 - \frac{l_1\omega}{L_2\cos\phi + L_1}\omega, \tag{5.3}$$

$$\dot{\phi} = \omega, \tag{5.4}$$

$$\dot{\omega} = (-\omega + k_\omega \omega_\mathrm{ref})/T_\omega, \tag{5.5}$$

$$\dot{v}_2 = (-v_2 + k_v v_\mathrm{ref})/T_v. \tag{5.6}$$

5.1.3 Kinematic Models for the Compact ADT

Compact ADTs (c-ADT) are used in construction sites where an f-ADT is not suitable due to its larger size. c-ADTs have a faster dynamic in steering since considerably less torque is required to steer. Due to these faster dynamics, using the articulation rate as the control variable is difficult and unnecessary since it requires an extra state in the vehicle model and an additional intermediate control loop. To establish a suitable

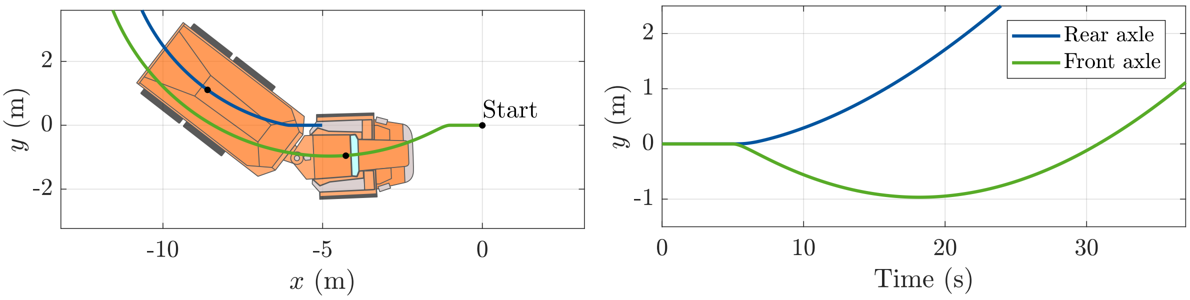


Fig. 5.2: Non-minimum phase behavior of the FAR model compared to the RAR model when driving backwards. Initial yaw angle $\psi_1 = \phi = 0\,\mathrm{deg}$. ϕ is set to $40\,\mathrm{deg}$ at $t = 5\,\mathrm{s}$. $v_1 = 0.5\,\mathrm{m\,s^{-1}}$.

dedicated model for c-ADTs, the steering dynamic is directly modeled using a first-order low-pass element. This can be given as

$$\frac{\phi}{\phi_{\mathrm{ref}}} = \frac{k_\phi}{1 + T_\phi s}, \text{ or as } \dot{\phi} = (-\phi + k_\phi \phi_{\mathrm{ref}})/T_\phi, \tag{5.7}$$

where ϕ_{ref} is the desired articulation angle, k_ϕ is the gain, and T_ϕ is the time constant of the steering. When ω in the model from 5.1.1 is replaced by (5.7), and the model is expanded with the speed first-order low-pass element, the kinematic model of a c-ADT for driving forward can be achieved. The state vector of this model, which is designed with front axle referencing ($\square^{\mathrm{FAR}}$) for the c-ADT ($\square_{\mathrm{c}}$), is given as $\boldsymbol{x}_{\mathrm{c}}^{\mathrm{FAR}} = [x_1, y_1, \psi_1, \phi, v_1]^\top$. The desired articulation angle ϕ_{ref} and the desired speed v_{ref} define the control input vector $\boldsymbol{u}_{\mathrm{c}}^{\mathrm{FAR}} = [v_{\mathrm{ref}}, \phi_{\mathrm{ref}}]$. This model is called the $\mathrm{FAR}_{\mathrm{c}}$ model, and its state equations are given as:

$$\begin{aligned}
\dot{x}_1 &= v_1 \cos\psi_1, && (5.8)\\
\dot{y}_1 &= v_1 \sin\psi_1, && (5.9)\\
\dot{\psi}_1 &= \frac{\sin\phi}{L_2 + L_1\cos\phi} v_1 + \frac{L_2(-\phi + k_\phi\phi_{\mathrm{ref}})}{T_\phi(L_2 + L_1\cos\phi)}, && (5.10)\\
\dot{\phi} &= (-\phi + k_\phi\phi_{\mathrm{ref}})/T_\phi, && (5.11)\\
\dot{v}_1 &= (-v_1 + k_v v_{\mathrm{ref}})/T_v. && (5.12)
\end{aligned}$$

Similar to f-ADTs, the kinematic ADT model for driving backwards must be established for c-ADTs as well. Following the same steps as in Section 5.1.2 yields the kinematic model of a c-ADT for driving backwards. The state vector and the control input vector of this model, which is called the $\mathrm{RAR}_{\mathrm{c}}$ model, are given as $\boldsymbol{x}_{\mathrm{c}}^{\mathrm{RAR}} = [x_2, y_2, \psi_2, \phi, v_2]^\top$ and $\boldsymbol{u}_{\mathrm{c}}^{\mathrm{RAR}} = [v_{\mathrm{ref}}, \phi_{\mathrm{ref}}]$, respectively. The state equations of the ADT are given as:

$$\begin{aligned}
\dot{x}_2 &= v_2 \cos\psi_2, && (5.13)\\
\dot{y}_2 &= v_2 \sin\psi_2, && (5.14)\\
\dot{\psi}_2 &= \frac{\sin\phi}{L_2\cos\phi + L_1} v_2 + \frac{L_2(-\phi + k_\phi\phi_{\mathrm{ref}})}{T_\phi(L_2\cos\phi + L_1)}, && (5.3)\\
\dot{\phi} &= (-\phi + k_\phi\phi_{\mathrm{ref}})/T_\phi, && (5.4)\\
\dot{v}_2 &= (-v_2 + k_v v_{\mathrm{ref}})/T_v. && (5.5)
\end{aligned}$$

5.2 Trajectory-Following MPC Experimental Results

In order to find the optimal control input that minimizes the distance between the vehicle state and a given reference trajectory $\boldsymbol{x}_{\mathrm{ref}}$, an MPC is developed for each of the four presented ADT models. The setup of the MPCs, including the cost function, linearization, discretization, constraints, and dead time compensation is the same as the setup in Chapter 4.

5.2.1 Compact ADT

5.2.1.1 Experimental Setup

The objective of the experiment is for the c-ADT to follow a given trajectory forward with the front axle as the reference point, which means by using the $\mathrm{FAR_c}$ model. After reaching the end of the trajectory, the vehicle stops and drives the same trajectory backwards. This part of the experiment is done once with the $\mathrm{FAR_c}$ and once with the $\mathrm{RAR_c}$ model to compare the functionality of the models for the task. Since the testing ground is an unpaved mining field, it is decided to limit the speed of the vehicle to $2\,\mathrm{m\,s^{-1}}$. Higher speed proved to be damaging to the vehicle and to the equipment. Table 5.1 lists the parameters of the MPCs.

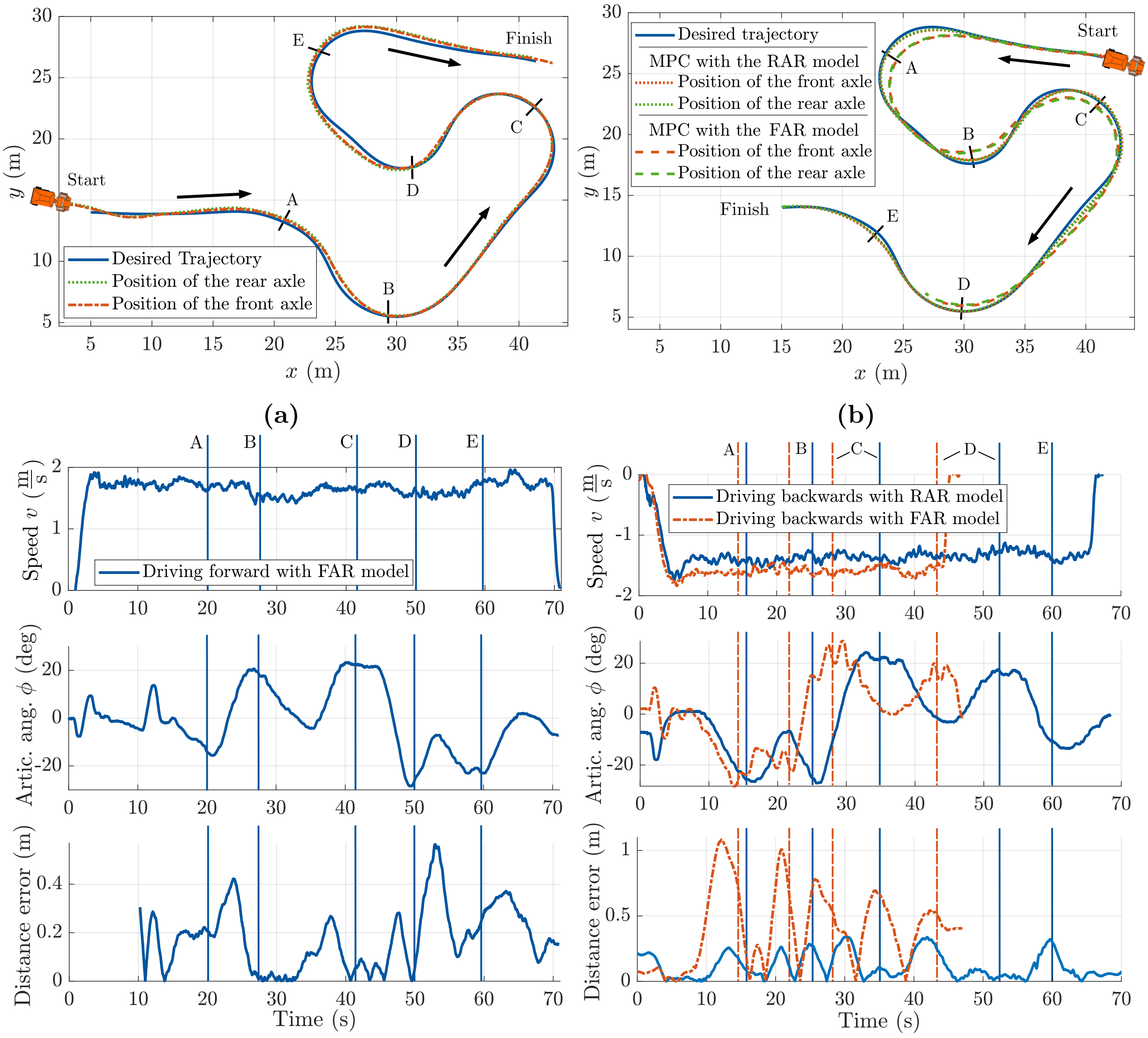
The trajectory-following MPCs run on an industrial PC (IPC) at 20 Hz. The IPC is connected to the vehicle CAN bus, through which the MPCs send the control inputs to the vehicle. The current state vector of the vehicle $\boldsymbol{x}$, which is required to close the feedback loop, is formed with the information on the CAN bus and the data from the localization module.

5.2.1.2 Experimental Results

Fig. 5.3 shows the results of the trajectory-following MPCs for the c-ADT. Figures 5.3a and 5.3b show the plot of the reference and the driven trajectory in the experiments, and Figures 5.3c and 5.3d the vehicles speed v, articulation angle ϕ, and the distance error

Table 5.1: c-ADT MPC parameters.

Parameter	Symbol	Value
Vehicle lengths	L_1, L_2	$0.80\,\mathrm{m}$, $0.84\,\mathrm{m}$
Speed constants	T_v, k_v	$1.25\,\mathrm{s}$, 1
Steering constants	T_ϕ, k_ϕ	$0.67\,\mathrm{s}$, 1
Weight matrix	$\boldsymbol{Q}$	$\mathrm{diag}([100, 100, 0, 0, 0])$
Weight matrix	$\boldsymbol{R}$	$\mathrm{diag}([30, 20])$
Further parameters	$T_\mathrm{d}, T_\mathrm{s}, N$	$0.5\,\mathrm{s}$, $0.3\,\mathrm{s}$, 20
MPC's input constraints	$\lvert\phi_\mathrm{ref}\rvert_\mathrm{max}$,$\lvert v_\mathrm{ref}\rvert_\mathrm{max}$	30 deg, $2\,\mathrm{m\,s^{-1}}$



(a) **(b)**

(c) **(d)**

Fig. 5.3: c-ADT experimental results. (a) Reference and actual trajectory plots for the c-ADT in driving forward with the MPC that uses the $\mathrm{FAR_c}$ model. (b) Reference and actual trajectory plots for the c-ADT in driving backwards with $\mathrm{FAR_c}$ and $\mathrm{RAR_c}$ models. (c) and (d) Speed, articulation angle, and distance error during the experiments of driving forward and backwards, respectively.

between the vehicle's reference axle and the reference trajectory. Table 5.2 summarizes the results of the experiments. As Fig. 5.3a shows, the vehicle's start position is not at the start position of the trajectory. Therefore, the MPC drives the vehicle to the start point and follows the trajectory afterward. The vehicle exhibits no noticeable oscillation about the trajectory other than at the beginning, where the MPC tries to reach the reference trajectory and causes an overshoot. As mentioned in the state-of-the-art discussion in

Section 1.2.4, it is important that the maximum lateral error stays within the required limit when close to the destination or the finish point and to keep the MAE within the acceptable limits during the whole trajectory. The MPC manages to achieve a maximum lateral error of 22.9 cm in the last 5 m of the trajectory and an MAE of 17.6 cm. The overall maximum lateral error is 56.3 cm.

Fig. 5.3b shows the trajectory plots for driving backwards for the MPCs with the $\mathrm{RAR_c}$ and $\mathrm{FAR_c}$ models with the reference point on the rear and the front axle, respectively. The latter had to be stopped before finishing the trajectory shortly after curve D since the vehicle's deviation from the reference trajectory appeared to indicate unpredictable behavior of the vehicle. This situation happened repeatedly in multiple attempts. As it is shown in Figures 5.3b and 5.3d, the MPC with the $\mathrm{RAR_c}$ model performs better in terms of the lateral error. This MPC achieves an overall MAE of 11.5 cm and a maximum lateral error at the finishing 5 m of the trajectory of 10.3 cm. The overall maximum lateral error is 33.9 cm. In contrast, the MPC with the $\mathrm{FAR_c}$ model achieved an overall MAE of 40.1 cm with a maximum lateral error of 1.1 m. From these results, it is evident that using the $\mathrm{RAR_c}$ model in an MPC for driving backwards substantially improves the results in terms of lateral error. This improvement is achieved by using the $\mathrm{RAR_c}$ model since it does not show the non-minimum-phase behavior at its reference point when driving backwards.

Table 5.2: c-ADT MPC results.
In this table, (‡) marks this work's contributions and (†) marks state-of-the-art approaches.

