

Driver assistance for steering while reversing long combination vehicles

TME180 Automotive Engineering Project, HT 2025

Anirudh Sreeram
Chaojie Lu
Harshith Pidur Kuppusamy
Junyu Zhao
Nihar Prakash Mehta
Zichen Huang

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2026

Driver assistance for steering while reversing long combination vehicles

TME180 Automotive Engineering Project, HT 2025

© Anirudh Sreeram, Chaojie Lu, Harshith Pidur Kuppusamy, Junyu Zhao, Nihar Prakash Mehta, Zichen Huang, 2026

Supervisor: Bengt Jacobson, Vehicle Dynamics group, Mechanics and Maritime Sciences

Supervisor: Zhaohui Ge, Vehicle Dynamics group, Mechanics and Maritime Sciences

Examiner: Alexey Vdovin, Mechanics and Maritime Sciences

Studentarbeten – Mekanik och maritima vetenskaper (M2) – Projektarbete

Department of Mechanics and Maritime Sciences

Chalmers University of Technology

SE-412 96 Göteborg

Sweden

Telephone +46 (0)31 772 1000

Abstract

Long combination vehicles, such as A-double configurations, pose significant challenges during low-speed reversing due to their length and multiple articulation points, which increase the risk of instability and jackknifing. This project investigates a driver-assist approach for reverse steering of such vehicles, aiming to improve controllability during low-speed manoeuvres.

A desktop simulation framework was developed using TruckMaker for vehicle modelling and MATLAB/Simulink for control implementation and visualisation. The reversing behaviour was described using a kinematic single-track model for articulated vehicles, and an LQR-based feedback controller was designed to stabilise articulation angles and regulate the reversing trajectory. Driver input is provided through a knob-based interface specifying the desired turning radius, which is used to calculate steering commands for the tractor and the rear trailer.

The system was evaluated in simulation through multiple reversing scenarios, including straight-line, constant-radius, and misaligned initial conditions, demonstrating stable reversing behaviour without jackknifing and consistent path tracking under the tested conditions. In addition, a CAN communication framework was implemented to explore real-vehicle integration. While vehicle state signals were successfully received in real time, steering command transmission could not yet be validated due to gateway limitations. The project establishes a simulation and HIL-ready framework for future refinement and real-vehicle validation.

Keywords: Vehicle Dynamics, Long combination vehicles, A-doubles, Reverse steering, LQR-based feedback controller, Driver-assist, HMI

Acknowledgements

We express our deepest gratitude to our supervisors at Chalmers University of Technology, Professor Bengt Jacobson and Zhaohui Ge, who are in many ways the architects of this project. Their expert guidance, planning, valuable feedback, and unwavering support have been instrumental in every stage of this project. Their encouragement and structured collaboration helped us overcome all the challenges and achieve concrete, measurable results.

Our heartfelt thanks go to Pavan Kumar Adiga Nagaraj from Volvo Trucks for his industrial perspective, insightful advice, and the opportunity to work on real-world challenges. This report largely builds on the work begun by him and Nivedita Krishnakumar, also from Volvo. Tommi Saarikoski from Volvo GTT also has our thanks for the wonderful day at Hällered Proving Ground.

We would like to further thank Fredrik von Corswant and Daniel Poveda Pi from the REVERE laboratory for their constructive discussions, technical assistance, access to their trucks, and generous help throughout this project. Their input and collaboration were vital in troubleshooting and moving forward in every step of our project.

Furthermore, we appreciate the support we received from Alexander Hagglund and Shreyas Kogalur from IPG Automotive, Sweden. Their support and constant involvement, often requested unscheduled and at short notice, was instrumental in setting up the vehicle model on IPG TruckMaker. We also thank Elektroteknologsektionens teletekniska avdelning (ETA) for their facilities, where we produced our rotary knobs.

Lastly, we are grateful to our friends and families for their constant support during the course of this project.

Anirudh Sreeram
Chaojie Lu
Harshith Pidur Kuppusamy
Junyu Zhao
Nihar Prakash Mehta
Zichen Huang

Gothenburg, January 2026

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this project.

Indices

- i Vehicle unit number
- j axle number

Parameters

- L_1 Wheelbase
- w Trackwidth
- b_1 Distance from hitch 1 to tractor's rear axle
- L_2 Distance from hitch 1 to Semi-Trailer 1 rear axle
- b_2 Distance from rear axle of Semi-Trailer 1 to hitch 2
- L_3 Distance from hitch 2 to axle of dolly
- b_3 Distance from axle of dolly to hitch 3
- L_4 Distance from hitch 3 to rear axle of semi-trailer
- M_1 Mass of the tractor
- M_2 Mass of Semi-Trailer 1
- M_3 Mass of dolly
- M_4 Mass of Semi-Trailer 2

Variables

- δ_1 Steering angle of the tractor [rad]
- δ_2 Steering angle of the semi-trailer-2 [rad]
- v_{1x} Longitudinal velocity of the tractor [m/s]
- ϕ_1 Yaw angle of the tractor [rad]
- ϕ_2 Yaw angle of the semi-trailer-1 [rad]
- ϕ_3 Yaw angle of the dolly [rad/s]
- ϕ_4 Yaw angle of the semi-trailer-2 [rad]
- θ_1 Articulation angle between the tractor and the semi-trailer-1 [rad/s]
- θ_2 Articulation angle between the semi-trailer-1 and the dolly [rad/s]
- θ_3 Articulation angle between the dolly and the semi-trailer-2 [rad/s]
- X_{11} X coordinate of the front axle of the tractor [m]
- Y_{11} Y coordinate of the front axle of the tractor [m]
- X_{12} X coordinate of the rear axle of the tractor [m]
- Y_{12} Y coordinate of the rear axle of the tractor [m]
- X_{21} X coordinate of the lumped axle of the semi-trailer-1 [m]
- Y_{21} Y coordinate of the lumped axle of the semi-trailer-1 [m]
- X_{31} X coordinate of the lumped axle of the dolly [m]
- Y_{31} Y coordinate of the lumped axle of the dolly [m]
- X_{41} X coordinate of the lumped axle of the semi-trailer-2 [m]
- Y_{41} Y coordinate of the lumped axle of the semi-trailer-2 [m]
- X_{p1} X coordinate of the rear coupling point on the tractor unit [m]
- Y_{p1} Y coordinate of the rear coupling point on the tractor unit [m]
- X_{p2} X coordinate of the rear coupling point on the semi-trailer-1 unit [m]
- Y_{p2} Y coordinate of the rear coupling point on the semi-trailer-1 unit [m]
- X_{p3} X coordinate of the rear coupling point on the dolly unit [m]
- Y_{p3} Y coordinate of the rear coupling point on the dolly unit [m]

Contents

1	Introduction	7
1.1	Background	7
1.2	Literature review	8
1.3	Goals	8
2	Methodology	9
2.1	Vehicle Modelling	9
2.1.1	Vehicle Model on TruckMaker	9
2.1.2	Simulink Model	10
2.2	Knob module implementation	12
2.2.1	Virtual knob in MATLAB Simulink	12
2.2.2	Physical knob - v1	12
2.2.3	Knob model - v2	15
2.3	Controller and mathematical modelling	16
2.3.1	Single-Track Kinematic Model	16
2.3.2	LQR-Based Feedback Controller	18
2.4	Visualisation	18
2.5	CAN module implementation	19
2.5.1	CAN communication framework and DBC file	19
2.5.2	Simulink CAN interface model	19
2.5.3	Real-vehicle test environment	20
3	Results	21
3.1	Desktop Simulator	21
3.1.1	Straight line reverse	21
3.1.2	Constant high radius	23
3.1.3	Constant low radius	25
3.1.4	90° reverse	27
3.1.5	Misaligned units	29
3.1.6	Reverse parking manoeuvre	31
3.2	Real vehicle test and CAN communication	33
3.2.1	Signal reception (successful)	33
3.2.2	Steering request transmission (pending)	33
4	Test day at Hällered Proving Ground	34
5	Conclusion and Future Work	36
5.1	Conclusion	36
5.2	Future work	36
5.2.1	Evaluate alternative control strategies like MPC	36
5.2.2	Dynamic model refinement	36
5.2.3	Articulation angle estimation	36
5.2.4	Improvements to the visualisation display	37
5.2.5	Real vehicle testing and deployment	37

1 Introduction

1.1 Background

Long-combination heavy vehicles improve road freight efficiency. One such combination is an A-double, which consists of a tractor unit, a first semi-trailer, a dolly, and a second semi-trailer, enabling two standard 13.6 m trailers to be hauled in one trip without changing the basic loading unit. Since 2023, Sweden has allowed these A-double combinations, approximately 34.5 m in length and with a maximum load of 74 tonnes, to operate on designated routes. This has been supported by test-track and real road evaluations to ensure safety and manoeuvrability. Similar trials have been launched in Finland, the Netherlands, Spain, and Germany, while Australia, New Zealand, and parts of Canada have incorporated A-double configurations within their Performance-Based Standards (PBS) frameworks. Compared with traditional semi-trailer combinations, the A-double offers substantial benefits in productivity and cost efficiency, enabling one driver to transport nearly twice the load, reducing total vehicle-kilometres travelled, and aligning with existing port and rail intermodal systems better. It has therefore become a promising measure to enhance freight capacity and corridor-level logistics efficiency.



Figure 1: A-double scaled model (REVERE)

Despite these advantages, A-double vehicles still face significant challenges during low-speed operations in ports, docks, and loading and unloading zones. Their long, multi-section structure makes shunting, reversing, and manoeuvring in confined spaces very complex and demanding for drivers; reversing into a logistics hub, or even just to park, can take several minutes. Even slight steering errors can lead to trajectory deviation or instability, and relying solely on the front vehicle's steering is usually insufficient to control the rear trailer's path even at very low speeds. These manoeuvrability issues currently limit the widespread use and operational safety of double trailer combinations. In this work, we will focus on the problem of steering while reversing these combinations stably.

1.2 Literature review

This section will discuss previous work on trailer reverse assistance.

Van de Wouw and Ritzen (2015) approached the low-speed trailer steering issue by modelling it as a kinematic tracking task of an off-axle tractor-trailer system. They developed a nonlinear control law that guides the trailer's rear trajectory to coincide with the path traced by the tractor's front axle. Their proposed control strategy effectively narrowed the swept-path envelope and mitigated tail-swing behaviour, which was confirmed by simulation under typical cornering manoeuvres [1].

Roempke and Liu (2024) using steer-by-wire, prototyped a steer-by-wire trailer backup assist: driver inputs (knob/steering-wheel modes) were mapped to hitch-angle or hitch-rate commands, with real-time estimation and torque feedback to prevent jackknifing and improve reversing accuracy. Simulation and vehicle tests have shown faster, more precise manoeuvres for novice and experienced drivers [2].

Raad et al. (2016) patented a practical trailer backup assistance system that integrates hitch-angle measurement with dynamically adjusted steering limits determined by the trailer's geometry and articulation rate. By constraining steering commands within these adaptive bounds, the system preserves the desired curvature of motion and prevents jackknifing, representing a commercial realisation of control principles originally developed in academic research [3].

Wang (2015) designed and experimentally verified an Active Trailer Steering (ATS) scheme for double-trailer heavy vehicles through a driver-hardware-in-the-loop real-time simulation platform. The study demonstrated that ATS markedly enhances both low-speed path tracking and high-speed lateral stability. Moreover, a model-reference adaptive control method was proposed to maintain consistent performance under different speeds and loading conditions. ISO-14791 closed-loop SLC has been recommended for RA assessment [4].

Ge et al. (2024) revisited the problem of reversing long combination vehicles from a kinematic perspective, emphasising how articulation-angle evolution shapes feasible motion. They noted that the commonly used SSCL boundary provides only a partial picture, particularly for vehicle sets with multiple joints. By introducing the concepts of CAARP and LAARP, the study distinguishes between states where articulation angles remain steerable and those where only limited backward travel is possible. Their simulations indicate that usable reversing configurations may extend beyond SSCL, broadening the basis for planning and control strategies [5].

1.3 Goals

- This project focuses on controlling the steering system of A-double vehicles while reversing. Specifically, we build on the work started by Adiga and Krishnakumar (2025) in the thesis '*Reverse Assist Steering Control for Long Combination Vehicles*' [6]. They proposed utilising steered axles on the last trailing unit in combination with the regular steered axle at the tractor, with angles for both decided in real time by a controller of their own design. Within the scope of this project, we focus on building a Simulink-based desktop simulator with such a controller, integrated with a physical rotary knob for driver input. The manoeuvre would be simulated on TruckMaker. The user is required to input the desired radius using this physical knob, based on observing the vehicle real-time via the visualisation module and visuals from TruckMaker's Movie NX.
- Based on performance in SIL/HIL simulations, an attempt on VIL simulations was made at REVERE. A CAN communication module was added on Simulink to establish real-time communication between the desktop simulator and the real-vehicle. This is a preparation to set up a real vehicle version of the simulator where different controllers and HMI could be subjectively evaluated and demonstrated.

2 Methodology

We begin building a simulation environment by using IPG TruckMaker 14 for vehicle modelling and MATLAB/Simulink for implementing and testing the control algorithms and creating real-time visualisation of the manoeuvre which would be used to display on the infotainment screen. We use a physical knob that can dynamically control the desired turning radius. This setup is used to make multiple test manoeuvres to analyse the performance of the system.

Upon satisfactory results from the SIL/HIL tests, we then attempted to integrate this solution with a real-vehicle and virtual trailer. In this report, we will present the progress we have made with the desktop simulator and the real vehicle test. The modules in this project can be broadly classified into the following sub-sections:

1. Vehicle model on TruckMaker
2. Knob module implementation
3. Controller and Mathematical Modelling
4. Visualisation
5. CAN module implementation

2.1 Vehicle Modelling

2.1.1 Vehicle Model on TruckMaker

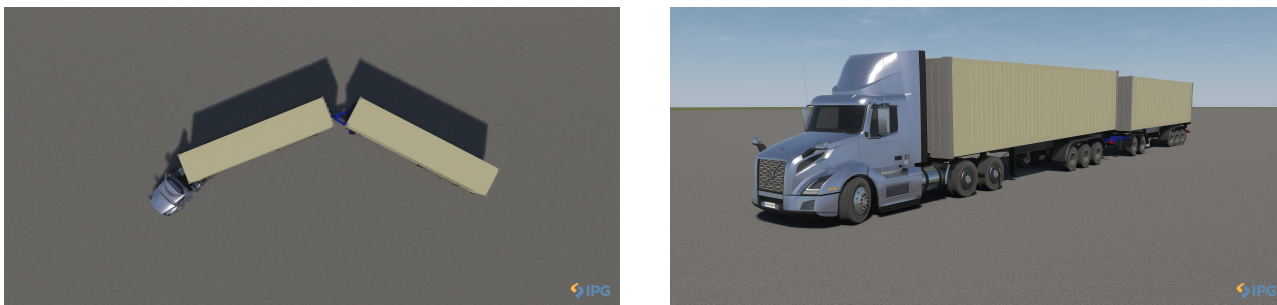


Figure 2: A-double on TruckMaker

For this project, we use the FH64 Tractor Electric as our base model for the tractor. Since the exact model is not available on TruckMaker, we use one of the existing models and adjust the dimensions. The parameters are as stated in table. Here, we made lumped axle assumptions on all the units. This vehicle model has 3 axles on the tractor (6x4), 3 axles on each semi-trailer, and 2 axles on the dolly. The axle spread is 1.25 m. This model will be used to test the Simulink model for various reversing manoeuvres.

Parameters	Description	Value	Unit
L_1	Wheelbase	4.585	m
w	Trackwidth	2.076	m
b_1	Distance from hitch 1 to tractor's lumped rear axle	0.409	m
L_2	Distance from hitch 1 to Semi-Trailer 1 lumped rear axle	9	m
b_2	Distance from lumped rear axle of Semi-Trailer 1 to hitch 2	3	m
L_3	Distance from hitch 2 to lumped axle of dolly	3.5	m
b_3	Distance from lumped axle of dolly to hitch 3	0.4	m
L_4	Distance from hitch 3 to lumped rear axle of semi-trailer	9	m
M_1	Mass of the tractor	6473	kg
M_2	Mass of Semi-Trailer 1	11050	kg
M_3	Mass of dolly	3700	kg
M_4	Mass of Semi-Trailer 2	11050	kg

Table 1: Vehicle parameters

2.1.2 Simulink Model

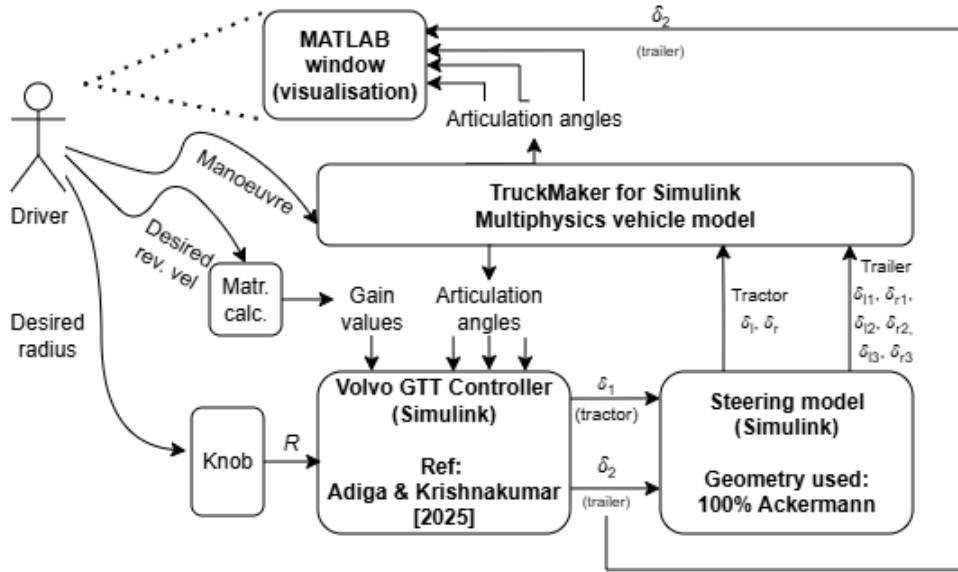


Figure 3: Flowchart of Simulink Model

Based on the vehicle parameters and vehicle speed, an LQR is used to determine the gain values on the controller which are computed and stored before the simulation begins. A low-speed reversing manoeuvre on TruckMaker is initiated. The desired radius is input using the knob using the aid from the visualisation module and Movie NX. The feedback to the controller are the articulation angles which are fetched in real-time and the controller receives these signals. Based on this, a desired steering angle for the front axle of the tractor and lumped rear axle of the second semi-trailer is obtained.