Evaluation description	**Driving scenario, used model**	**Value, (max. allowed value)**
Overall MAE	‡Forward, $\mathrm{FAR_c}$	17.6 cm, (40 cm)
	‡Backwards, $\mathrm{RAR_c}$	11.5 cm, (40 cm)
	†Backwards, $\mathrm{FAR_c}$	40.1 cm, (40 cm)
Max. lat. error in the last 5 m of the trajectory	‡Forward, $\mathrm{FAR_c}$	22.9 cm, (60 cm)
	‡Backwards, $\mathrm{RAR_c}$	10.3 cm, (60 cm)
	†Backwards, $\mathrm{FAR_c}$	54 cm, (60 cm)
Overall max. lat. error	‡Forward, $\mathrm{FAR_c}$	56.3 cm
	‡Backwards, $\mathrm{RAR_c}$	33.9 cm
	†Backwards, $\mathrm{FAR_c}$	1.1 m

5.2.2 Full-Sized ADT

5.2.2.1 Experimental Setup

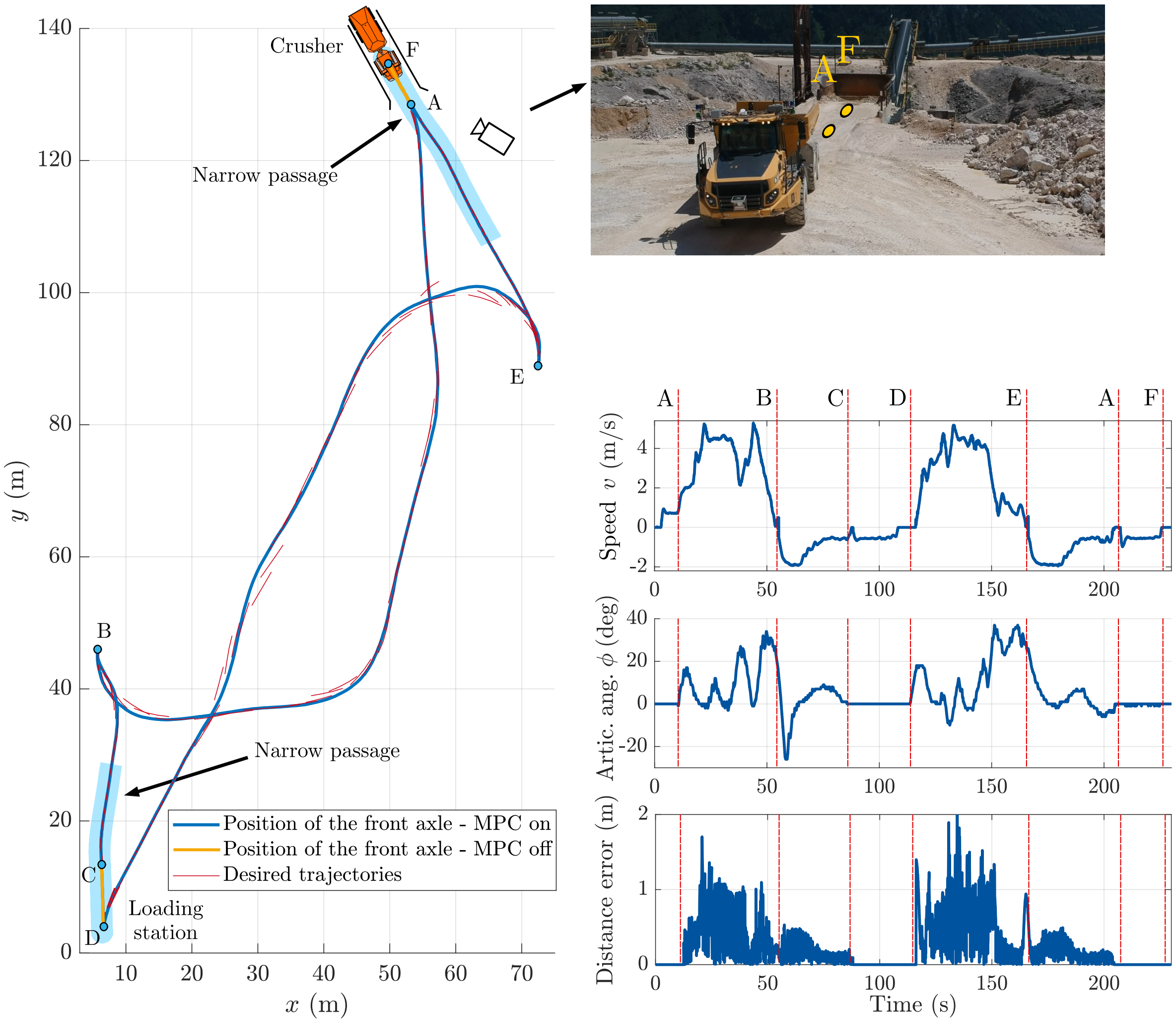
In this experiment, the objective is to fulfill the routine task of an f-ADT in a surface mine. This includes driving to the loading area, reversing into the loading position, driving to the crusher/unloading area, and finally reversing into the crusher. With trajectory plots shown in Fig. 5.4a, the ADT starts at the crusher A with an empty bed and drives forward with the MPC with the $\mathrm{FAR_f}$ model to the vicinity of the loading area B. At this point, the vehicle reverses into the loading station C using the $\mathrm{RAR_f}$ model. The planner reduces the reference trajectory during this phase to achieve more accuracy. When the vehicle has reached the threshold C, the MPC stops, and if the vehicle has reached the correct yaw angle, the vehicle drives backwards with an open-loop controller with constant speed and zero steering angle to reach the loading station D. This is necessary due to safety reasons. After the loading process is completed, the MPC drives the ADT back to the crusher area (E) and then reverses into the crusher A. Here, again, the MPC stops when the vehicle reaches threshold A and is fully inside the crusher hallway, and if the correct yaw angle is reached, it drives the last meters backwards with the open-loop controller with constant speed and zero steering to reach the final position F.

In the f-ADT, the MPCs run also at 20 Hz on a robot operating system (ROS) enabled laptop, which is connected to the vehicle via CAN bus. The information on the CAN bus and the localization data from a localization module form the state vector of the vehicle $\boldsymbol{x}$. The localization module and the ROS/CAN interface are implemented by xtonomy GmbH. The vehicle is a Bell B30E, and the testing ground is in an unpaved and bumpy mining area. Table 5.3 lists the MPC parameters.

Unlike the experiment with the c-ADT, no results could be obtained for the reversing part

Table 5.3: f-ADT MPC parameters.

Parameter	Symbol	Value
Vehicle lengths	L_1, L_2	1.36 m, 3.65 m
Speed constants	T_v, k_v	1.25 s, 1
Steering constants	T_ω, k_ω	0.5 s, 1
Weight matrix	$\boldsymbol{Q}$	$\mathrm{diag}([100, 100, 0, 0, 0, 0])$
Weight matrix	$\boldsymbol{R}$	$\mathrm{diag}([1, 1])$
Further parameters	$T_\mathrm{d}, T_\mathrm{s}, N$	0.5 s, 0.3 s, 20
MPC's state constraints	$\lvert\phi\rvert_\mathrm{max}$	42 deg
MPC's input constraints	$\lvert\omega_\mathrm{ref}\rvert_\mathrm{max}, \lvert v_\mathrm{ref}\rvert_\mathrm{max}$	12 deg/s, 8 m s^{-1}



(a) **(b)**

Fig. 5.4: F-ADT experimental results. (a) Reference and actual trajectory plots for the f-ADT carrying out a cycle of an ADT's routine operation in a mining field. ADT width 2.99 m, crusher width 4.99 m. (b) Speed, articulation angle, and distance error during the experiment.

of this experiment with the MPC with the $\mathrm{RAR_f}$ model due to immediate infeasibility in the MPC.

The trajectory planning for this vehicle is a black box developed by xtonomy GmbH, which generates a new trajectory at 1 to 5 Hz. As a result, it is not possible to have a static reference trajectory to compare with the driven path. To make a meaningful visual comparison possible, the relevant segments of each generated reference trajectory have been selectively plotted in Fig. 5.4a in red.

Table 5.4: f-ADT MPC results.
In this table, (‡) marks this work's contributions and (†) marks state-of-the-art approaches.

Evaluation description	**Driving scenario, used model**	**Value, (max. allowed value)**
Overall MAE	‡Forward, FAR_f	49.4 cm, (60 cm)
	‡Backwards, RAR_f	15.9 cm, (60 cm)
	†Backwards, FAR_f	Infeasible MPC
Max. lat. error at passages A, C	‡Backwards, RAR_f	26.7 cm, 21.3 cm (50 cm)
	†Backwards, FAR_f	Infeasible MPC
Overall max. lat. error	‡Forward, FAR_f	2.1 m
	‡Backwards, RAR_f	52.4 cm
	†Backwards, FAR_f	Infeasible MPC

5.2.2.2 Experimental Results

Fig. 5.4a shows the reference trajectories from the trajectory planner and the actual driven path. A photograph showing the vehicle entering the crusher backwards during the test is also shown here. Fig. 5.4b shows the speed, articulation angle, and lateral distance between the vehicle and the trajectory. Table 5.4 summarizes the experimental results.

As Fig. 5.4a shows, the vehicle's driven path does not contain noticeable oscillations. Furthermore, as Fig. 5.4b shows, the vehicle achieves higher speed than the required $4\,\text{m}\,\text{s}^{-1}$ during the experiment over 30 s of the experiment with a maximum speed of $5.2\,\text{m}\,\text{s}^{-1}$.

A direct comparison with state-of-the-art methods, specifically employing the FAR_f model when driving backwards, was hindered by the infeasibility of the resulting MPC a few seconds into the experiment. This is due to the non-minimum phase behavior of the model when driving backwards, as previously discussed.

In this experiment, the overall MAE in the lateral error is 34.6 cm, and the MAE in driving forward and backwards is 49.4 cm and 15.9 cm, respectively. The maximum lateral error is 2.1 m when driving forward and 52.4 cm when driving backwards. At the narrow passages at the crusher way A and the loading way C, the maximum error is 26.7 cm and 21.3 cm, respectively. This is below the expected 50 cm that was required to ensure the safety of the vehicle and equipment.

Based on these results, it can be concluded that the control concepts have successfully fulfilled the initial expectations and can be implemented in real-world scenarios.

Based on discussions with mine operators, the margin between achieved and allowed accuracy should be over 20%. However, this condition is not fulfilled for the MAE in the f-ADT. This may be related to the reference trajectory, which is planned without taking the physical limits of the vehicles into account. The implementation of a model-based planner could therefore improve the results.

Another source of error in these experiments is model uncertainties, which is due to sideslip will be discussed thoroughly in the following Chapters.

5.3 Summary

In this Chapter, the development and implementation of models and MPCs for autonomous forward and backwards driving for both compact ADTs with faster steering dynamics and full-sized ADTs were presented. By moving the reference point of the model to the rear axle, the issue with the non-minimum phase behavior of the front-axle referencing when driving backwards was successfully addressed, and sufficient accuracy for driving backwards in narrow passages, such as crushers was achieved. Furthermore, by using system identification techniques, it became possible to directly use articulation angle as the control variable in the model of compact ADTs and to benefit from their faster steering dynamics. The experimental validation of the MPCs shows that the MPCs met the specified requirements and show potential to be used in real-world applications.

6 Sideslip Angle and Parameter Estimation

6.1 Overview

The previous Chapter presented different trajectory-following MPCs for the entire cycle of the routine operation of an ADT. This included driving forward and backwards for two major ADT sizes. As also discussed in detail in the previous Chapter, the presented MPCs did increase the operation speed of the ADTs by 3 to $4\,\mathrm{m\,s^{-1}}$ in some cases compared to the state-of-the-art methods. This was achieved by employing the different variation of the ADT kinematic model in an MPC.

As discussed and shown in Chapter 3, kinematic models achieve lower accuracies especially when cornering, which is, among other reasons, primarily due to the assumption of a sideslip-free motion of the vehicle. This assumption means to assume that the vehicle moves in the same direction it is pointed at, which is not true especially in curves [68].

The goal in this and the next Chapter is to further increase the operation speed of the ADTs or to further improve the performance of the controllers at current speeds. This can be achieved by employing models in the MPC that consider the sideslip angle, which is neglected in the approaches presented so far, so that the MPC can compensate for it.

To accomplish this goal, there are a variety of options and methods that can be applied to the ADTs. These options are investigated in detail in the next Chapter. Nevertheless, regardless of the exact method used, the sideslip angle's current value is critical for the functionality of the resulting controller to close the feedback loop. However, measuring the sideslip angle directly requires sensors that are not available in our and other commercial vehicles (see Chapter 1.2.6). Moreover, it is not possible to calculate the sideslip angle solely based on the odometry data from the localization module, as the noise in these signals renders the resulting sideslip angle practically unusable. Due to these reasons, this Chapter presents state estimation algorithms for sideslip angle estimation in ADTs, exploring both the state-of-the-art methods and also the new approaches.

The sideslip angle at a point of interest on the vehicle body has different values depending on where this point is located. Since the reference point for the control is on the front axle, the point of interest is placed where the value of the sideslip angle should be estimated, on the front axle. For the filters in this Chapter, the data from the localization module, an inertial measurement unit (IMU), and the vehicle states from the CAN bus are assumed to be knwon.

Another topic this Chapter aims to investigate is parameter estimation in ADTs. As mentioned in Chapter 3, the inclusion of actuation behavior models in the vehicle model is crucial for MPC performance. As a result, in Chapter 3, this behavior was modeled in the form of first-order low-pass elements with the help of an offline identification process for the parametrization. In this Chapter, a method for parameter estimation is presented to estimate the value of actuation identification or other important parameters online. By establishing this online estimation algorithm for these parameters, the need for time-consuming offline identification can be eliminated, and the algorithms are enabled to work on new ADTs without a priori identified actuation behavior.

The layout in this Chapter starts by presenting a short introduction to joint extended Kalman filters, which is used extensively in this work. Then, three state estimation methods are introduced, which include one state-of-the-art approach and two new methods. The third state estimation filter introduces the capability of parameter estimation and can estimate certain vehicle parameters in addition to the state variables. At the end of this Chapter, the evaluation of the state and parameter estimation in the validated simulation environment is presented.

The contents of this chapter have only been established and tested for the f-ADT, as the c-ADT's operating speed is not high enough for the resulting sideslip angle to have a considerable effect on the controller's performance.

The content of this Chapter corresponds to the publication [M1].

6.2 Joint Extended Kalman Filter

The joint extended Kalman filter (JEKF) is an extension of the well-established extended Kalman filter (EKF) that can additionally incorporate parameter estimation in the estimation process. This can be achieved by first adding the parameter vector, which consists of the parameters that are desired to be estimated, to the system's state vector. Then, a set of differential equations that describe the newly added states are added to the system model. One of the methods that can be incorporated for modeling the newly added states is to employ the Brownian motion, which is also known as random walk. This means to assume that the derivative of the parameter is equal to zero [74]. Modeling the parameter as random walk is suitable especially when the parameter does not exhibit frequent or high-frequency changes in its value.