Steering model

Since the controller designed by Adiga and Krishnakumar [2025] (discussed in further sections) is based on a single-track model, we adjust the TruckMaker steering model to Ackermann geometry, to attempt to replicate the single-track model that the controller model thinks it is working with. This model takes the controller's desired wheel angles at the tractor and trailer and translates them into two road wheel angles at the tractor and six road wheel angles at the trailer. These road wheel angles are then fed directly into the TruckMaker model's wheel angle input blocks in real time (that is, bypassing the internal steering models and the steering wheel input options in the Manoeuvre window).

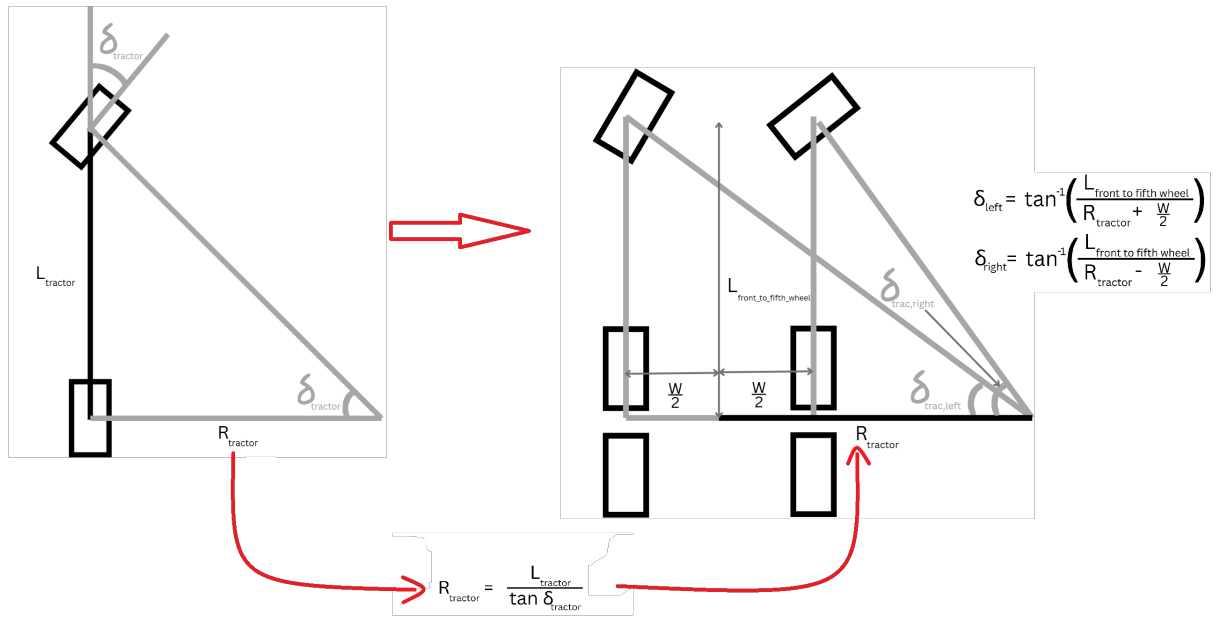


Figure 4: Ackermann-geometry wheel angle calculation for the tractor

These simple algebraic functions are implemented in Simulink as shown in Figure 5. Since we knew from Adiga and Krishnakumar [2025] that the controller could demand large steering angles, especially at the tractor, the small angle approximation $\tan\theta \approx \theta$ was not used.

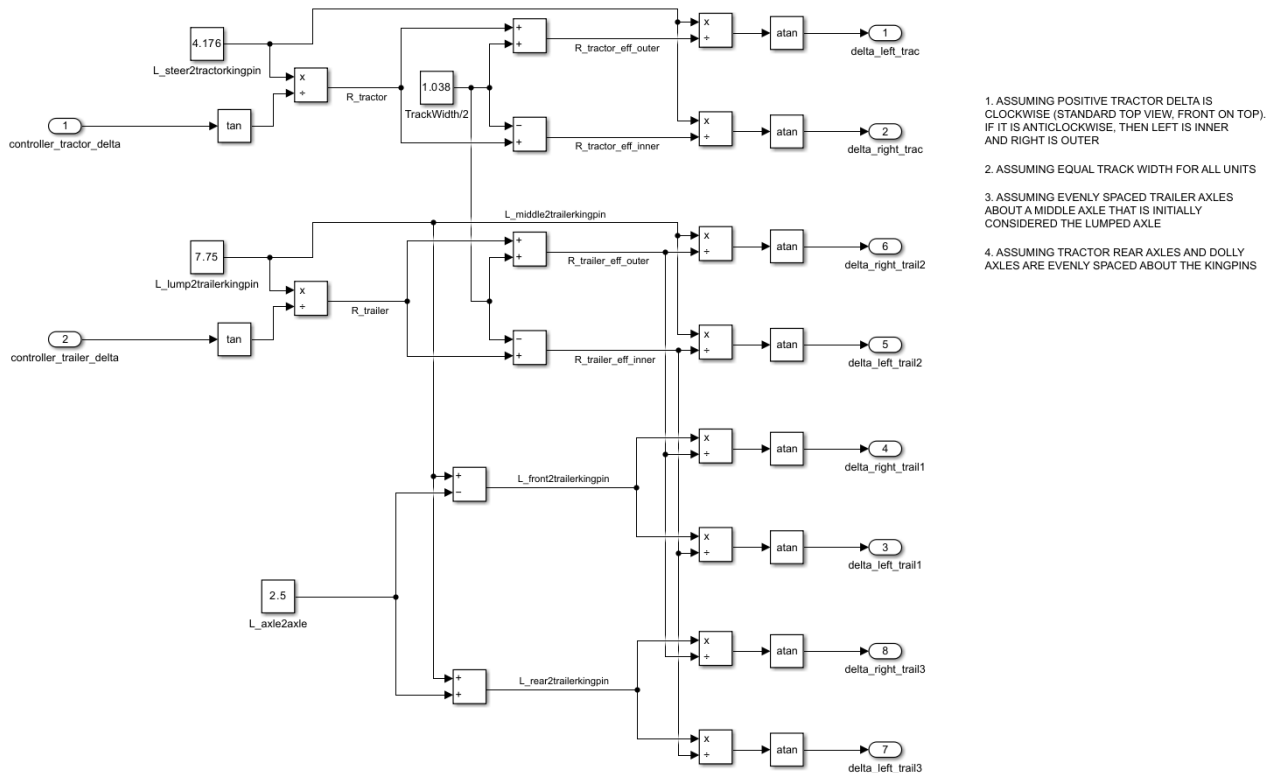


Figure 5: Ackermann-geometry calculations in Simulink

2.2 Knob module implementation

2.2.1 Virtual knob in MATLAB Simulink

Since we intend to control the radius of the last unit, for this project we treat the desired radius as the turning radius of semi-trailer 2. We first implemented a virtual driver input in Simulink using the built-in *Knob* component. In order to make the design of the virtual knob's markings more reasonable and user-friendly, the knob inputs the curvature of the desired path κ (in m^{-1}) rather than the direct radius, directly connects to the controller's input, so that $\kappa = 0$ corresponds to straight-line driving with turning radius $R \rightarrow \infty$ (since $\kappa = 1/R$). Rotating the knob clockwise and counter-clockwise represents right and left steering directions, respectively, with larger magnitudes indicating tighter turns (smaller R value). This prototype allowed us to validate the human-in-the-loop interface and the controller's expected behaviour (zero at center, left/right symmetry, and smooth variation) entirely in simulation before introducing hardware.

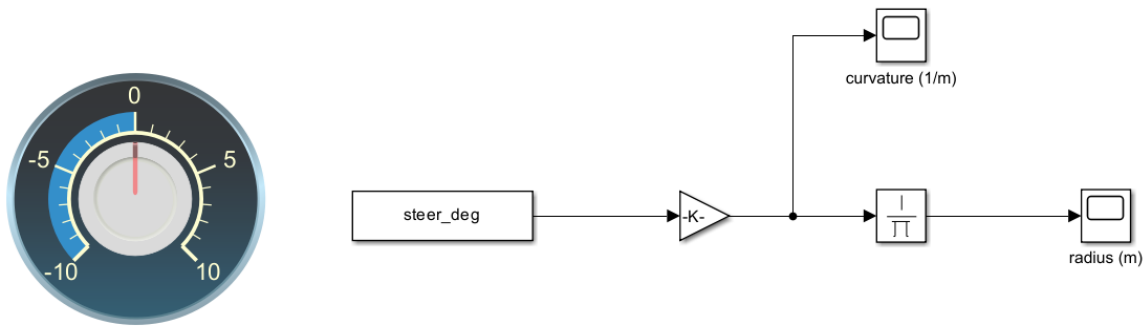


Figure 6: Virtual knob implementation

On the scale, negative values indicate a steer to the left, while positive values indicate a steer to the right. Even though the Simulink component allows us to input infinitely, we set the input to ten levels on each side, which is more in line with the hardware limitations of our next step (physical knob). The scope components here are only used to check whether the virtual knob has been set up successfully; in later works, the output will be used to connect to the input of the controller.