In order to set up the equations for a JEKF, the following time-discrete, nonlinear system is assumed:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t), \tag{6.1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C} \cdot \boldsymbol{x}(t) + \boldsymbol{v}(t). \tag{6.2}$$

Here, the nonlinear system equations are presented by $\boldsymbol{f}(\cdot)$, the system state, input, and measurement vectors are presented by $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, and $\boldsymbol{y}(t)$, and the output system matrix by $\boldsymbol{C}$. The process noise is marked by $\boldsymbol{w}(t)$, and the measurement noise by $\boldsymbol{v}(t)$ with $\boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{Q}), \boldsymbol{v}_t \sim \mathcal{N}(0, \boldsymbol{R})$. The process noise $\boldsymbol{w}(t)$ and measurement noise $\boldsymbol{v}(t)$ are both zero-mean and normally distributed with $\boldsymbol{Q}$ and $\boldsymbol{R}$ as their covariance matrices [44].

The JEKF requires linearized and time-discrete system matrices for the prediction and correction steps of the estimation. Similar to Section 4.1.2, by employing the Jacobian matrix for linearization and matrix exponential with the sampling time T_s for the discretization, the linear, time-discrete system matrices $\boldsymbol{A}_\mathrm{d}(k)$, $\boldsymbol{B}_\mathrm{d}(k)$, and $\boldsymbol{C}_\mathrm{d}$ can be calculated. Further information about the detailed implementation of a Kalman filter can be found in [17].

6.3 Sideslip Angle Estimation

6.3.1 Approach One: State of the Art

The first approach is derived directly from [74], where a Kalman filter is used that employs the kinematic-slip model from Section 3.5.2.

This model, which is also used for the estimation in the next Section, includes both the front and rear axle's sideslip angles in its equations. However, in this and the next Section, the rear sideslip angle is assumed to be zero. This assumption simplifies the estimation process by removing an extra variable.

Assuming that the rear axle experiences no sideslip does not necessarily pose an issue for the estimation since the rear sideslip angle is negligible. This is because the heavier weight and the involvement of four wheels in the rear axle, as opposed to two in the front, increase the vehicle-ground friction, contributing to the smaller and negligible sideslip angle of the rear axle. Furthermore, since the reference point of the control is on the front axle, only the front sideslip angle is of importance.

The front sideslip angle α is incorporated as a parameter without a dedicated differential equation in the model. In order to estimate a variable using a Kalman filter, it is necessary to have a differential equation for that variable. Accordingly, the sideslip angle must be incorporated in the state vector of the vehicle model in order to estimate its value. As mentioned earlier in the state-of-the-art discussions in Section 1.2.6, in this approach, the sideslip angle state in the equations is modeled as random walk.

The aforementioned adjustments result in the following system of equations for the estimation, and Figure 6.1 illustrates the kinematic configuration of an ADT with front sideslip.

$$\dot{x}_1 = v_1 \cos(\psi_1 + \alpha), \tag{6.3}$$

$$\dot{y}_1 = v_1 \sin(\psi_1 + \alpha), \tag{6.4}$$

$$\dot{\psi}_1 = \frac{\sin(\phi + \alpha)}{L_2 + L_1 \cos(\phi)} v_1 + \frac{L_2}{L_2 + L_1 \cos(\phi)} \omega, \tag{6.5}$$

$$\dot{\phi} = \omega \quad (\phi = \psi_1 - \psi_2), \tag{6.6}$$

$$\dot{\alpha} = 0. \tag{6.7}$$

In this system, v_1 and v_2 are the front and rear axle's speed, respectively. Vectors (x_1, y_1) and (x_2, y_2) are the position vector of the front and rear axles of the vehicle in inertial coordinates. The articulation angle is denoted by ϕ, while ψ_1 and ψ_2 are the front and the rear yaw angle, respectively. Additionally, ω is the articulation angle rate, α is the front sideslip angle, and L_1 and L_2 denote the distance from the hinge to the front and the rear axle, respectively. The measurement, the input, and the state vector of this system used in the estimation are shown below:

$$\begin{aligned} \boldsymbol{x}_{\text{se-base}} &= [x_1, y_1, \psi_1, \phi, \alpha]^T, \ \boldsymbol{u}_{\text{se-base}} = [\omega, v_1]^T, \\ \boldsymbol{y}_{\text{se-base}} &= [x_1, y_1, \psi_1, \phi]^T, \end{aligned} \tag{6.8}$$

where se-base refers to state estimation with the baseline version of the kinematic-slip model.

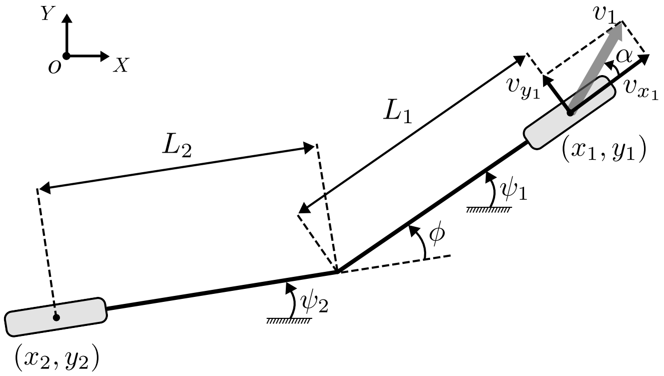


Fig. 6.1: Kinematic-slip configuration of an ADT with front sideslip. Adapted from [M2] ©2022 IEEE.

To use this model in the estimation, the linearized time-discrete system matrices must be established. As previously presented in Section 4.1.2, by employing the Jacobian matrix for linearization and matrix exponential with the sampling time T_s for discretization, the linear, time-discrete system matrices will be achieved.

This marks the first Kalman filter, which is referred to in the following as $\mathrm{EKF}_{\text{se-base}}$ as in state estimation using the baseline version of the kinematic-slip model.

6.3.2 Approach Two: Using the Extended Kinematic-Slip Model

As discussed before in Section 1.2.6, in this second approach, it is intended to use the same kinematic model as in the previous Section but add system knowledge to the equation of the sideslip angle. For model knowledge, the sideslip angle differential equation from [67] is employed. In the following, a step-by-step proof for this equation is presented.

To begin, the equation of the sideslip angle is written for a specified point of interest on the vehicle:

$$\alpha = \arctan \frac{v_{y_1}}{v_{x_1}}, \tag{6.9}$$

where v_{y_1} and v_{x_1} are the lateral and longitudinal elements of the speed of the moving point, respectively. Assuming that the sideslip angle is small, using the small-angle approximation yields:

$$\alpha = \frac{v_{y_1}}{v_{x_1}}. \tag{6.10}$$

Now, the derivative of the sideslip angle can be calculated as follows:

$$\dot{\alpha} = \frac{\dot{v}_{y_1} v_{x_1} - \dot{v}_{x_1} v_{y_1}}{v_{x_1}^2}. \tag{6.11}$$

By employing the assumption of a small sideslip angle the small-angle approximation again, the following equation for the derivative of the sideslip angle can be established:

$$\dot{\alpha} = \frac{\dot{v}_{y_1}}{v_{x_1}}. \tag{6.12}$$

On the other hand, the following equations describing the motion of a moving point in space are known:

$$a_{x_1} = \dot{v}_{x_1} - v_{y_1} \dot{\psi}, \tag{6.13}$$

$$a_{y_1} = \dot{v}_{y_1} + v_{x_1} \dot{\psi}, \tag{6.14}$$

where a_{x_1} and a_{y_1} are the longitudinal and lateral accelerations at the moving point, respectively, while $\dot{\psi}$ is the yaw angle rate. By using (6.14), the equation for the sideslip angle can be re-established:

$$\dot{\alpha} = \frac{a_{y_1} - v_{x_1}\dot{\psi}_1}{v_{x_1}}. \tag{6.15}$$

Equation (6.15) is a well-established equation for sideslip angle [67] and is widely used in estimation approaches as the yaw angle rate and lateral acceleration ($\dot{\psi}_1$ and a_{y_1}) are variables that an IMU can deliver. Using this equation helps integrate model knowledge in the kinematic model for the sideslip angle. The system model can now be formulated again as follows:

$$\begin{aligned}
\dot{x}_1 &= v_1 \cos(\psi_1 + \alpha), && (6.16)\\
\dot{y}_1 &= v_1 \sin(\psi_1 + \alpha), && (6.17)\\
\dot{\psi}_1 &= \frac{\sin(\phi + \alpha)}{L_2 + L_1\cos(\phi)} v_1 + \frac{L_2}{L_2 + L_1\cos(\phi)}\omega, && (6.18)\\
\dot{\phi} &= \omega \quad (\phi = \psi_1 - \psi_2), && (6.19)\\
\dot{\alpha} &= \frac{a_{y_1} - v_{x_1}\dot{\psi}_1}{v_{x_1}}. && (6.20)
\end{aligned}$$

The measurement, input, and state vector are shown below for the state estimation using the extended kinematic-slip model se-ext:

$$\begin{aligned}
\boldsymbol{x}_{\text{se-ext}} &= [x_1, y_1, \psi_1, \phi, \alpha]^T,\ \boldsymbol{u}_{\text{se-ext}} = [\omega, v_1, a_{y_1}]^T,\\
\boldsymbol{y}_{\text{se-ext}} &= [x_1, y_1, \psi_1, \phi]^T.
\end{aligned} \tag{6.21}$$

The variables in the system vectors have been defined in the previous Section, and therefore, will not be redefined here for brevity.

Similar to Approach One, the linear and time-discrete system matrices are required and can be achieved by using the Jacobian matrix for linearization and the matrix exponential for time discretization.

This marks the second EKF, which is referred to from now on as $\text{EKF}_{\text{se-ext}}$.

6.3.3 Approach Three: Using the Dynamic Model

As mentioned earlier in the literature review in Section 1.2.6, the research in vehicle state estimation is extensive, and there have been many successful attempts to achieve accurate state estimation results for Ackermann vehicles by employing a dynamic vehicle model. However, these methods cannot be directly used on ADTs. This is not only due to the different steering mechanisms but also due to the very limited sensor setup in commercial ADTs that hinders the applicability of the state-of-the-art methods. This issue is explained in more detail below.

Figure 3.6 showed the side view of a tire with the applied torque and the resulting torques and forces. For a state-of-the-art state estimation algorithm to work, a variable or a combination of these variables is required to be known. This known variable would function as an input to the vehicle model in the filter. For example, the applied torque T, or the resulting rotational speed of the wheel Ω, or the resulting tire forces F_x and F_y, etc. Since none of the mentioned information about the wheels is available on the ADTs, current approaches do not apply to ADTs for state estimation. In summary, there are no sensors available to deliver information about any inputs to the vehicle body.

However, the state estimation algorithm is being designed as a companion for the trajectory-tracking MPCs. When a trajectory-tracking MPC is running, the reference values for the articulation angle rate ω_{ref} and speed v_{ref} are known. The idea of this Section is, that these reference values can serve as virtual sensors that provide variables that practically function as inputs to the vehicle body and control the vehicle's motion. Therefore, if a dynamic model is created with these two variables as inputs, it can be used for state estimation in an ADT.

The reduced dynamic ADT model presented in Section 3.6.2 is a dynamic model that fulfills the defined requirement by having the reference articulation angle rate ω_{ref} and speed v_{ref} as its inputs. Therefore, this model is used for the state estimation in ADTs with a Kalman filter.

In addition to the state estimation, it is desired that this Kalman filter can also estimate parameters as a part of the estimation. Among the parameters that are used in the reduced dynamic model, two stand out since they are the only parameters that are not measurable or are subject to changes while driving. Therefore, they are included in the parameter estimation. These parameters are the tire cornering stiffness C_α and the articulation angle rate time constant T_ω. Cornering stiffness C_α was introduced to this model in (3.62), and the articulation angle rate time constant T_ω in (3.59). These parameters may change in value in different vehicle loads, road conditions, and temperatures.

Since the articulation angle rate time constant T_ω is used in its inverse form in the equations, the parameter $\omega_T := 1/T_\omega$ is introduced and included in the parameter estimation instead. From now on, ω_T is referred to as the articulation angle rate time constant. These parameters are added to the state vector of the reduced dynamic model as random walk. The final vehicle model that is used in the state estimation has the following state, measurement, and input vector:

$$\begin{aligned} \boldsymbol{x}_{\text{se-dyn}} =& [x_{\text{f}}, y_{\text{f}}, \psi_1, \psi_2, u_{\text{f}}, v_{\text{f}}, \dot{\psi}_1, \dot{\psi}_2, C_\alpha, \omega_{\text{T}}]^\top, \\ \boldsymbol{y}_{\text{se-dyn}} =& [x_{\text{f}}, y_{\text{f}}, \psi_1, \psi_2, u_{\text{f}}, v_{\text{f}}, \dot{\psi}_1, \dot{\psi}_2]^\top, \ \boldsymbol{u}_{\text{se-dyn}} = [\omega_{\text{ref}}, v_{\text{ref}}]^\top. \end{aligned} \tag{6.22}$$

In the system vectors, $(x_{\text{f}}, y_{\text{f}})$ is the position of the front center of gravity, and u_{f} and v_{f} are the longitudinal and lateral speed elements at the front center of gravity, respectively. The front and rear yaw angles are denoted by ψ_1 and ψ_2, respectively. This model is used in a JEKF for ADT state and parameter estimation. Similar to the previous approaches, the linearized and time-discrete system matrices must be established by employing the Jacobian matrix for linearization and the matrix exponential for discretization.

As the state vector indicates, the sideslip angle is not itself a state in this model, and its value has to be calculated using the estimated value of the lateral speed at the front center of gravity v_{f} by employing the kinematic analysis of the vehicle dynamics:

$$\alpha = \arctan \frac{v_{y_1}}{v_{x_1}} = \arctan \frac{v_{\text{f}} - b_1 \dot{\psi}_1}{u_{\text{f}}}. \tag{6.23}$$

In this equation, v_{x_1} and v_{y_1} are the longitudinal and lateral speed elements at the front axle, v_{f} is the lateral speed at the center of gravity and is an estimated state, and b_1 is the distance between these two points.

This is the third Kalman filter, which is referred to as $\text{JEKF}_{\text{se-dyn}}$ from now on.

After establishing the Kalman filters, in the next Section, the simulation results are presented and the filters are discussed.

6.3.4 Filter and Estimation Setup

The previous Section presented three filters that are to be implemented and compared with each other in this Section. These filters are $\text{EKF}_{\text{se-base}}$, an EKF with the baseline version of the kinematic-slip model that originates from the state-of-the-art methods directly, $\text{EKF}_{\text{se-ext}}$, a second EKF with the extended version of the kinematic-slip model where actual model information is used in the equation of the sideslip angle and IMU data is utilized, and finally $\text{JEKF}_{\text{se-dyn}}$, a JEKF with the reduced dynamic model, which can also estimate important parameters.

$\text{EKF}_{\text{se-base}}$ receives its signals from the CAN bus of the vehicle and the navigation filter. $\text{EKF}_{\text{se-ext}}$ receives its information not just from the CAN bus and the navigation filter but also from the IMU. Finally, the $\text{JEKF}_{\text{se-dyn}}$ additionally requires the reference variables for articulation angle rate ω_{ref} and speed v_{ref}, which can be provided from the companion MPC that runs in parallel. The filters run at 100 Hz. Figure 6.2 shows the setup of the filters.

The Kalman filters are tuned using expert knowledge and empirical trial and error. The covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ of the filters are as follows.

$$
\begin{aligned}
\mathbf{Q}_{\text{se-base}} =& 10^{-3} \cdot \text{diag}([1, 1, 1, 1, 0.01]),\\
\mathbf{R}_{\text{se-base}} =& \text{diag}([1, 1, 1.5 \cdot 10^{-3}, 1.5 \cdot 10^{-5}]),\\
\mathbf{Q}_{\text{se-ext}} =& \mathbf{Q}_{\text{se-base}},\ \mathbf{R}_{\text{se-ext}} = \mathbf{R}_{\text{se-base}},\\
\mathbf{Q}_{\text{se-dyn}} =& 10^{-4} \cdot \text{diag}([1, 1, 1, 1, 1, 1, 1, 1, 12 \cdot 10^{7}, 0.01]),\\
\mathbf{R}_{\text{se-dyn}} =& \text{diag}([1, 1, 1.5 \cdot 10^{-3}, 1.5 \cdot 10^{-5}, 5 \cdot 10^{-2},\\
& 5 \cdot 10^{-2}, 5 \cdot 10^{-2}, 5 \cdot 10^{-3}]).
\end{aligned}
\tag{6.24}
$$

6.4 State Estimation Simulation Evaluation

This Section presents the simulation setup, validation results, and a discussion about the filters established in the previous Sections.

6.4.1 Simulation Setup

Earlier in Chapter 3, the simulation environment has been validated with real-world data in terms of the position of the vehicle, its heading, and articulation angle. However, the signals that the filters receive when running on the real vehicle are not perfect and are affected by noise. This means, that if the tuning and testing of the filters occur in the simulation environment, the signals used by the filters need to be at least as noisy as in reality. This will help establish a better evaluation of the filters and better prepare and tune them for real experiments. This can be achieved by adding noise to the simulation signals that are of interest to the filters. The characteristics of the added noise must be determined by establishing a noise analysis.

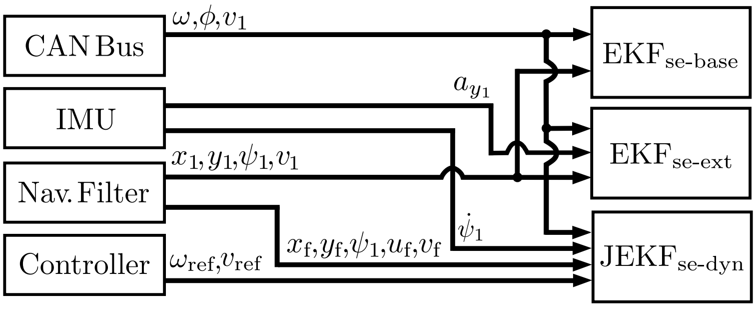


Fig. 6.2: Filter setup. Adapted from [M1].

For the noise analysis, the localization module is investigated where the signals of interest in reality come from. As mentioned earlier in Chapter 2, the localization module of the vehicle has been developed and implemented by the authors of [12] and [13]. In these works, the authors present an in-depth explanation of their navigation and localization modules. Furthermore, they present a detailed analysis of the measurement noise on the resulting signals compared to ground truth. In this case, the ground truth corresponds with the simulation signals. Therefore, noise is added to each signal with a standard deviation based on the error analysis in the mentioned works. Table 6.1 lists the relevant measurement noise parameters. This noise is the reason why a filter is required at all and the sideslip angle cannot be calculated by only using the localization module data and vehicle geometry.

In order to avoid overfitting the filters to the simulation data, the filter tuning presented in the previous Section intentionally deviated from the exact standard deviations of known measurement noise, prioritizing generalization over extreme precision.

6.4.2 Simulation Scenario

The goal of this simulation is to evaluate, how the filters perform when the vehicle does the routine work of an ADT. During the 500 s simulation, the vehicle follows a reference articulation angle rate ω_{ref} and a reference speed v_{ref}. The more dynamic the reference values to the vehicle are, the better the filters can estimate and converge. However, the values of the reference signals in the simulation deliberately avoid using the whole spectrum of the vehicle's dynamic range. This creates a more challenging scenario for the filters. Figure 6.3 shows the first 25 s of the actuation. In the rest of the simulation, this initial pattern is repeated. As this Figure also indicates, the simulated articulation angle rate ω is only up to 5 deg/s as opposed to the full range of the vehicle's dynamic capabilities of 12 deg/s.

Table 6.1: Input signals noise analysis.

Parameter (front body)	**Symbol**	**Error standard deviation**
Axle position in local coordinates	x_1, y_1	0.1 m
Axle speed in vehicle coordinates	v_{x_1}, v_{y_1}	0.01 m/s
Yaw and articulation angle	ψ_1, ϕ	$3 \cdot 10^{-4}$ rad, $3.1 \cdot 10^{-6}$ rad
Articulation and yaw angle rate	ω, $\dot{\psi}_1$	0.01 rad/s
IMU lateral acceleration	a_{y_1}	0.1 m/s^2

The initialization of all states is at 0, and the initial values of the parameters in the JEKF are at 70% of their true value assuming that an initial guess for these parameters with 30% accuracy is known. In order to examine the JEKF's behavior when facing changes in the parameters after the initial convergence, the value of cornering stiffness C_α changes to 80% of its value at $t = 350\,\mathrm{s}$. This sudden and drastic change in the parameter is not realistic and serves as a testing scenario to evaluate how the filter reacts in such cases.

Furthermore, in order to better prepare the filters for real-world applications, and to test the filters under unforeseeable circumstances, the filters are tested not only with the discussed measurement noise, but also with factors of three, five, and ten of the measurement noise.

6.4.3 Simulation Results

Figure 6.4 shows the results of the parameter estimation with the $\mathrm{JEKF}_{\text{se-dyn}}$ and Figure 6.5 shows the results of the sideslip angle estimation of all filters. In Figure 6.4, the y axis shows the normalized values for the parameters so that the goal is to converge to 1. In Figure 6.5, the results of the filters are compared with the ground truth. Additionally, the results of calculating the sideslip angle only using the IMU data and the position vector from the navigation and localization module are also added to the comparison. In this Figure, the absolute sideslip angle error is shown over the whole simulation duration, and the real values of the estimated variables are shown only over 30 s of the simulation for better visibility.

Table 6.2 presents an error analysis of the results in terms of the mean absolute error (MAE) and the root mean square error (RMSE) of all filters. The error analysis is done not just with the expected noise standard deviations, but also with factors of three, five, and ten of the measurement noise. This is also included in this table.

The analysis begins with the parameter estimation results. As Figure 6.4 also indicates, both parameters manage to converge to their reference values after approximately 200 s.

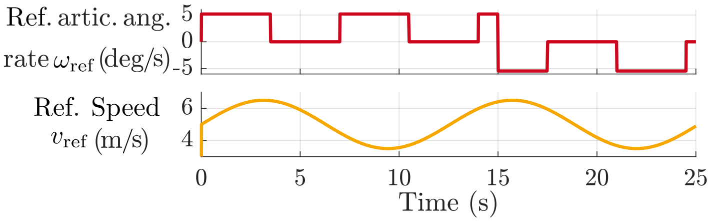


Fig. 6.3: Reference articulation angle rate ω_{ref} and speed v_{ref} used as input signals to the vehicles in the simulation. Adapted from [M1].

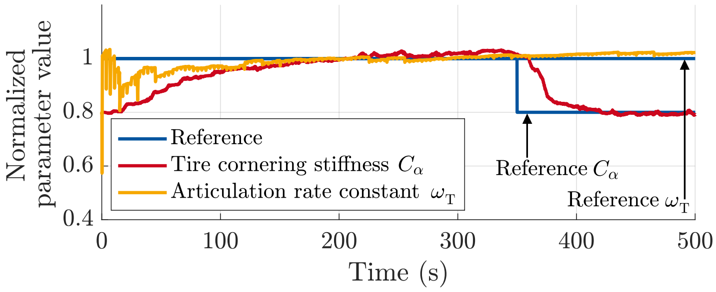


Fig. 6.4: Parameter estimation results. Adapted from [M1].

The converged values show a maximum error of 5%. When the value of the tire cornering stiffness C_α changes at t =350 s, the estimated value converges after 80 s again.

The analysis continues with the sideslip angle estimation. Figure 6.5 shows, firstly, that calculating the sideslip directly by using the noisy signals from the navigation filter results in a noisy sideslip angle signal that is not suitable to be used further. This also supports the initial claim about the navigation filter results being practically unusable for calculating the sideslip angle directly. The filters converge 2 s after the simulation starts. $\mathrm{EKF}_\text{se-base}$ delivers acceptable results in terms of MAE and RMSE but suffers from a noticeable phase delay, which is expected due to the fact that the sideslip angle is modeled with random walk in this approach. As a result, the filter's estimation for the sideslip angle is reactive, occurring only after the changes in the true value of the sideslip angle have already manifested in other vehicle states. Consequently, this estimation is delayed.

Table 6.2: Error analysis (values in degrees).

Noise factor	**Filter**	**MAE**	**RMSE**
Noise factor 1	$\mathrm{EKF}_\text{se-base}$	1.49	1.76
	$\mathrm{EKF}_\text{se-ext}$	1.00	1.24
	$\mathrm{JEKF}_\text{se-dyn}$	0.16	0.20
Noise factor 3	$\mathrm{EKF}_\text{se-base}$	1.52	1.82
	$\mathrm{EKF}_\text{se-ext}$	1.27	1.58
	$\mathrm{JEKF}_\text{se-dyn}$	0.26	0.33
Noise factor 5	$\mathrm{EKF}_\text{se-base}$	1.56	1.87
	$\mathrm{EKF}_\text{se-ext}$	1.50	1.87
	$\mathrm{JEKF}_\text{se-dyn}$	0.33	0.41
Noise factor 10	$\mathrm{EKF}_\text{se-base}$	1.65	2.00
	$\mathrm{EKF}_\text{se-ext}$	1.96	2.45
	$\mathrm{JEKF}_\text{se-dyn}$	0.45	0.56

$\text{EKF}_{\text{se-ext}}$ delivers better but comparable results to $\text{EKF}_{\text{se-base}}$ in terms of error analysis. The more important improvement in the results of $\text{EKF}_{\text{se-ext}}$ is the decreased phase delay. This development was also expected since the sideslip angle differential equation directly utilizes the current system information from the IMU data. Consequently, the estimation of the sideslip angle is not solely dependent on the development of other states, allowing the filter to respond more quickly to changes in the sideslip angle and thereby reducing the phase delay. However, this inclusion of model knowledge and this added dependency on IMU information makes the filter more susceptible to unexpected noise, especially when it affects the IMU signal since it directly affects the estimated variable. This is evident from Table 6.2 where in noise factor ten, the $\text{EKF}_{\text{se-base}}$ delivers better results than the $\text{EKF}_{\text{se-ext}}$.

Finally, as expected, the $\text{EKF}_{\text{se-dyn}}$ delivers the best results both in terms of the error analysis and the phase delay where no noticeable phase delay is observed. This filter functions acceptably even under an unexpected noise factor of ten. Another important observation is that while all filters deliver MAEs under 2°, the JEKF se-dyn is the only filter that manages to keep the absolute error value under 1° at all times after the initial convergence.

As the results indicate, all filters converge and deliver the sideslip angle with different qualities. The results further indicate that adding model knowledge or sensor information to the filters improves the estimation results. This means that achieving better results in terms of the MAE and phase delay between the sideslip angle and its estimated value comes at the cost of developing more complex system models and employing more sensor information. As a result, none of the filters is the absolute better choice for the state estimation, and depending on the use case and available resources, one should be chosen.

Another important aspect of the results specifically of the JEKF is the proof of concept in using the reduced dynamic model. As mentioned in Section 1.2.6, one of the main goals of the estimation was to display the ability of the reduced dynamic model in capturing the important dynamic aspects of the vehicle, while being a simplified version of the full dynamic model. Now that this specific setup of the reduced dynamic model has proven to properly mirror the important dynamic aspects of the vehicle, in the future, other setups of this model with the inclusion of different sensor setups can also be used for the state estimation. This filter marks the first utilization of dynamic models in ADT control and state estimation.

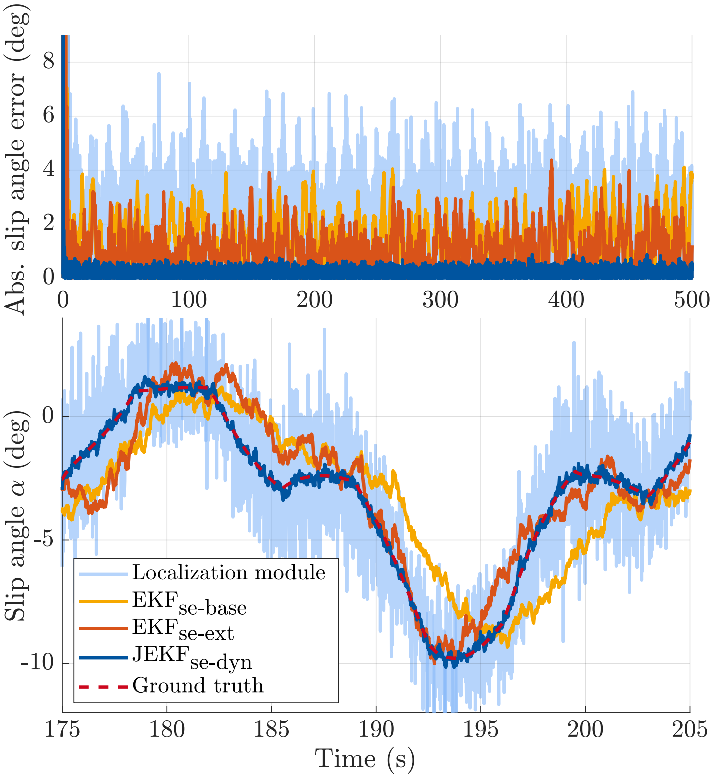


Fig. 6.5: Estimated sideslip angle α and the absolute error of the sideslip angle estimation using the filters established in this Chapter. The sideslip angle, calculated by only using the signals from the localization module, and the ground truth are also presented. Adapted from [M1].

6.5 Summary

This Chapter presented three state estimation algorithms based on Kalman filtering with different levels of setup complexity. This Chapter also showed that more accurate models that utilize more system knowledge and sensor information deliver better results for state estimation at the cost of setup difficulty. Furthermore, this Chapter represents a milestone, marking the first use of a fully dynamic vehicle model for state and parameter estimation in ADTs, setting a new standard in the field. The next Chapter presents MPC concepts that employ the estimated sideslip angle and try to compensate for it.

7 Trajectory-Following MPC with Sideslip Angle Compensation

7.1 Sideslip Angle Compensation Objective

The previous Chapter discussed the importance of creating more accurate models for the MPCs to increase the operational speed and control accuracy. It was also highlighted that neglecting the sideslip angle is the primary cause of reduced accuracy in kinematic models compared to dynamic models. Consequently, it was concluded that by addressing the issue of sideslip angle, the model accuracy and controller performance can be improved.

There are a variety of methods that can help consider the sideslip angle in the MPC. Some of the most common methods include using a vehicle dynamic model, which automatically includes the sideslip angle in its equations, or using learning methods such as in [40], which learns the model uncertainties including the sideslip angle. However, this Chapter intends to address this issue by incorporating the sideslip angle in the kinematic-slip model from Chapter 3. Together with the state estimation from the previous Chapter providing the current sideslip angle, this Chapter establishes the entire toolchain required for a trajectory-following MPC with sideslip angle compensation.