2.2.2 Physical knob - v1

After validating the concept in a purely simulated environment, we developed a physical knob module to enable hardware-in-the-loop (HIL) testing and to evaluate the human-machine interaction under more realistic conditions. The module is built around an STM32F103C8 microcontroller, an incremental rotary encoder, and a small OLED display, as shown in Figure 7. The objective of the hardware implementation is to reproduce the exact behaviour of the virtual knob described in Section Figure 8, while introducing meaningful physical constraints such as real-time user input and communication delays that more closely resemble a real vehicle interface.

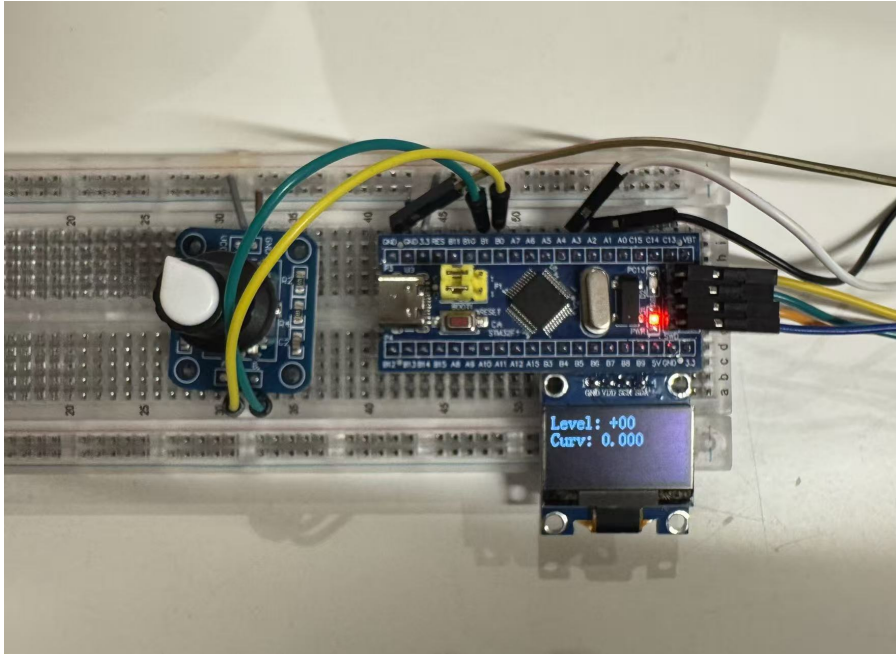


Figure 7: Physical Knob

rear_steer_knob_demo

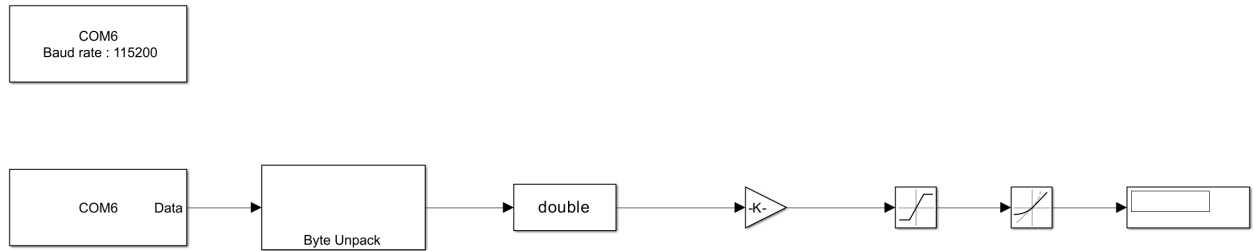


Figure 8: Simulink Interface for the Physical Knob

Quantised steering levels through encoder detents The incremental encoder provides a quadrature signal that is processed by a custom `GetCount()` routine to accumulate integer steps. Each detent corresponds to exactly one discrete steering “level”. This is a limitation imposed by the fact that this encoder with 19 detents was the only one readily available initially. The internal variable

$$\text{count} \in [-9, +9]$$

is used to represent the driver’s steering intention. Here, $\text{count} = 0$ is the centred position, representing straight-line driving ($\kappa = 0$ and thus $R = \infty$), whereas increasing positive or negative values represent right or left turns, respectively. By keeping the physical range symmetric and limited, the hardware knob enforces a natural, intuitive control metaphor while preventing unrealistic curvature inputs.

Integer-based curvature computation On the microcontroller, floating-point operations are intentionally avoided due to their relatively high computational cost on the Cortex-M3 architecture and to prevent rounding instabilities around $\kappa = 0$. Instead, curvature is represented internally as an integer milli-curvature:

$$\kappa_{\text{milli}} = 1000\kappa, \quad \kappa = \frac{\text{count}}{72}.$$

The firmware uses a custom rounding function that correctly handles both positive and negative numbers, ensuring that the curvature crosses zero smoothly. This robust handling of the zero-crossing is crucial, because

the controller input in Simulink must accept signed curvature values. Without careful rounding, fluctuations between -1 , 0 , and $+1$ milli-curvature could cause undesirable discontinuities or even destabilise the simulation. The final output is an integer in the range

$$\kappa_{\text{milli}} \in [-125, +125],$$

which offers adequate resolution for the controller while remaining noise-free.

Real-time OLED feedback To provide visual confirmation to the user, the knob includes a 128×64 OLED display driven through interface. The display is updated at approximately 10 Hz in the main loop and shows:

- the current encoder level (integer count in $[-9, +9]$),
- the corresponding curvature value in human-readable decimal format.

A formatting function converts the integer milli-curvature to a `d.ddd` string representation for display. This allows the user to monitor the effect of each detent movement and verify that the curvature is symmetric across the left and right directions. Since the OLED refresh rate is independent from the transmission rate, the display handling cannot interfere with the communication timing.

Serial communication and deterministic timing To integrate the hardware knob into the Simulink environment, curvature values must be sent to the host PC at a controlled and constant rate. To guarantee this timing, the firmware uses a hybrid approach combining the HAL framework with low-level register configuration.

USART2 is initialised manually in register-level configuration mode and operates at 115 200 baud (8N1). The curvature value is transmitted as a signed 16-bit little-endian integer, ensuring efficient data transfer and compatibility with Simulink’s `Serial Receive` block.

Deterministic timing is enforced by configuring TIM3 as a periodic timer:

$$f_{\text{TIM3}} = 100 \text{ Hz} \quad (\text{period} = 10 \text{ ms}).$$

In each timer interrupt, the latest curvature value is loaded and sent as two bytes. This ensures that:

1. the PC receives curvature samples at precisely 100 Hz,
2. communication is not affected by OLED refresh or main loop execution time,
3. the output remains temporally aligned with the controller’s sampling frequency.

This design effectively reproduces the behaviour of a real vehicle sensor delivering periodic steering measurement updates.

Main-loop responsibilities and separation of concerns The main loop of the firmware operates independently from the communication subsystem. Its responsibilities are limited to:

1. reading and clamping the encoder count;
2. computing the integer curvature value;
3. updating the OLED display;
4. making the curvature value available to the interrupt handler.

By separating computation, display handling, and communication across different execution contexts, the firmware avoids interference between subsystems and ensures that HIL experiments remain stable even under heavy user interaction.

Simulink integration and controller transparency On the Simulink side (Figure 7), the serial input is converted from the signed 16-bit integer back into a real-valued curvature using the inverse scaling. The physical knob therefore delivers:

$$\kappa(t) = \frac{\kappa_{\text{milli}}(t)}{1000}.$$

Because the format and meaning of this signal exactly match those of the virtual knob, the existing steering controller can be used without modification. Thus, the transition from simulation-only testing to hardware-in-the-loop testing is seamless.

Robustness considerations Several safety and robustness measures were included to ensure stable operation:

- a zero-crossing-safe curvature computation to prevent simulation crashes,
- range clamping to avoid invalid steering commands,
- an interrupt-driven communication scheme to guarantee consistent sampling,
- a fallback error handler with LED indication in case of hardware failure.

These measures collectively provide a stable physical interface for testing user-controller interactions and verifying the controller behaviour under realistic input conditions.

Overall, the physical knob implementation faithfully replicates the behaviour of the virtual knob while introducing real-world interaction properties. This enables controlled HIL experiments, smooth controller integration, and reliable curvature input for the path-following system.

2.2.3 Knob model - v2

We encountered two key drawbacks in the first prototype of the driver knob :

- The steering is discrete and limited to 19 distinct values, which means turning the knob is not a continuous increment but a step input type signal sent to the controller.
- The use of 'curvature' as an index to define the turning of the vehicle seemed un-intuitive, even more so for potential new users.