The challenge of this method, as mentioned in Chapter 1.2, is that the inclusion of the sideslip angle in this model is solely as a parameter. As a result, the future state of the sideslip angle when given an input and an initial state cannot be predicted, unlike other vehicle states. Consequently, the predictive aspect of an MPC can not be realized for the sideslip angle directly.

As discussed thoroughly in Section 1.2.4, the authors in [59] explore the application of the kinematic-slip model in an MPC. To highlight the limitations of this method, a summary of the key problems is presented in the following:

- The actuator phase delay is not considered.
- The dead time is not considered.
- It is unclear, if and how the sideslip angle is obtained for the length of the MPC prediction horizon. In other words, only the initial sideslip angle value is used in

the MPC. It is presumed that the same sideslip angle value is used throughout the entire length of the MPC's prediction horizon.

The first two issues have been proven in Chapters 4 and 5 to lead to poor results or in some cases, cause the MPC to become infeasible within a few seconds of operation. The third issue could lead to various problems depending on the test scenarios due to the inadequate MPC setup. However, the discussed work uses a simulation environment that employs the same model as the MPC without phase delay or dead time on a circular track, where the sideslip angle remains constant. This explains how the MPC functions despite the presented limitations. As a result, a direct comparison with this state-of-the-art approach is not possible.

To address these issues, this Chapter presents two new MPCs. At the end of this Chapter, a comparison between the new MPCs, the established path-following MPC from Chapter 4, and the forward trajectory-following MPC from Chapter 5 is presented. The path-following MPC from Chapter 4 is called $\text{MPC}_{\text{pf-Fr}}$, and the FAR_{f} trajectory-following MPC from Chapter 5 is referred to as $\text{MPC}_{\text{tf-kin}_\text{f}}$ from now on. The abbreviation pf-Fr and tf-kin_f stand for path-following MPC using the Frenet coordinates and the trajectory-following MPC using the kinematic model for the f-ADT, respectively. Furthermore, since the current value of the sideslip angle is not known in the real vehicle, the state estimation algorithm from the previous Chapter is utilized to close the feedback loop, putting the collaboration of the algorithms to the test.

The implementations of this Section are solely developed and tested for the f-ADT since the operational speeds for the compact ADT (c-ADT) are limited, and the sideslip angle does not play a noticeable role for that vehicle [68]. The contents of this Chapter are based on [M1].

7.2 MPC Setup

This Section presents the setup for two new MPCs, which are $\text{MPC}_{\text{tf-slip}_\text{A}}$ and $\text{MPC}_{\text{tf-slip}_\text{B}}$.

7.2.1 MPC Model

As discussed in Section 7.1, the model for the trajectory-following MPC is based on the kinematic-slip model (3.10) from Chapter 3. However, similar to the assumption in Chapter 6 explained in Section 6.3.1, the MPC uses the assumption that the rear sideslip angle is zero. After adding the identification equation for articulation angle rate ω from (3.1) and speed v_1 from (3.63), the following model is established:

$$\dot{x}_1 = v_1 \cos(\psi_1 + \alpha), \tag{7.1a}$$

$$\dot{y}_1 = v_1 \sin(\psi_1 + \alpha), \tag{7.1b}$$

$$\dot{\psi}_1 = \frac{\sin(\phi + \alpha)}{L_2 + L_1 \cos(\phi)} v_1 + \frac{L_2}{L_2 + L_1 \cos(\phi)} \omega, \tag{7.1c}$$

$$\dot{\omega} = (-\omega + k_\omega \omega_{\text{ref}})/T_\omega, \tag{7.1d}$$

$$\dot{v}_1 = (-v_1 + k_v v_{\text{ref}})/T_v. \tag{7.1e}$$

In this system, v_1 is the speed at the front axle. Vectors (x_1, y_1) and (x_2, y_2) are the position vector of the front and rear axles of the vehicle in inertial coordinates. Parameter ϕ is the articulation angle, while ψ_1 and ψ_2 are the front and the rear yaw angle, respectively. Additionally, ω is the articulation angle rate, α is the front sideslip angle, and L_1 and L_2 denote the distance from the hinge to the front and the rear axle, respectively.

The inputs to this model are the reference speed v_{ref} and articulation angle rate ω_{ref} forming the control inputs vector from the MPC as $\boldsymbol{u}_{\text{tf-slip}} = [\omega_{\text{ref}}, v_{\text{ref}}]$. The state vector is $\boldsymbol{x}_{\text{tf-slip}} = [x_1, y_1, \psi_1, \phi, \omega, v_1]$. The abbreviation tf-slip here refers to trajectory-following using the kinematic-slip model.

The MPC requires linearized and time-discrete system matrices, which can be achieved by following the same steps introduced in Section 4.1.2 yielding the following system:

$$\boldsymbol{x}_{\text{tf-slip,d}}(k+1) = \boldsymbol{A}_{\text{tf-slip,d}} \boldsymbol{x}_{\text{tf-slip,d}}(k) + \boldsymbol{B}_{\text{tf-slip,d}} \boldsymbol{u}_{\text{tf-slip,d}}(k+1), \tag{7.2}$$

where $\boldsymbol{A}_{\text{tf-slip,d}}$ and $\boldsymbol{B}_{\text{tf-slip,d}}$ are the system and input matrices. The abbreviation tf-slip means trajectory-following MPC using the kinematic-slip model.

7.2.2 Sideslip Angle over Prediction Horizon

As mentioned earlier in Section 7.1, one of the key issues with the state-of-the-art methods in sideslip angle compensation was neglecting the sideslip angle over the prediction horizon and only considering the initial sideslip angle.

Both MPCs presented in this Chapter, $\text{MPC}_{\text{tf-slip}_\text{A}}$ and $\text{MPC}_{\text{tf-slip}_\text{B}}$, follow the same principles in their idea as they both utilize the kinematic-slip model. However, $\text{MPC}_{\text{tf-slip}_\text{A}}$ is closely similar to the state-of-the-art method as it only relies on the initial sideslip angle and does not consider sideslip angle values over the prediction horizon. For all the subsequent steps, it assumes a sideslip angle of zero, practically using the kinematic-slip model only in the first MPC step and switching to the baseline kinematic model (3.9) after that.

The second MPC $\text{MPC}_{\text{tf-slip}_\text{B}}$, however, employs an approach to predict the values of the future sideslip angles. In this approach, the MPC utilizes the dynamic model from Section 3.6.2 and the MPC output from the latest optimization cycle and obtains future sideslip angles externally. This is done by integrating the dynamic vehicle model. For the integration, the Runge-Kutta method of order four with an integration time-step of 0.01 s is employed. The predicted sideslip angle values are then set inside the MPC matrices when establishing the Hesse matrix. Figure 7.1 illustrates the setup of these new MPCs where Block 2 represents the content of this Section. The function of Block 1 is explained in the following Section.

7.2.3 Dead Time Compensation

Dead time compensation for the new MPCs follows the same principle as the previous MPCs originally discussed in Section 4.1.3. This principle involves predicting vehicle states for after the dead time and providing them to the MPC as the current state. However, in order to fully compensate for the dead time, the value of the sideslip angle also must be predicted. This is not possible if, similar to the previous MPC setups, the kinematic vehicle model is employed in the integration.

To address this issue, two different methods for the two new MPCs are acquired. For $\text{MPC}_{\text{tf-slip}_\text{A}}$, the compensation is only for the dead time in the kinematic vehicle states and not in the sideslip angle. This method avoids further complexities but relies on outdated sideslip angle values. This assumption holds when the vehicle is on a circular track with no changes in the required articulation angle, as the sideslip angle remains

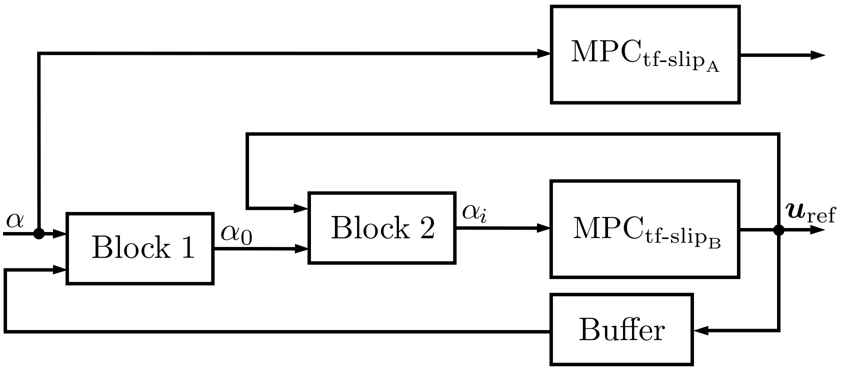


Fig. 7.1: Structure of the $\text{MPC}_{\text{tf-slip}_\text{A}}$ and $\text{MPC}_{\text{tf-slip}_\text{B}}$ in terms of the prediction for the sideslip angle. Block 1 represents the compensation of the dead time by utilizing the dynamic vehicle model and the buffered MPC output. Block 1 yields the α_0, which is the predicted sideslip angle after the dead time. Block 2 represents the process of predicting the sideslip angles throughout the MPC prediction horizon, yielding $\alpha_i, \quad i \in [1, N]$, where N is the MPC horizon. Adapted from [M2] ©2022 IEEE.

constant before and after the dead time. However, the method becomes less accurate as articulation and consequently sideslip angle changes increase during the dead time.

In contrast, for $\text{MPC}_{\text{tf-slip}_\text{B}}$, the reduced dynamic model from Section 3.6.2 is utilized to compensate for the dead time not only in the kinematic states but also in the sideslip angle. While this method adds complexity to the MPC by incorporating the dynamic model and its parameters, it provides a prediction for the sideslip angle.

All further MPC setups including the reference trajectory preparation, cost function, constraints, and solver are identical to the MPC setup in Chapter 5.

7.3 Sideslip Angle Compensating MPC Simulation Evaluation

In the following, the best practice analysis of the vehicle operation routine is presented followed by a discussion of the simulation setup and finally simulation results.

7.3.1 Best Practice Analysis and Reference Trajectory Design

The goal of the simulation in this Chapter is to compare the performance of all the (forward) MPCs presented in this work in a more challenging setup than before. In order to achieve this goal, a best practice analysis (BPA) is performed on the vehicle. The BPA was presented and illustrated in Chapter 1.2 and is explained here in more detail and visualized in Figure 7.2 again for better accessibility. To establish the BPA, vehicle data during its routine operation in a mine were gathered consecutively over 118 h. The vehicle was driven by a human driver carrying out its normal tasks. Figure 7.2 illustrates the frequency of occurrences for specific combinations of steering angle and speed during the test. The data were sampled every second, and each grid cell in the plot represents how many of these sampled points fall within a given range of steering angles and speeds. The color intensity within each cell reflects the frequency of sampled points in that range: warmer colors indicate higher frequencies (more sampled points).

As evident in the Figure, a speed higher than $7\,\mathrm{m\,s^{-1}}$ is mainly achieved with an articulation angle close to 0 deg. Furthermore, the most common operation speed for the vehicle is around $2\,\mathrm{m\,s^{-1}}$. Another observation is that speeds between $5\,\mathrm{m\,s^{-1}}$ and $7\,\mathrm{m\,s^{-1}}$ are primarily achieved at articulation angles less than 10 deg. Another important reading is that the full range of the articulation angle is used at speeds lower than $3\,\mathrm{m\,s^{-1}}$. The design of the reference trajectory in this Chapter is inspired by the BPA.

The reference trajectory is designed to include the more challenging aspects of the speed-articulation angle plot. This means driving at higher speeds and higher articulation

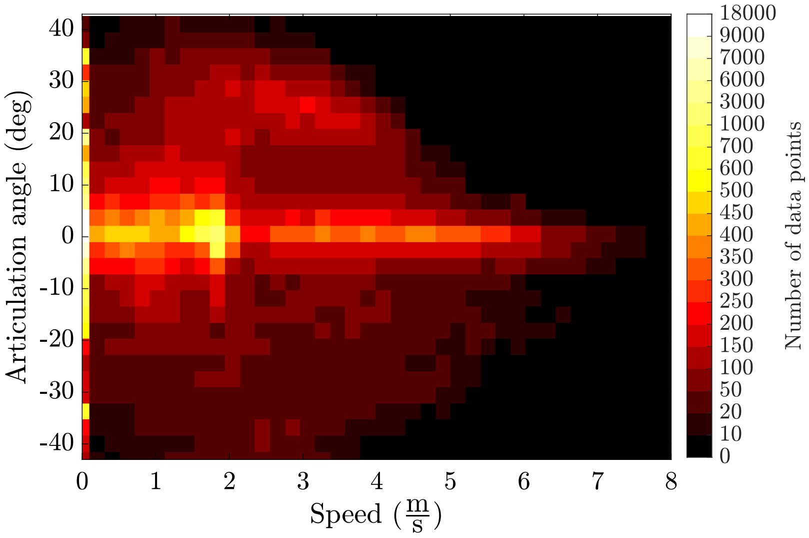


Fig. 7.2: Visualization of the BPA. Adapted from [M2] ©2022 IEEE.

angles. The designed reference trajectory pushes the vehicle to its dynamic limits and creates a proper testing possibility to compare the MPCs. Figure 7.3 shows the designed reference trajectory, and Figure 7.4 compares the reference trajectory with the BPA. As Figure 7.4 indicates, the reference trajectory covers not only the routine and most driven operation points of the ADT, indicated by articulation angles between $-20\,\mathrm{deg}$ and $20\,\mathrm{deg}$, but also more dynamic operation points even out of the ADT routine, marked by articulation angles above $20\,\mathrm{deg}$ at speeds above $4\,\mathrm{m\,s^{-1}}$.

7.3.2 Simulation Setup and Results

The goal of the simulation is to remain as close as possible to the given reference trajectory. In this simulation, four MPCs are put to the test: $\mathrm{MPC_{pf\text{-}Fr}}$ from Chapter 4, $\mathrm{MPC_{tf\text{-}kin_f}}$ from Chapter 5, and $\mathrm{MPC_{tf\text{-}slip_A}}$ and $\mathrm{MPC_{tf\text{-}slip_B}}$ from this Chapter. As mentioned before, the entire toolchain is established by employing the sideslip angle estimation from the previous Chapter in the feedback loop. Therefore, the value for the sideslip angle in the MPCs comes from the state estimation method $\mathrm{JEKF_{se\text{-}dyn}}$ from the previous Chapter.

Figure 7.5 presents the results of the simulations for the MPCs during this test in terms of the lateral error and the speed. Table 7.1 presents the error analysis in terms of root mean squared error (RMSE). Firstly, the first two MPCs are compared, $\mathrm{MPC_{pf\text{-}Fr}}$ and $\mathrm{MPC_{tf\text{-}kin_f}}$, with RMSEs of $1.35\,\mathrm{m}$ and $0.69\,\mathrm{m}$, respectively. $\mathrm{MPC_{pf\text{-}Fr}}$ drives with a constant speed of $4\,\mathrm{m\,s^{-1}}$ as expected while $\mathrm{MPC_{tf\text{-}kin_f}}$ finds the optimal speed to follow the trajectory. Comparing the RMSEs of these two MPCs shows how including the speed in the control improves the overall performance of the controller. The lateral error plot of these two MPCs indicates that at 30% of the path, the first major difference between the two MPCs occurs. This is due to the higher speed of $\text{MPC}_{\text{tf-kin}_\text{f}}$ pushing the vehicle to its limits. At the second half of the path, $\text{MPC}_{\text{tf-kin}_\text{f}}$ reduces the speed to follow the

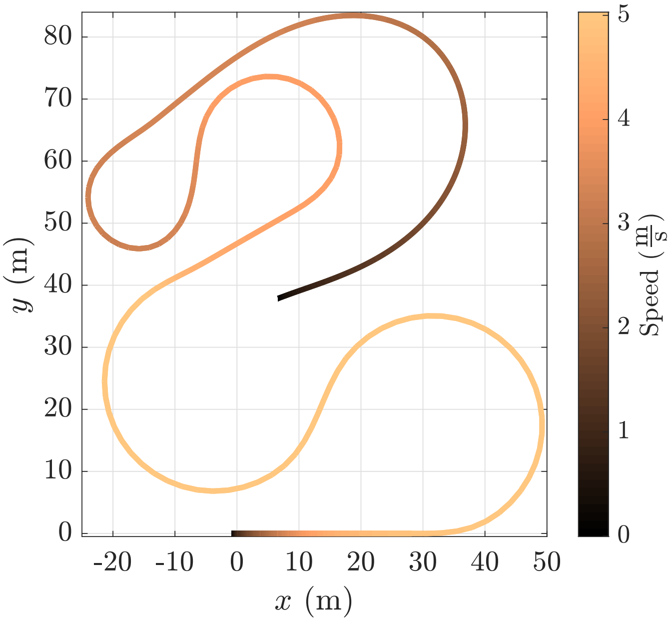


Fig. 7.3: Reference trajectory for the simulation. Adapted from [M2] ©2022 IEEE.

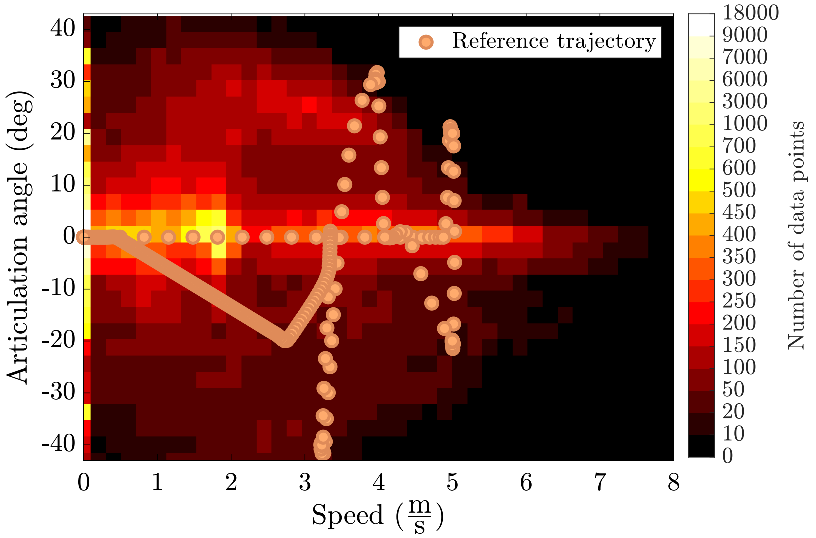


Fig. 7.4: BPA vs. reference trajectory. The BPA is at the background and the copper-colored circles represent data points from the reference trajectory. Adapted from [M2] ©2022 IEEE.

trajectory while MPC$_{\text{pf-Fr}}$ maintains its speed causing its lateral error to suffer.

It is, however, expected that with speed included as a controller output in the optimization problem, the controller achieves better results than a controller limited to a constant speed not being able to reduce the speed at sharp curves. Consequently, these results do not indicate that trajectory-following MPCs are inherently superior to path-following approaches, but that they suit the requirements of this project better where a trajectory planner is already available, and the routine operation involves frequent changes in the speed.

The focus of the comparison now is shifted to the MPCs from this Chapter MPC$_{\text{tf-slip}_\text{A}}$ and MPC$_{\text{tf-slip}_\text{B}}$. With an RMSE of 0.64 m and 0.60 m and an improvement in the maximum overshoot of more than 20% (11% of the path) compared to MPC$_{\text{tf-kin}_\text{f}}$, both of these MPCs prove to improve the results. MPC$_{\text{tf-slip}_\text{B}}$ shows another 6% improvement in the RMSE and up to 14% in the maximum overshoot compared to MPC$_{\text{tf-slip}_\text{A}}$. As expected, a noticeable difference between the performance of these two MPCs occurs when the vehicle experiences sudden changes in its direction (11% and 30% of the path). At such

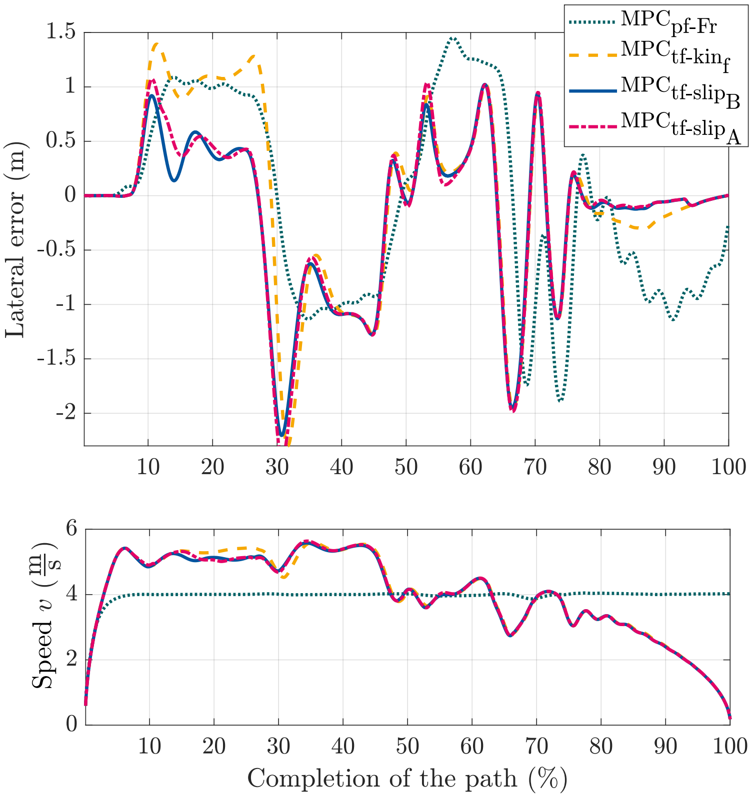


Fig. 7.5: MPC performance comparison. Both the lateral error and speed are plotted over the completion of the path and not over time for better comparability. Adapted from [M2] ©2022 IEEE.

Table 7.1: Error analysis of the MPC performance during the simulation.

MPC name	**RMSE**
$\mathrm{MPC}_{\text{pf-Fr}}$	1.35 m
$\mathrm{MPC}_{\text{tf-kin}_\text{f}}$	0.69 m
$\mathrm{MPC}_{\text{tf-slip}_\text{A}}$	0.64 m
$\mathrm{MPC}_{\text{tf-slip}_\text{B}}$	0.60 m

moments, $\mathrm{MPC}_{\text{tf-slip}_\text{B}}$ reacts faster with the predicted values of the sideslip angle in its matrices, while $\mathrm{MPC}_{\text{tf-slip}_\text{A}}$ is reactive to such changes and can consider them only after they have already happened.

In conclusion, these MPCs all fulfill the defined requirements and should be employed based on the available resources, specific use cases, the allowed complexity, and the desired accuracy. At the end of this discussion, it should be noted that even though the two new MPCs discussed in this Chapter fulfilled the requirement and managed to partially compensate for the sideslip angle, they are both limited in their application because their model treats the sideslip angle as a parameter and not a state. In other words, although the sideslip angle over the MPC prediction horizon was predicted and set inside MPC matrices, the predictive aspect of the MPC can only be realized when both the future values and the dynamic of a variable are known. Consequently, these MPCs cannot perform as accurately as an MPC incorporating the sideslip angle as a state. Therefore, to achieve higher speeds or better accuracies at current speeds, it is recommended to investigate the other proposed methods at the beginning of this Chapter including using a dynamic vehicle model or learning-based methods.

7.4 Summary

This Chapter introduced two new MPC concepts that can consider and compensate for the sideslip angle. While one MPC only relies on the initial sideslip angle, in the other MPC approach, the future sideslip angles are acquired and set in the MPC matrices and considered in the optimization problem. In comparison with a path-following MPC, the trajectory-following MPCs of this work managed to achieve better results since they can better manage situations involving frequent changes in the speed.

8 Discussion and Outlook

8.1 Introduction to the Discussion

This Chapter presents the synthesis of the research presented throughout this work and situates the key findings within the broader context of autonomous ADTs in mining environments. The aim is to reflect on the contributions of this work and to evaluate the results in a broader context. Additionally, this Chapter provides an outlook for future research by addressing open questions and current trends in the field.

8.2 Synthesis of Key Findings

The main goal of this project was to develop an autonomous fleet of ADTs within the context of mining operations. This work contributed by developing the control approach for the vehicles. Each Chapter of this work provided the required complementary advancements to reach this goal.

At the very beginning of the project, extensive testing was conducted to achieve a deep understanding of the ADT's dynamics. These tests highlighted important aspects of these vehicles, including the actuation delay and dead time. Furthermore, these tests also highlighted the unique dynamic effects of the steering in these vehicles, which differed from conventional Ackermann vehicles. Based on these findings and due to the challenges associated with conducting tests on the actual vehicle, the focus of this work became twofold: control and simulation. Given the limited availability of the vehicle for testing, simulation proved as a crucial tool and provided a platform for fine-tuning and testing the control approaches.

For the simulation, the dynamic modeling of the vehicle without explicitly modeling the cylinders was one of the main objectives. Such a model could contribute to a modular, expandable, and accurate simulation environment while also being suitable for model-based algorithms such as an MPC or state estimation methods. The dynamic modeling approach presented in this work fulfills this objective and managed to improve the accuracy of the simulation when compared to the simulation with the commonly used kinematic model. In validation tests, the simulated vehicle achieved a moving angle error of 3° compared to the 31° when the kinematic model was used during a 35 s test drive. **This addresses the first research question of this work, which explored how well the vehicle can be modeled**. The expandability of the model allows further improvement when more advanced sub-components are used such as magic formula [64] for the tires.

To develop the trajectory-following controllers, vehicle models were established for the MPCs. For this goal, the existing kinematic vehicle models were improved using the identified system delays. Modeling the system delays, which was done by conducting identification processes on the test results, proved to improve the control results compared to the state-of-the art approaches through contributing to the predictive feature of the MPC.

Another important aspect that further contributed to the control performance was considering different ADT sizes in modeling. This highlighted the importance of establishing tailored models for different systems to address even the slight differences in their dynamics. The same approach proved to be important when dedicated models were established for driving backwards, further contributing to the control results. With this contribution, the autonomous fleet would not be limited to driving forward or to specific ADT sizes, further making the autonomous solution interesting for mine owners.

The second research question explored how well an MPC can control the ADTs. Conducting the system identification and dividing the vehicle operation based on size and movement direction, e.g., forward or backwards, were the necessary steps on the way of answering the second research question. The final MPCs successfully fulfilled the defined requirements from Section 1.2, achieving results that are shown in tables 5.2 and 5.4. These MPCs were able to achieve higher speeds than the state-of-the-art methods without increasing the lateral error. This was possible, through the predictive aspect of the MPCs and their ability to adapt speed during the operation. These speeds are the typical operational speeds for a human driver and achieving autonomy while driving in those speeds meant to become one step closer to the realization of autonomous ADTs in mining environments. These results are discussed in more detail in Chapter 5.

Despite the successful experimental results, these MPCs present just one possible answer to the second research question while also exposing the shortcomings of the current framework, which primarily include the neglecting of the sideslip angle due the kinematic nature of the models. Furthermore, the realizations about the importance of the identification also highlighted a shortcoming in this current framework, as the currently offline identification does not allow online adaptation of the parameters.

The important role of the sideslip angle in control performance especially while cornering was demonstrated while comparing the real vehicle with models with and without sideslip angle consideration. **This led to the third research question, which explored the possible vehicle models that can be employed for the sideslip angle estimation**

and aims to discuss the resulting phase delays. In this work, three vehicle models with increasing fidelity were employed in the estimation. The first model was from the literature, the second model was the same model expanded by a single state for the sideslip angle, and the third model was a simplified version of the dynamic model. This work compared these models within an estimation framework and showed that only the dynamic model managed to deliver the sideslip angle with the expected accuracy with hardly noticeable phase delay. The discussion about the exact accuracy of each filter is presented in detail in Chapter 6.

With the sideslip angle estimation completed, **the fourth research question presented itself, which explored how much the control performance can be improved by employing sideslip angle in the control process.** Adding sideslip estimation and consideration to the control framework improved control performance by 20% in the maximum overshoot in a simulated track that was inspired by the routine operation of the vehicle. This improvement was expected since sideslip angle is one of the main causes of lateral error, as discussed in Chapter 3. However, the control approach presented in this work is only tested in a simulation and might not function as well in real-world scenarios. Furthermore, the consideration of the sideslip angle in the MPC is without a dedicated differential equation and solely as a parameter. This issue limits the functionality of the controller and the quality of the results. This realization means that with the sideslip angle now detected as an important factor affecting the control performance, it needs further research to be fully addressed.

The final demonstration involved the vehicles operating autonomously without humans, driving through routine operational paths of an ADT multiple times. This addressed the initial motivation of the project, which was to remove human drivers from the demanding driving situations in mining environments. The successful demonstration showcased that the MPCs can reliably complete the operational routine of an ADT. This outcome validated the controllers' performance in real-world scenarios and showcased their potential to contribute to safety and efficiency in mining operations.

8.3 Recent Developments in the Field and Current Trends

In the years since the project began and ended, the research in the field of autonomous ADTs has continued. As the newer studies such as [8, 22, 31, 37, 48, 62, 78, 81, 92] indicate, there has been a rise in the studies that focus on using model-based controllers. The trend of using NMPCs instead of MPCs has also risen probably since the MPCs do not necessarily offer an advantage over NMPCs anymore. This notion is made by considering how computers have become cheaper and NMPCs have become more common in recent years and can reportedly run at the same speeds as MPCs when proper algorithms are utilized [35]. However, using an NMPC does not necessarily contribute to the controller performance at the presented literature, as the operation speed is limited.

The system delays, which were not considered in the existing literature when the project started, are now considered in some works such as [48]. The results of employing system delays indicate improvements in terms of lateral error, which comply with the findings of this thesis. There are other attempts to improve the vehicle model as well, such as in [22], where a dynamic model is used inside an MPC framework. However, the dynamic model assumes the vehicle to be a rigid body at all times not affected dynamically by the steering. As a result, vehicle speed is still limited to $2\,\mathrm{m\,s^{-1}}$ in the tests.