The new version of the knob was made using an Arduino-nano, a smooth turning potentiometer which was newly obtained, and an OLED display to provide feedback to the user.

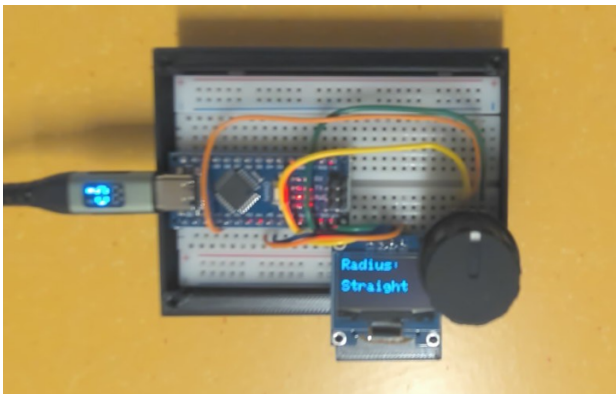


Figure 9: Knob v2

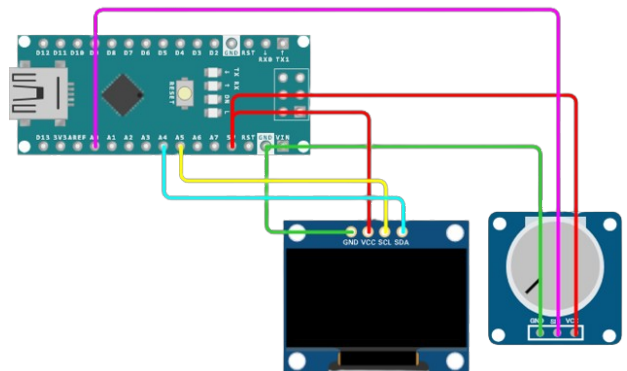


Figure 10: Knob schematic

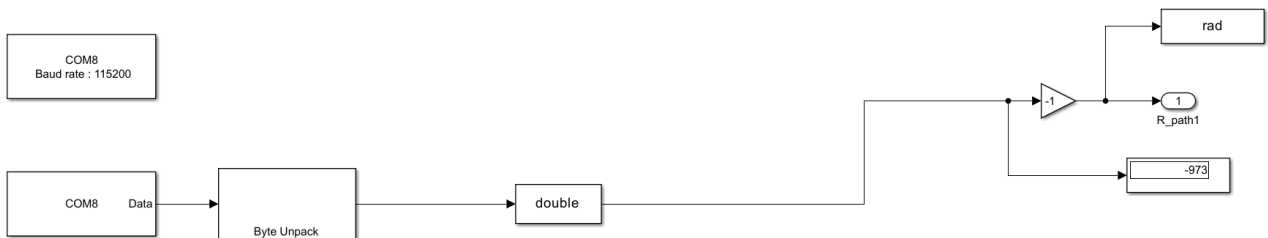


Figure 11: Knob v2 Simulink block

Float-based turning radius computation The analog pin readout by the Arduino is between 0 and 1024, so the convention was set that the values will be normalised at the midpoint by subtracting 512 from the potentiometer pin readout (now -512 to 0 is considered a left turn and 0 to +512 is considered a right turn). This symmetric normalisation simplifies downstream processing and allows the mapping function to treat left and right turns identically.

Signal smoothing Since floating point values are inherently unstable even at a fixed knob position due to rounding errors, a moving average filter was added to take the average of 30 samples. This ensures fluctuations caused by Analog to Digital quantisation and hand jitters do not make the value jump unpredictably - producing a stable steering request signal.

A window size of 30 samples was empirically chosen as the best compromise between responsiveness and stability.

Adaptive mapping of input-output The limits of the turning radius were selected as follows :

- The smallest turning radius was set to 5m in both directions.
- The straight position occurring at the midpoint of the knob's rotation corresponds to 1000m as it was a large enough value to be considered as 'infinite turning radius' with respect to the A-double's dimensions.
- In the crossover point at 0 although the value jumps from -1000 to +1000 in effect this does not affect the motion of the truck as both values are large enough to be treated as infinite radius (straight line).

We soon realised that distributing these values linearly made precise control impossible in the small to medium radius range (5m-100m) while unnecessarily using a large range of knob travel for large radius values which do not require fine control.

So the mapping between analog pin readout and turning radius was made a piecewise linear function with 2 different slopes (which are tunable based on user's sense of control).

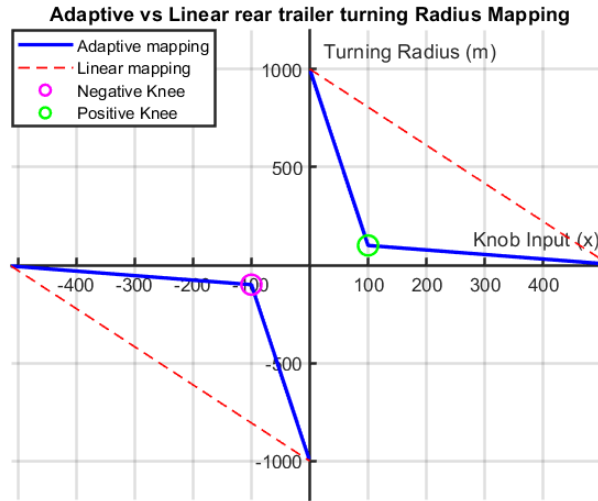


Figure 12: Knob input to turning radius mapping

As depicted in *Fig.14*, after splitting the input-output mapping line as 2 lines of different slopes, we get much more sensitive and precise control between 5m and 100m on either side but the tradeoff is very high sensitivity in the large radius range (100m to 1000m) which is not going to negatively affect the driving experience.

Overall, this adaptive mapping significantly improves low-radius manoeuvrability while maintaining intuitive control across the full steering range.

2.3 Controller and mathematical modelling

In the development of the vehicle dynamics model and control strategy for this project, we have referred to and built upon, the established framework presented by Adiga & Krishnakumar (2025).

2.3.1 Single-Track Kinematic Model

The Figure 13 represents a single-track kinematic model of an A-double combination vehicle.

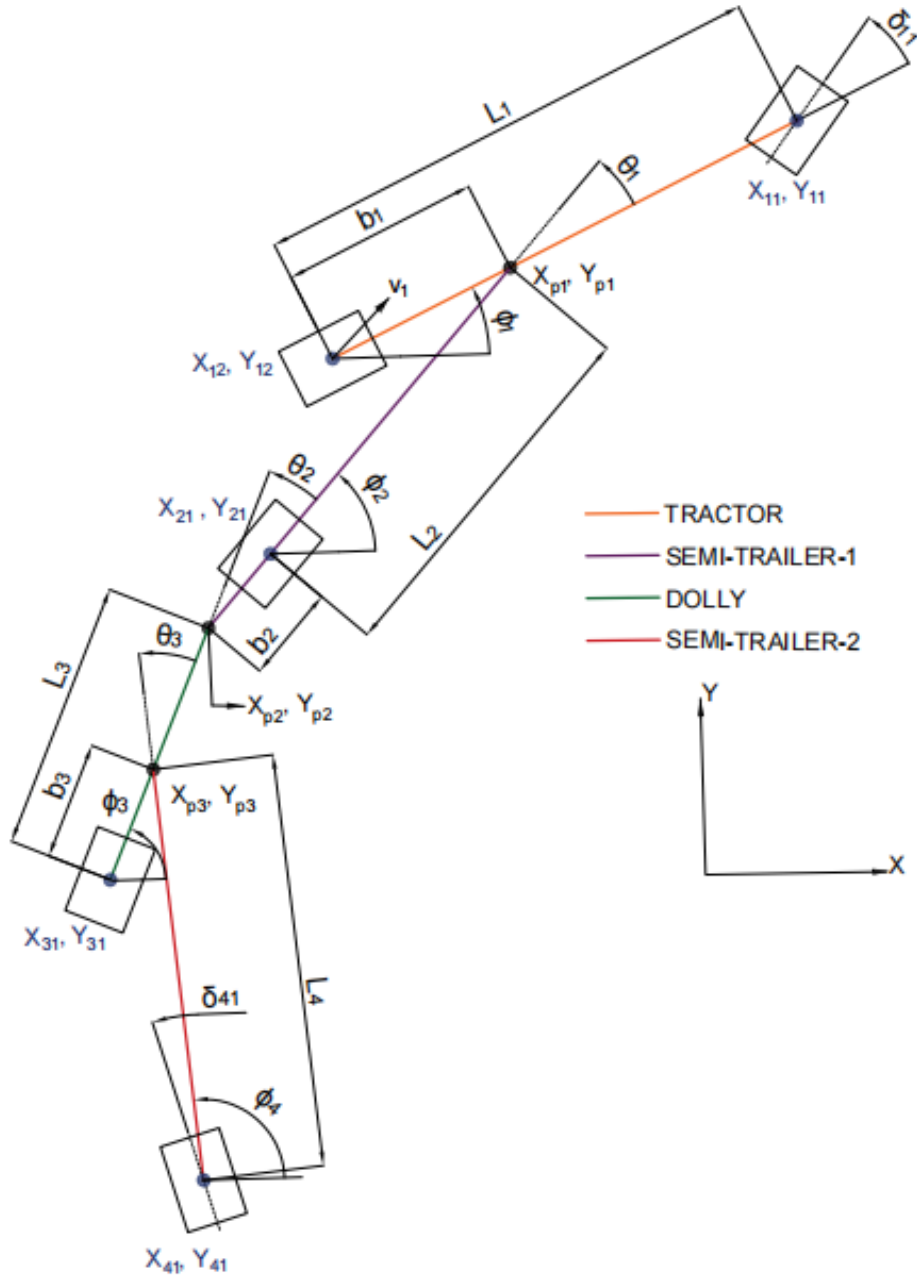


Figure 13: Single-Track Kinematic Model

The core of our vehicle modelling approach is based on the single-track (or bicycle) kinematic model adapted for articulated vehicles, as detailed in the referenced work. This model simplifies the multi-body system by lumping the axles of each unit and relies on geometric relationships under the assumption of no lateral slip, which is valid for low-speed manoeuvres. The model defines the state of the vehicle through the articulation angles between consecutive units. The key kinematic relationship we adopted is the definition of the articulation angle θ_i and its rate $\dot{\theta}_i$:

$$\theta_i = \phi_i - \phi_{i+1} \quad (1)$$

$$\dot{\theta}_i = \dot{\phi}_i - \dot{\phi}_{i+1} \quad (2)$$

where ϕ_i is the yaw angle of the i -th unit. The non-holonomic constraints at each axle, which enforce the no-slip condition, were also implemented in a similar form. For a steered axle, this constraint is given by:

$$\dot{X}_{ij} \sin(\phi_i + \delta_{ij}) - \dot{Y}_{ij} \cos(\phi_i + \delta_{ij}) = 0 \quad (3)$$

where δ_{ij} is the steering angle of the axle. This model effectively captures the essential low-speed manoeuvring behaviour of the vehicle with a tractable number of states.

2.3.2 LQR-Based Feedback Controller

For the controller design, we implemented a state-feedback control architecture analogous to the one utilised by Adiga & Krishnakumar (2025). The nonlinear kinematic model is first linearised around a straight-line equilibrium point ($\mathbf{x}_0 = [0, 0, 0]^T$, $\mathbf{u}_0 = [0, 0]^T$) using Jacobian linearisation to obtain a linear state-space representation:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (4)$$

where the state vector $\mathbf{x} = [\theta_1, \theta_2, \theta_3]^T$ represents the articulation angles, and the input vector $\mathbf{u} = [\delta_{11}, \delta_{41}]^T$ comprises the steering angles of the tractor and the trailer. The system matrices \mathbf{A} and \mathbf{B} are derived from the partial derivatives of the nonlinear kinematic equations.

An optimal Linear Quadratic Regulator (LQR) was then designed to minimise the quadratic cost function:

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (5)$$

By carefully tuning the weighting matrices \mathbf{Q} (positive semi-definite) and \mathbf{R} (positive definite), we obtained a gain matrix \mathbf{K} that stabilises the system and provides the control law $\mathbf{u} = -\mathbf{K}\mathbf{x}$, ensuring stable path-following during complex reversing manoeuvres.

While we have adapted these core components to suit the specific parameters and performance requirements of our project, the fundamental modelling methodology and control structure remain nearly identical to the approach demonstrated in [6].

2.4 Visualisation

For the purpose of relaying information about the articulation and positioning of the different units, we wanted to make a simple and near real-time display.

A function block was implemented in Simulink that takes in the articulation angles and steering angles from the controller and plots the articulated configuration of the truck. This visualisation works with virtual TruckMaker model, as demonstrated and a real vehicle, provided the controller receives the required signals $[\delta_1, \theta_1, \theta_2, \theta_3, \phi'_1, t, v_x, v_y, R_{desired}]$.

A rate transition block was added in Simulink to reduce the computational load by down-sampling the TruckMaker signals at a rate of 2Hz (0.5s). This of course is due to hardware limitation where TruckMaker and Simulink share resources. If a real truck fitted with sensors is used as the plant model then the visualisation script can sample at a higher rate if required.

The plot shows a simple bird's eye view while keeping the tractor position fixed at the origin, giving the driver a clearer sense of direction and relative motion of the trailers and eliminating any 'drifting' effect of the truck.

Each unit is represented by a simple rectangle, the articulation angles are used to compute the pose of the articulated vehicle at each frame refresh. It also displays an arc indicating the projected path of the rear of the last trailer - it is not a time-based look-ahead instead it is proportional to the instantaneous turn radius request. This choice avoids the instability that would arise from a velocity-dependent preview: fluctuations in speed would cause the arc length to expand or contract, making the visualisation inconsistent and harder to interpret. Additionally, the visualisation runs as close to real time as possible, and introducing extra computations for a time-based prediction would compromise update rate.

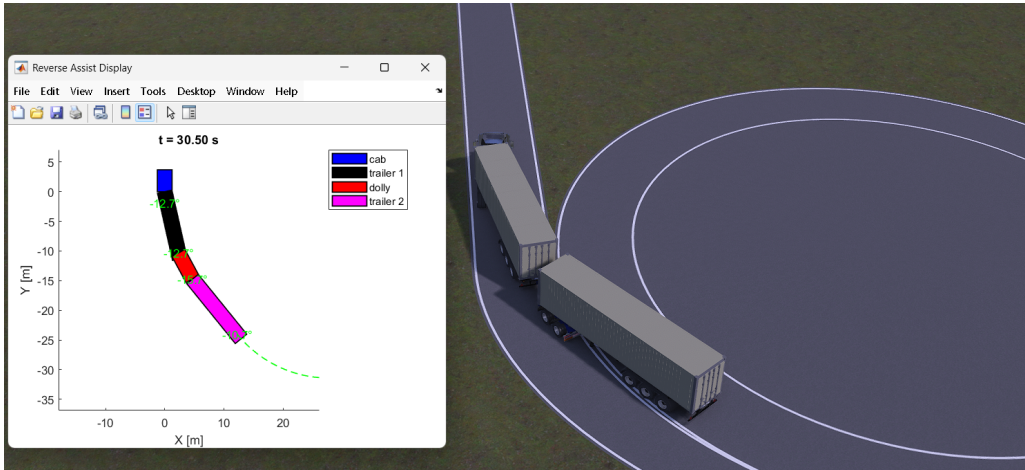


Figure 14: Simulink plot of a TruckMaker scenario

2.5 CAN module implementation

This section describes how CAN communication between the Simulink model and the real vehicle is implemented, including the CAN interface structure, the use of the DBC file, the Simulink model configuration, and the results observed during the initial vehicle tests. The objective is to generate steering request signals in Simulink and send them to the truck, while receiving vehicle speed and other feedback signals over the same CAN network.

2.5.1 CAN communication framework and DBC file

The REVERE laboratory provides a vehicle platform together with a CAN database (DBC) file that specifies the structure of CAN messages and how physical signals are encoded. In this project, the DBC file is loaded via the Vehicle Network Toolbox so that the `CAN Pack` and `CAN Unpack` blocks can automatically perform bit-level packing/unpacking and scaling between raw data and physical units.

In the current Simulink model, two categories of signals are of primary interest:

- **Vehicle speed:** successfully received and decoded in Simulink by means of `CAN Receive` and `CAN Unpack`.
- **Steering request:** generated in Simulink according to the controller output and packed into a steering request message, used to command the steering angle at the front axle.

The exact CAN identifiers, signal lengths and scaling factors are all defined in the DBC file delivered by the laboratory.

2.5.2 Simulink CAN interface model

The Simulink model contains two CAN data paths:

1. **Simulink \rightarrow Vehicle (transmit path):** the controller output (desired steering quantity, typically an equivalent front wheel angle) is passed to a `CAN Pack` block that encodes the signal according to the DBC definition. The resulting message is then sent to the vehicle via a `CAN Transmit` block.
2. **Vehicle \rightarrow Simulink (receive path):** a `CAN Receive` block listens to the vehicle CAN bus, and a `CAN Unpack` block decodes the broadcast vehicle speed message into a physical signal, which is used as feedback in the control model.

Before connecting to the real vehicle, the complete workflow was first validated using a *MathWorks Virtual CAN channel*. The virtual test confirmed that

- the DBC file can be loaded correctly;
- `CAN Pack` and `CAN Unpack` are configured consistently;
- under a fixed-step size of $T_s = 0.01$ s, the transmit and receive behaviour is periodic and deterministic;
- the packed and unpacked signals can be reproduced exactly in a `Scope` block.