Neglecting the system dead-time remains an issue in the current literature, which explains the limited vehicle speed in the mentioned current works. In the author's opinion, the trend of neglecting the sideslip angle, the dead time, and system delays continues since the current approaches fulfill the speed requirements, which do not include driving faster then $3\,\mathrm{m\,s^{-1}}$. In such scenarios, considering the mentioned factors do not contribute meaningfully to the controller performance.

For the simulation, the trend has not changed and kinematic models or physics-based software is further utilized ([37]). Establishing and employing a dynamic model for the vehicle is still mostly avoided.

In the current state of the art, there is a rise in research about path and trajectory planning for ADTs in works such as [2, 53, 90, 91, 99] with a focus on planning the path when driving backwards. Research in this area has the potential to considerably contribute to the controller performance since the controller results could suffer under inadequate planning algorithms.

In addition to the mentioned advancements, there is also the trend to fuse learning methods such as reinforcement learning, Gaussian Process Regression (GPR), or other methods with an MPC to learn model uncertainties or even the dynamic behavior of specific variables. This trend is currently rising in Ackermann vehicles with studies such as [23, 49, 72]. In [19], a GPR model is trained to learn unmodeled system dynamics and nonlinearities in an MPC for controlling an ADT. These approaches can complement the current framework of this thesis to replace the offline identification methods. Furthermore, learning methods can be utilized to improve kinematic models' accuracy compared to dynamic models. These are topics of future research for ADTs.

8.4 Conclusion and Outlook

This work presented the fundamental framework toward the realization of an autonomous fleet of ADTs. This framework addressed two challenges, which were the simulation and the control, by providing the crucial analysis of the special steering dynamics of the ADTs and the associated challenges, which included the inherent phase delay and dead time. The result was a complete simulation and control framework, which took the especial dynamic of ADTs into consideration in aspects such as different ADT sizes and forward and backwards driving.

An important outcome of this research was the successful dynamical modeling of the vehicle, which contributed to both the simulation and control. Additionally, the experimental validation of the simulation and controllers proved the functionality and reliability of the methods. Furthermore, the modular nature of all the implementation of this work paves the way for future implementations both in simulation and in control.

As discussed in Chapter 3, the performance of the simulation can improve by employing more accurate tire models, using a three-dimensional vehicle model, or by employing more accurate power distribution. Additionally, to better replicate the conditions in a mining field, different grounds should also be added to the simulation features to replicate the rainy, snowy, or dry weather conditions and their effect on the driving dynamics.

On the controller's side, there are multiple paths that can be taken to improve the results. A first step is to create an MPC using the dynamic model of the vehicle. Alternatively, the learning methods that were briefly mentioned in the previous Section have shown promising results in the control of Ackermann vehicles and have the potential to improve the results for ADTs as well. These learning methods can be utilized to learn about the road conditions, the wear and tear of the vehicle, or even about variables such as the sideslip angle, to better bridge the gap between the vehicle model and the real vehicle.

Bibliography

[1] Abel, D.: *Umdruck zur Vorlesung Regelungstechnik und Ergänzungen (Höhere Regelungstechnik)*. 33. Aufl. Aachen: Verlagshaus Mainz, 2009.

[2] Akegawa, T. ; Ohno, K. ; Kojima, S. ; Yamada, K. ; Go, W. ; Suzuki, T. ; Kiribayashi, S. ; Komatsu, T. ; Miyamoto, N. ; Suzuki, T. ; Shibata, Y. ; Asano, K. ; Nagatani, K. ; Tadokoro, S.: Path Planning of the Turning Back of an Autonomous Large-Scale Six-Wheeled Dump Truck for Loading/Leaving Sediment Based on Backhoe Work. In: *2022 IEEE/SICE International Symposium on System Integration (SII)*. ISSN: 2474-2325. Jan. 2022, pp. 511–518. doi: `10.1109/SII52469.2022.9708879`.

[3] Alshaer, B. ; Darabseh, T. ; Momani, A.: Modelling and control of an autonomous articulated mining vehicle navigating a predefined path. en. In: *International Journal of Heavy Vehicle Systems* 21. no. 2 (2014), p. 152. doi: `10.1504/IJHVS.2014.061640`.

[4] Altafini, C.: Why to use an articulated vehicle in underground mining operations?. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. Vol. 4. ISSN: 1050-4729. May 1999, 3020–3025 vol.4.

[5] Altafini, C.: A Path-Tracking Criterion for an LHD Articulated Vehicle. In: *The International Journal of Robotics Research* 18. no. 5 (May 1999). Publisher: SAGE Publications Ltd STM, pp. 435–441.

[6] Antonov, S. ; Fehn, A. ; Kugi, A.: Unscented Kalman filter for vehicle state estimation. en. In: *Vehicle System Dynamics* 49. no. 9 (Sept. 2011), pp. 1497–1520. doi: `10.1080/00423114.2010.527994`.

[7] Bai, G. ; Liu, L. ; Meng, Y. ; Luo, W. ; Gu, Q. ; Ma, B.: Path Tracking of Mining Vehicles Based on Nonlinear Model Predictive Control. en. In: *Applied Sciences* 9. no. 7 (Jan. 2019). Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 1372. doi: `10.3390/app9071372`.

[8] Bai, G. ; Liu, S. ; Zhou, B. ; Huang, J. ; Zheng, Y. ; Elham, E.: Path Tracking for Electric Mining Articulated Vehicles Based on Nonlinear Compensated Multiple Reference Points Linear MPC. en. In: *World Electric Vehicle Journal* 15. no. 9 (Sept. 2024). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 427. doi: `10.3390/wevj15090427`.

[9] Bai, G. ; Meng, Y. ; Liu, L. ; Luo, W. ; Gu, Q. ; Li, K.: A New Path Tracking Method Based on Multilayer Model Predictive Control. en. In: *Applied Sciences* 9. no. 13 (Jan. 2019). Number: 13 Publisher: Multidisciplinary Digital Publishing Institute, p. 2649.

[10] Bauerle, T. ; Dugdale, Z. ; Poplin, G.: Mineworker fatigue: A review of what we know and future decisions. In: *Mining engineering* 70. no. 3 (Mar. 2018), p. 33.

[11] Bellamy, D. ; Pravica, L.: Assessing the impact of driverless haul trucks in Australian surface mining. en. In: *Resources Policy* 36. no. 2 (June 2011), pp. 149–158. doi: `10.1016/j.resourpol.2010.09.002`.

[12] Benz, D. ; Gehrt, J.-J. ; Zweigel, R. ; Abel, D.: Speed Sensor-Aided Navigation Filter for Robust Localization in GNSS-Denied Mining Environments. en. In: ISSN: 2330-3646. Jan. 2022, pp. 1457–1468. doi: `10.33012/2022.18194`.

[13] Benz, D. ; Gehrt, J.-J. ; Zweigel, R. ; Abel, D.: Speed Sensor-Aided Navigation Filter for Robust Localization in GNSS-Denied Mining Environments. en. In: *NAVIGATION: Journal of the Institute of Navigation* 70. no. 1 (2023), navi.566. doi: `https://doi.org/10.33012/navi.566`.

[14] Benz, D. ; Weseloh, J. ; Abel, D. ; Vallery, H.: CIOT: Constraint-Enhanced Inertial-Odometric Tracking for Articulated Dump Trucks in GNSS-Denied Mining Environments. en. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, pp. 10587–10593. doi: `10.1109/ICRA48891.2023.10160664`.

[15] Bhatti, G. ; Mohan, H. ; Raja Singh, R.: Towards the future of smart electric vehicles: Digital twin technology. In: *Renewable and Sustainable Energy Reviews* 141 (May 2021), p. 110801. doi: `10.1016/j.rser.2021.110801`.

[16] Borrelli, F. ; Bemporad, A. ; Morari, M.: *Predictive Control for Linear and Hybrid Systems*. en. Google-Books-ID: 7NUoDwAAQBAJ. Cambridge, UK: Cambridge University Press, June 2017.

[17] Brown, R. G. ; Hwang, P. Y. C.: *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. 3rd ed. New York: Wiley, 1997.

[18] Brundtland, G. H.: Our Common Future—Call for Action. en. In: *Environmental Conservation* 14. no. 4 (Jan. 1987). Publisher: Cambridge University Press, pp. 291–294. doi: `10.1017/S0376892900016805`.

[19] Caldwell, J. ; Marshall, J. A.: Towards Efficient Learning-Based Model Predictive Control via Feedback Linearization and Gaussian Process Regression. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Sept. 2021, pp. 4306–4311. doi: `10.1109/IROS51168.2021.9636755`.

[20] Chen, B.-C. ; Hsieh, F.-C.: Sideslip angle estimation using extended Kalman filter. en. In: *Vehicle System Dynamics* 46. no. sup1 (Sept. 2008), pp. 353–364. doi: `10.1080/00423110801958550`.

[21] Chen, C. ; Tomizuka, M.: Modeling And Control Of Articulated Vehicles. en. In: (1997).

[22] Chen, X. ; Cheng, J. ; Hu, H. ; Shao, G. ; Gao, Y. ; Zhu, Q.: A Novel Fuzzy Logic Switched MPC for Efficient Path Tracking of Articulated Steering Vehicles. en. In: *Robotics* 13. no. 9 (Sept. 2024). Number: 9 Publisher: MDPI AG, pp. 134–134. doi: `10.3390/robotics13090134`.

[23] Cui, Y. ; Peng, L. ; Li, H.: Filtered Probabilistic Model Predictive Control-Based Reinforcement Learning for Unmanned Surface Vehicles. In: *IEEE Transactions on Industrial Informatics* 18. no. 10 (Oct. 2022). Conference Name: IEEE Transactions on Industrial Informatics, pp. 6950–6961. doi: `10.1109/TII.2022.3142323`.

[24] Dekker, L. G. ; Marshall, J. A. ; Larsson, J.: Experiments in feedback linearized iterative learning-based path following for center-articulated industrial vehicles. en. In: *Journal of Field Robotics* 36. no. 5 (2019), pp. 955–972.

[25] Delrobaei, M. ; McIsaac, K. A.: Design and Steering Control of a Center-Articulated Mobile Robot Module. en. In: *Journal of Robotics* (), p. 15.

[26] DeSantis, R. M.: Modeling and Path-Tracking for a Load-Haul-Dump Mining Vehicle. en. In: *Journal of Dynamic Systems, Measurement, and Control* 119. no. 1 (Mar. 1997), pp. 40–47. doi: `10.1115/1.2801212`.

[27] Dou, F. ; Liu, W. ; Huang, Y. ; Liu, L. ; Meng, Y.: Modeling and path tracking for articulated steering vehicles. In: *2017 Chinese Automation Congress (CAC)*. Oct. 2017, pp. 5263–5268. doi: `10.1109/CAC.2017.8243715`.

[28] Doumiati, M. ; Victorino, A. C. ; Charara, A. ; Lechner, D.: Onboard Real-Time Estimation of Vehicle Lateral Tire–Road Forces and Sideslip Angle. In: *IEEE/ASME Transactions on Mechatronics* 16. no. 4 (Aug. 2011). Conference Name: IEEE/ASME Transactions on Mechatronics, pp. 601–614. doi: `10.1109/TMECH.2010.2048118`.

[29] Dubiński, J.: Sustainable Development of Mining Mineral Resources. en. In: *Journal of Sustainable Mining* 12. no. 1 (2013), pp. 1–6. doi: `10.7424/jsm130102`.

[30] earth, G.: *Map of a mining field close to Steyrling, Austria.*

[31] Ferrein, A. ; Reke, M. ; Scholl, I. ; Decker, B. ; Limpert, N. ; Nikolovski, G. ; Schiffer, S.: *Towards a Fleet of Autonomous Haul-Dump Vehicles in Hybrid Mines*. Pages: 288. Jan. 2023. doi: `10.5220/0011693600003393`.

[32] Gao, L. ; Dong, Y. ; Zhao, J.: Dynamic Modeling and Characteristic Analysis of Articulated Steering Vehicles. en. In: *Applied Sciences* 13. no. 8 (Jan. 2023). Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 5099. doi: `10.3390/app13085099`.

[33] Gao, Y. ; Shen, Y. ; Xu, T. ; Zhang, W. ; Güvenç, L.: Oscillatory Yaw Motion Control for Hydraulic Power Steering Articulated Vehicles Considering the Influence of Varying Bulk Modulus. In: *IEEE Transactions on Control Systems Technology* 27. no. 3 (May 2019). Conference Name: IEEE Transactions on Control Systems Technology, pp. 1284–1292. doi: `10.1109/TCST.2018.2803746`.

[34] *Gazebo*.

[35] Gros, S. ; Zanon, M. ; Quirynen, R. ; Bemporad, A. ; Diehl, M.: From linear to nonlinear MPC: bridging the gap via the real-time iteration. In: *International Journal of Control* 93. no. 1 (Jan. 2020). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207179.2016.1222553, pp. 62–80. doi: `10.1080/00207179.2016.1222553`.

[36] He, Y. ; Khajepour, A. ; McPhee, J. ; Wang, X.: Dynamic modelling and stability analysis of articulated frame steer vehicles. en. In: *International Journal of Heavy Vehicle Systems* (Nov. 2004). Publisher: Inderscience Publishers.

[37] Hejase, B. ; Ozguner, U.: Physics-Based Simulation and Automation of a Load-Haul-Dump Operation for an Articulated Dump Truck. en. In: *Vehicles* 4. no. 1 (Mar. 2022). Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, pp. 167–181. doi: `10.3390/vehicles4010011`.

[38] Hemami, A. ; Polotski, V.: Dynamics of a dual-unit articulated vehicle for path tracking control problem formulation. In: *Proceedings of the 1997 IEEE International Conference on Control Applications*. Oct. 1997, pp. 173–176. doi: `10.1109/CCA.1997.627533`.

[39] Hemami, A. ; Polotski, V.: Path Tracking Control Problem Formulation of an LHD Loader. In: *The International Journal of Robotics Research* 17. no. 2 (Feb. 1998). Publisher: SAGE Publications Ltd STM, pp. 193–199. doi: `10.1177/027836499801700206`.

[40] Hewing, L. ; Kabzan, J. ; Zeilinger, M. N.: Cautious Model Predictive Control Using Gaussian Process Regression. en. In: *IEEE Transactions on Control Systems Technology* 28. no. 6 (Nov. 2020), pp. 2736–2743. doi: `10.1109/TCST.2019.2949757`.

[41] Humphreys, D.: Mining productivity and the fourth industrial revolution. en. In: *Mineral Economics* 33. no. 1 (July 2020), pp. 115–125. doi: `10.1007/s13563-019-00172-9`.

[42] Jang, H. ; Topal, E.: Transformation of the Australian mining industry and future prospects. en. In: *Mining Technology* 129. no. 3 (Sept. 2020). Publisher: SAGE Publications, pp. 120–134. doi: `10.1080/25726668.2020.1786298`.

[43] Jeong, Y.: Integrated Vehicle Controller for Path Tracking with Rollover Prevention of Autonomous Articulated Electric Vehicle Based on Model Predictive Control. en. In: *Actuators* 12. no. 1 (Jan. 2023), p. 41.

[44] *Kalman Filtering and Neural Networks*. eng. 1., Auflage. OCLC: 904829704. New York, NY: John Wiley & Sons, 2004.