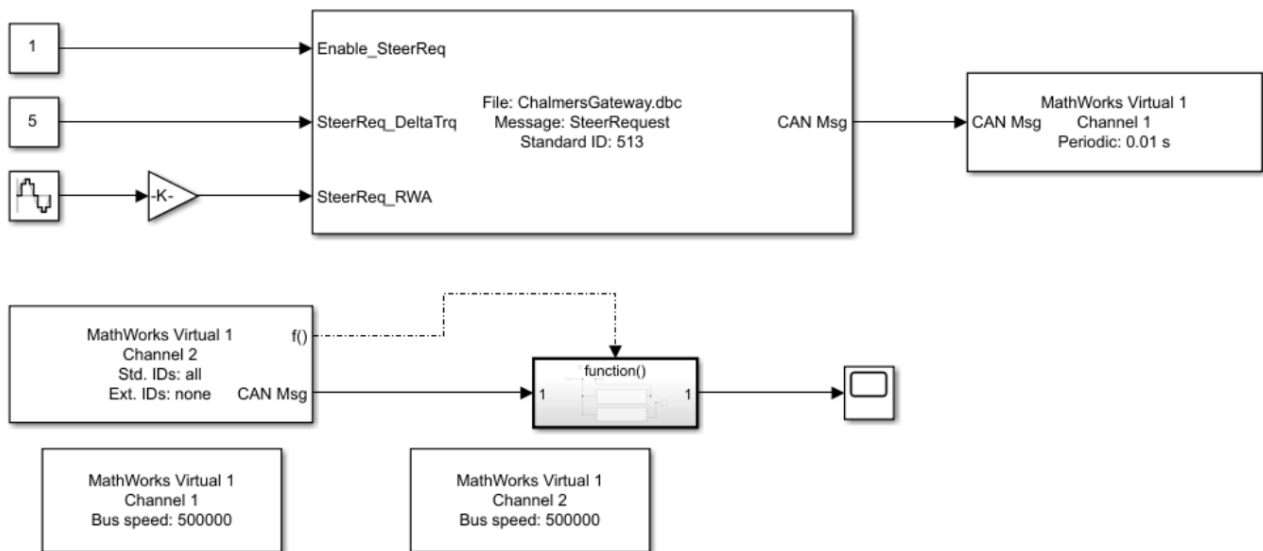


Figure 15: Simulink virtual CAN channel

2.5.3 Real-vehicle test environment

The real-vehicle experiments were conducted on the electric truck platform available in the REVERE laboratory, using the laboratory's existing USB-CAN hardware.

During the tests, the model was able to receive the vehicle's speed signals directly, which indicates that the CAN receive path is functioning correctly and that the hardware connection and gateway configuration on the laboratory side are valid for the channels used.

3 Results

3.1 Desktop Simulator

This desktop simulator was tested under multiple test cases which are listed below. Here we evaluate the steady-state performance of the simulator such as radius, articulation angles, yaw, etc.

3.1.1 Straight line reverse

Manoeuvre

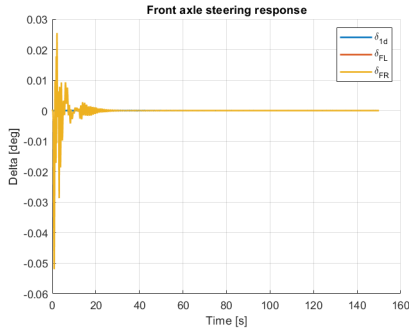
This is the simplest test case where the vehicle reverses with a constant speed in a straight line. Without the controller, we would expect a jack-knifing situation after a few seconds if no steering input is provided.



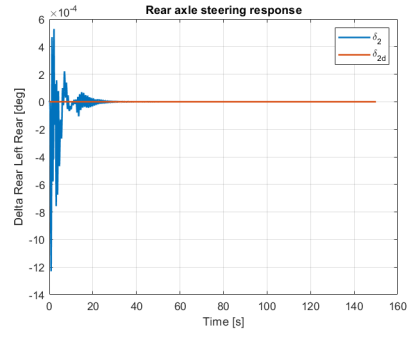
Figure 16: Straight line reverse

Performance

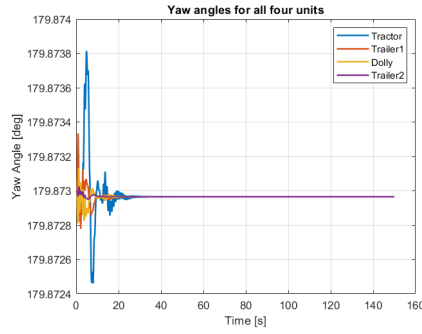
Here we see the vehicle manages to reverse without problems during the manoeuvre. From figures 17a and 17b we see some minor deviations on the front and rear steering to control the units from developing yaw movements. Figure 17c indicates none of the units experienced any yaw angle deviations. Figures 18a, 18b, and 18c also agree with this as we see fractional deviations on the articulation angles.



(a) Front Left and Front Right steering response

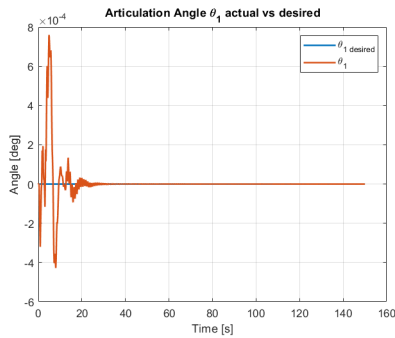


(b) Rear axle steering response

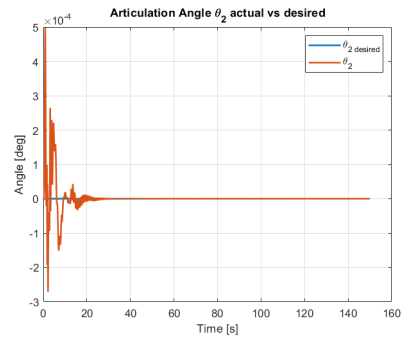


(c) Yaw angles of all units

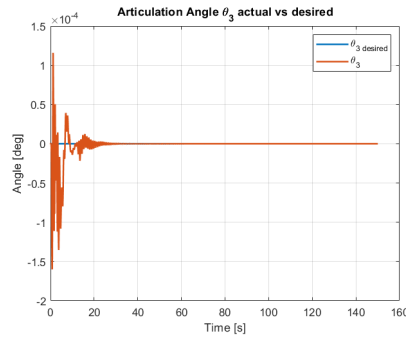
Figure 17: Performance metrics



(a) Articulation angle θ_1 response



(b) Articulation angle θ_2 response



(c) Articulation angle θ_3 response

Figure 18: Articulation angles

3.1.2 Constant high radius

Manoeuvre

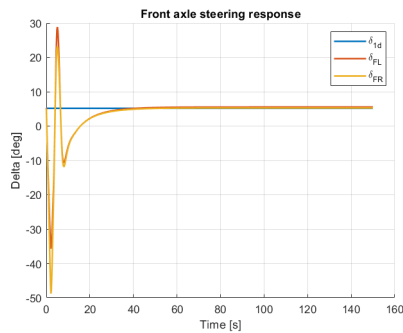
Here the vehicle reverses for a constant radius input of 50 metres.



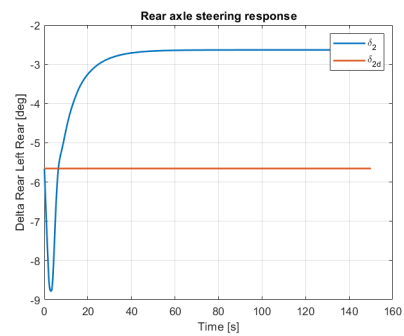
Figure 19: Constant high radius

Performance

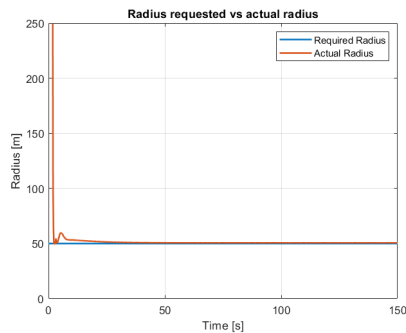
Here we see the vehicle manages to reverse without problems during the manoeuvre. From figures 20a and 20b we see the oscillatory response on the front and rear steering to control the articulation angles on the units thereby controlling the radius traversed. Figure 20c indicates the desired radius was closely matched within $\pm 1m$. Figures in 21 show the articulation angle response.



(a) Front Left and Front Right steering response

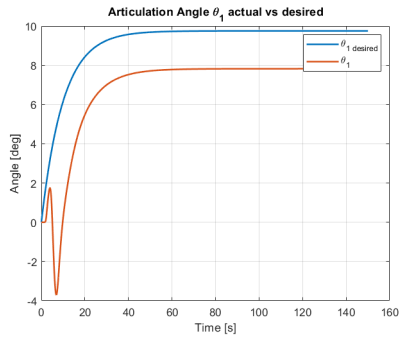


(b) Rear axle steering response

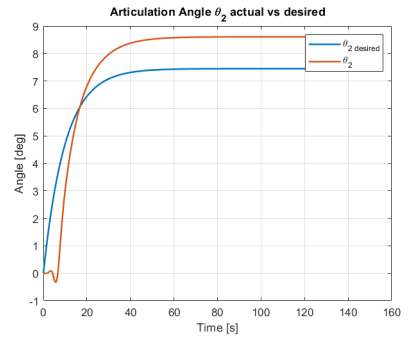


(c) Radius traversed

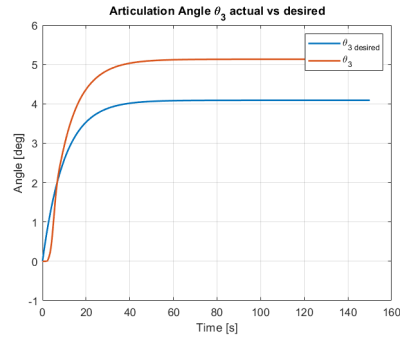
Figure 20: Performance metrics



(a) Articulation angle θ_1 response



(b) Articulation angle θ_2 response



(c) Articulation angle θ_3 response

Figure 21: Articulation angles

3.1.3 Constant low radius

Manoeuvre

Here the vehicle reverses for a constant radius input of 15 metres.