[45] Katriniok, A.: *Optimal vehicle dynamics control and state estimation for a low-cost GNSS-based collision avoidance system*. de. Als Ms. gedr. Fortschritt-Berichte VDI Reihe 8, Mess-, Steuerungs- und Regelungstechnik 1230. Düsseldorf: VDI-Verl, 2014.

[46] Kim, H. H. ; Ryu, J.: Sideslip angle estimation considering short-duration longitudinal velocity variation. en. In: *International Journal of Automotive Technology* 12. no. 4 (Aug. 2011), pp. 545–553. doi: `10.1007/s12239-011-0064-2`.

[47] Kroon, D.: *2D Line Curvature and Normals. MATLAB Central File Exchange.* 2011.

[48] Lee, T. ; Jeong, Y.: A Tube-Based Model Predictive Control for Path Tracking of Autonomous Articulated Vehicle. en. In: *Actuators* 13. no. 5 (May 2024). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 164. doi: `10.3390/act13050164`.

[49] Li, F. ; Li, H. ; Wu, C.: Gaussian Process Based Learning Model Predictive Control With Application to USV. In: *IEEE Transactions on Industrial Electronics* 71. no. 12 (Dec. 2024). Conference Name: IEEE Transactions on Industrial Electronics, pp. 16388–16397. doi: `10.1109/TIE.2024.3384617`.

[50] Luijten, M.: *Lateral dynamic behaviour of articulated commercial vehicles*. PhD thesis. Eindhoven University of Technology, 2010.

[51] Lynas, D.: Human Factor Issues with Automated Mining Equipment. en. In: *The Ergonomics Open Journal* 4. no. 1 (Aug. 2011), pp. 74–80. doi: `10.2174/1875934301104010074`.

[52] maps, A.: *Map of a mining field close to Buir, Germany.*

[53] Marouki, F.: *Optimization of Path and Trajectory for Underground Mining Machines*. eng. 2023.

[54] Meng, Q. ; Gao, L. ; Xie, H. ; Dou, F.: Analysis of the Dynamic Modeling Method of Articulated Vehicles. In: *Journal of Engineering Science and Technology Review* 10 (July 2017), pp. 18–27. doi: `10.25103/jestr.103.04`.

[55] Meng, Y. ; Gan, X. ; Wang, Y. ; Gu, Q.: LQR-GA Controller for Articulated Dump Truck Path Tracking System. en. In: *Journal of Shanghai Jiaotong University (Science)* 24. no. 1 (Feb. 2019), pp. 78–85. doi: `10.1007/s12204-018-2012-z`.

[56] Morin, P. ; Samson, C.: Motion Contr 34. Motion Control of Wheeled Mobile Robots. en. In: (2008).

[57] Nayl, T. ; Nikolakopoulos, G. ; Gustfsson, T.: Switching model predictive control for an articulated vehicle under varying slip angle. In: *2012 20th Mediterranean Conference on Control Automation (MED)*. July 2012, pp. 890–895. doi: `10.1109/MED.2012.6265751`.

[58] Nayl, T.: *Modeling, control and path planning for an articulated vehicle*. en. ISBN: 9789174396683 OCLC: 940736756. PhD thesis. Luleå: Department of Computer Science, electrical and space engineering, Luleå University of technology, 2013.

[59] Nayl, T. ; Nikolakopoulos, G. ; Gustafsson, T.: A full error dynamics switching modeling and control scheme for an articulated vehicle. en. In: *International Journal of Control, Automation and Systems* 13. no. 5 (Oct. 2015), pp. 1221–1232. doi: `10.1007/s12555-014-0049-9`.

[60] Neuseon, W.: *Raddumper 1501 online konfigurieren | Wacker Neuson Shop.*

[61] Newport Beach, C. M. S. C.: *MSC Adams.*

[62] Nikolovski, G. ; Limpert, N. ; Nessau, H. ; Reke, M. ; Ferrein, A.: Model-predictive Control with Parallelised Optimisation for the Navigation of Autonomous Mining Vehicles. In: *2023 IEEE Intelligent Vehicles Symposium (IV)*. ISSN: 2642-7214. June 2023, pp. 1–6. doi: `10.1109/IV55152.2023.10186806`.

[63] Pacejka, H.: *Tire and Vehicle Dynamics*. en. Google-Books-ID: JCHDuED3WIkC. Elsevier, Dec. 2005.

[64] Pacejka, H. B. ; Bakker, E.: THE MAGIC FORMULA TYRE MODEL. en. In: *Vehicle System Dynamics* 21. no. sup001 (Jan. 1992), pp. 1–18. doi: `10.1080/00423119208969994`.

[65] Pazooki, A. ; Rakheja, S. ; Cao, D.: Kineto-dynamic directional response analysis of an articulated frame steer vehicle. en. In: *International Journal of Vehicle Design* 65. no. 1 (2014), p. 1. doi: `10.1504/IJVD.2014.060063`.

[66] Petrov, P. ; Bigras, P.: A practical approach to feedback path control for an articulated mining vehicle. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*. Vol. 4. Oct. 2001, 2258–2263 vol.4. doi: `10.1109/IROS.2001.976406`.

[67] Piyabongkarn, D. ; Rajamani, R. ; Grogg, J. A. ; Lew, J. Y.: Development and Experimental Evaluation of a Slip Angle Estimator for Vehicle Stability Control. In: *IEEE Transactions on Control Systems Technology* 17. no. 1 (Jan. 2009). Conference Name: IEEE Transactions on Control Systems Technology, pp. 78–88. doi: `10.1109/TCST.2008.922503`.

[68] Rajamani, R.: *Vehicle Dynamics and Control*. en. Mechanical Engineering Series. Boston, MA: Springer US, 2012. doi: `10.1007/978-1-4614-1433-9`.

[69] Reina, G. ; Underwood, J. ; Brooker, G. ; Durrant-Whyte, H.: Radar-based perception for autonomous outdoor vehicles. en. In: *Journal of Field Robotics* 28. no. 6 (2011), pp. 894–913. doi: `10.1002/rob.20393`.

[70] Ridley, P. ; Corke, P.: Load Haul Dump Vehicle Kinematics and Control. en. In: *Journal of Dynamic Systems, Measurement, and Control* 125. no. 1 (Mar. 2003), pp. 54–59. doi: `10.1115/1.1541671`.

[71] Rogers, W. P. ; Kahraman, M. M. ; Drews, F. A. ; Powell, K. ; Haight, J. M. ; Wang, Y. ; Baxla, K. ; Sobalkar, M.: Automation in the Mining Industry: Review of Technology, Systems, Human Factors, and Political Risk. en. In: *Mining, Metallurgy & Exploration* 36. no. 4 (Aug. 2019), pp. 607–631. doi: `10.1007/s42461-019-0094-2`.

[72] Rokonuzzaman, M. ; Mohajer, N. ; Nahavandi, S. ; Mohamed, S.: Model Predictive Control With Learned Vehicle Dynamics for Autonomous Vehicle Path Tracking. In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 128233–128249. doi: `10.1109/ACCESS.2021.3112560`.

[73] Sam-Aggrey, H.: Assessment Of The Impacts Of New Mining Technologies: Recommendations On The Way Forward. In: *WIT Transactions on Ecology and the Environment* (2020).

[74] Scheding, S. ; Dissanayake, G. ; Nebot, E. M. ; Durrant-Whyte, H.: An experiment in autonomous navigation of an underground mining vehicle. In: *IEEE Transactions on Robotics and Automation* 15. no. 1 (Feb. 1999). Conference Name: IEEE Transactions on Robotics and Automation, pp. 85–95. doi: `10.1109/70.744605`.

[75] Schmid, C. ; Biegler, L.: Quadratic programming methods for reduced hessian SQP. en. In: *Computers & Chemical Engineering* 18. no. 9 (Sept. 1994), pp. 817–832. doi: `10.1016/0098-1354(94)E0001-4`.

[76] Schramm, D. ; Hiller, M. ; Bardini, R.: *Vehicle Dynamics*. en. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. doi: `10.1007/978-3-662-54483-9`.

[77] Selmanaj, D. ; Corno, M. ; Panzani, G. ; Savaresi, S. M.: Vehicle sideslip estimation: A kinematic based approach. en. In: *Control Engineering Practice* 67 (Oct. 2017), pp. 1–12. doi: `10.1016/j.conengprac.2017.06.013`.

[78] Song, R. ; Ye, Z. ; Wang, L. ; He, T. ; Zhang, L.: Autonomous Wheel Loader Trajectory Tracking Control Using LPV-MPC. In: *2022 American Control Conference (ACC)*. Atlanta, GA, USA: IEEE, June 2022, pp. 2063–2069.

[79] Steele, J. P. H. ; Ganesh, C. ; Kleve, A.: Control and scale model simulation of sensor-guided LHD mining machines. In: *IEEE Transactions on Industry Applications* 29. no. 6 (Nov. 1993). Conference Name: IEEE Transactions on Industry Applications, pp. 1232–1238. doi: `10.1109/28.259737`.

[80] Stoerkle, J.: *Lateral dynamics of multiaxle vehicles*. PhD thesis. Swiss Federal Institute of Technology (ETH) Zurich, 2013.

[81] Sun, N. ; Zhang, W. ; Yang, J.: Integrated Path Tracking Controller of Underground Articulated Vehicle Based on Nonlinear Model Predictive Control. en. In: *Applied Sciences* 13. no. 9 (Jan. 2023). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 5340. doi: `10.3390/app13095340`.

[82] Tabatabaei Oreh, S. H. ; Kazemi, R. ; Azadi, S.: A sliding-mode controller for directional control of articulated heavy vehicles. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 228. no. 3 (Feb. 2014). Publisher: IMECHE, pp. 245–262. doi: `10.1177/0954407013503628`.

[83] Tan, S. ; Zhao, X. ; Yang, J. ; Zhang, W.: A path tracking algorithm for articulated vehicle: Development and simulations. In: *2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*. Aug. 2017, pp. 1–6. doi: `10.1109/ITEC-AP.2017.8080797`.

[84] Thrun, S. ; Montemerlo, M. ; Dahlkamp, H. ; Stavens, D. ; Aron, A. ; Diebel, J. ; Fong, P. ; Gale, J. ; Halpenny, M. ; Hoffmann, G. ; Lau, K. ; Oakley, C. ; Palatucci, M. ; Pratt, V. ; Stang, P. ; Strohband, S. ; Dupont, C. ; Jendrossek, L.-E. ; Koelen, C. ; Markey, C. ; Rummel, C. ; Niekerk, J. van ; Jensen, E. ; Alessandrini, P. ; Bradski, G. ; Davies, B. ; Ettinger, S. ; Kaehler, A. ; Nefian, A. ; Mahoney, P.: Stanley: The robot that won the DARPA Grand Challenge. en. In: *Journal of Field Robotics* 23. no. 9 (2006). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20147, pp. 661–692. doi: `10.1002/rob.20147`.

[85] Trimboli, S. ; Di Cairano, S. ; Bemporad, A. ; Kolmanovsky, I. V.: Model predictive control for automotive time-delay processes: An application to air-to-fuel ratio control*. In: *IFAC Proceedings Volumes*. 8th IFAC Workshop on Time-Delay Systems 42. no. 14 (Jan. 2009), pp. 90–95. doi: `10.3182/20090901-3-RO-4009.00012`.

[86] Trucks, B.: *Bell B30E ADT*. en-US.

[87] Vallery, H.: *Umdruck zur Vorlesung Regelungstechnik und Ergänzungen (Höhere Regelungstechnik)*. ger. 47. Auflage. Aachen: Verlagshaus Mainz, 2023.

[88] Vallinder, G.: *Path Following Model Predictive Control for Center-Articulated Vehicles.*

[89] Voss, W.: *A Comprehensible Guide to Controller Area Network.* en. Google-Books-ID: PU6ppO3XbUwC. Copperhill Media, 2008.

[90] Wang, H. ; Li, G. ; Hou, J. ; Chen, L. ; Hu, N.: A Path Planning Method for Underground Intelligent Vehicles Based on an Improved RRT* Algorithm. en. In: *Electronics* 11. no. 3 (Jan. 2022). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 294. doi: `10.3390/electronics11030294`.

[91] Wang, Y. ; Liu, X. ; Ren, Z. ; Yao, Z. ; Tan, X.: Synchronized path planning and tracking for front and rear axles in articulated wheel loaders. In: *Automation in Construction* 165 (Sept. 2024), p. 105538. doi: `10.1016/j.autcon.2024.105538`.

[92] Wang, Z. ; Ma, X. ; Bao, H.: Research on Trajectory Tracking Control for Automatic Driving of Articulated Scraper. In: *2023 8th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS).* Okinawa, Japan: IEEE, Nov. 2023, pp. 402–408. doi: `10.1109/ICIIBMS60103.2023.10347772`.

[93] Wenzel, T. A. ; Burnham, K. J. ; Blundell, M. V. ; Williams, R. A.: Dual extended Kalman filter for vehicle state and parameter estimation. en. In: *Vehicle System Dynamics* 44. no. 2 (Feb. 2006), pp. 153–171. doi: `10.1080/00423110500385949`.

[94] Winkle, T.: Safety Benefits of Automated Vehicles: Extended Findings from Accident Research for Development, Validation and Testing. en. In: *Autonomous Driving: Technical, Legal and Social Aspects.* Ed. by Maurer, M. ; Gerdes, J. C. ; Lenz, B. ; Winner, H. Berlin, Heidelberg: Springer, 2016, pp. 335–364. doi: `10.1007/978-3-662-48847-8_17`.

[95] Yang, F. ; Wang, J. ; Zhang, C. ; Li, J. ; Xie, H. ; Zhuoge, Z.: The Impact of Human Activities on Net Primary Productivity in a Grassland Open-Pit Mine: The Case Study of the Shengli Mining Area in Inner Mongolia, China. en. In: *Land* 11. no. 5 (May 2022). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 743. doi: `10.3390/land11050743`.

[96] Yang, M. ; Bian, Y. ; Liu, G. ; Zhang, H.: Path Tracking Control of an Articulated Road Roller With Sideslip Compensation. In: *IEEE Access* 8 (2020). Conference Name: IEEE Access, pp. 127981–127992. doi: `10.1109/ACCESS.2020.3008455`.

[97] Zhang, M. ; Kecojevic, V. ; Komljenovic, D.: Investigation of haul truck-related fatal accidents in surface mining using fault tree analysis. en. In: *Safety Science* 65 (June 2014), pp. 106–117. doi: `10.1016/j.ssci.2014.01.005`.

[98] Zhang, X.: Study on Ride Comfort of the Articulated Dump Truck. In: *2018 5th International Conference on Information Science and Control Engineering (ICISCE).* July 2018, pp. 527–532. doi: `10.1109/ICISCE.2018.00116`.

[99] Zhang, Y. ; Wang, Y. ; Wu, M. ; Li, Z. ; Li, R.: Two-Stage Dichotomy Hybrid A* for Path Planning of Unmanned Mining Trucks in Large-Scale Open-Pit Mine. In: *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. ISSN: 2159-6255. July 2024, pp. 296–301. doi: `10.1109/AIM55361.2024.10637142`.