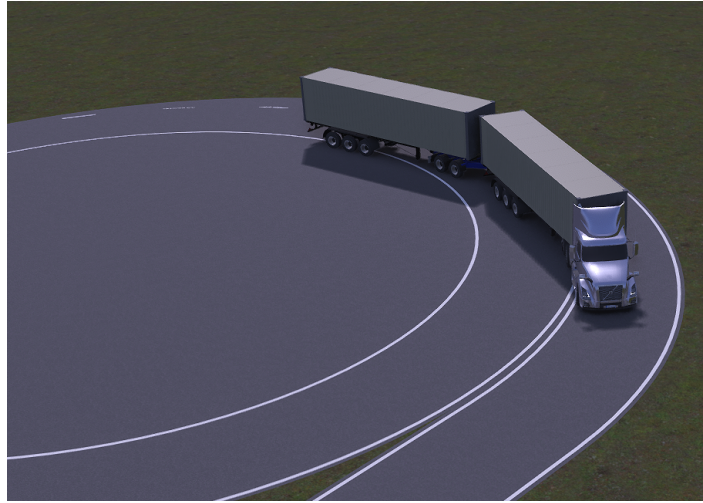
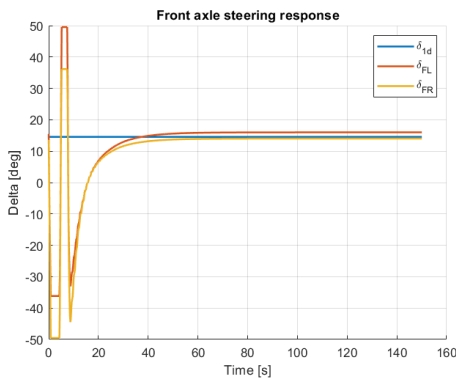


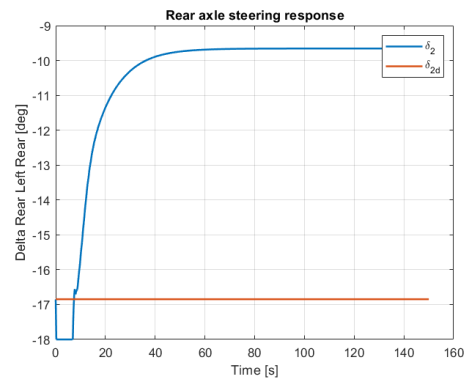
Figure 22: Constant small radius

Performance

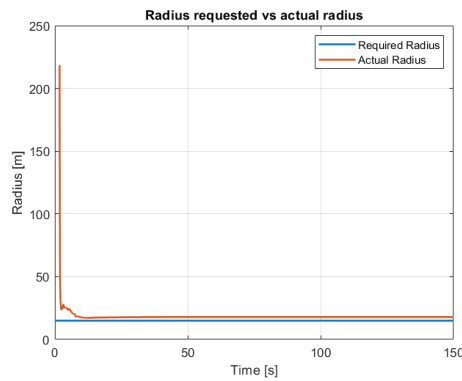
Here we see the vehicle manages to reverse without problems during the manoeuvre. From figures 23a and 23b we see the oscillatory response on the front and rear steering to control the articulation angles on the units thereby controlling the radius traversed. Figure 23c indicates the desired radius was closely matched within $\pm 2m$. Figures in 24 show the articulation angle response.



(a) Front Left and Front Right steering response

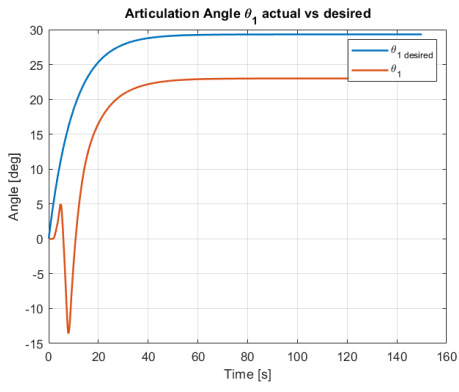


(b) Rear axle steering response

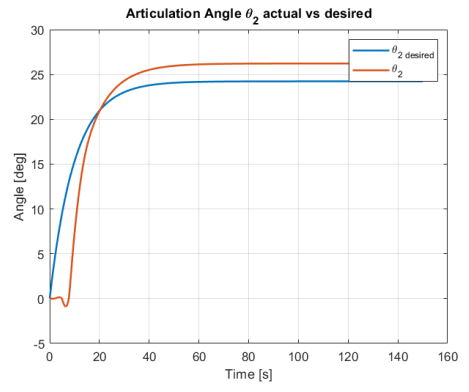


(c) Radius traversed

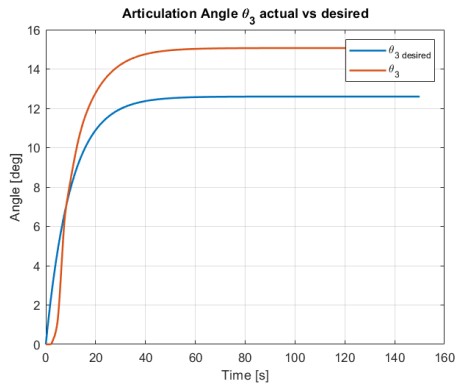
Figure 23: Performance metrics



(a) Articulation angle θ_1 response



(b) Articulation angle θ_2 response



(c) Articulation angle θ_3 response

Figure 24: Articulation angles

3.1.4 90° reverse

Manoeuvre

In this test case the vehicle is to reverse on a wide lane of an arterial road onto another road at a 90° intersection. An ideal manoeuvre would be reversing in the opposite direction of the intersecting road in a shallow radius and once the truck is close to the intersection turn sharply into the desired direction and straighten the combination.

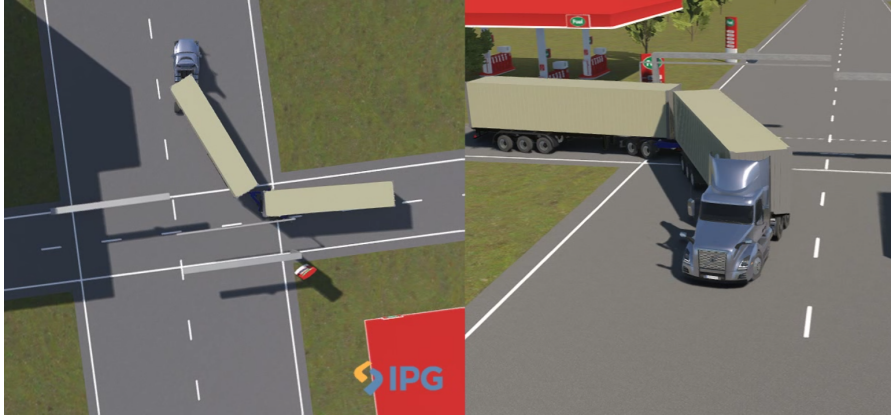


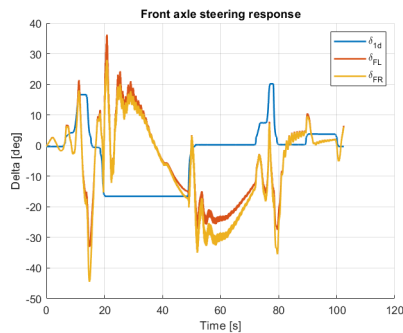
Figure 25: 90 degree reverse scenario

Performance

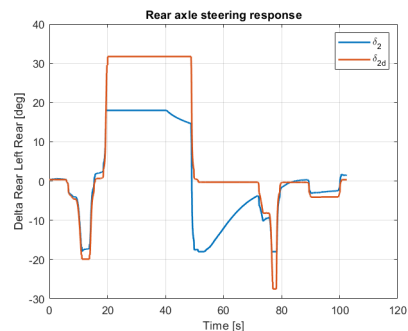
The truck is able to perform this as expected. If a narrow time window to enter the intersecting road is missed there is no counter manoeuvre in the reverse direction that can rectify it.

From the radius plot (Fig.28(c)) it is clear that the controller behaves asymmetrically - when moving from a larger radius to a smaller radius in a turn the controller tracks the required value closely but when straightening the combination (going from small to very large turn radius) the slope of the values is not as high.

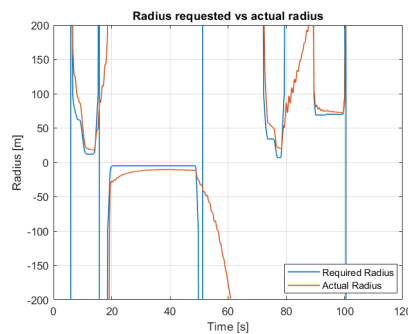
This could also explain why the articulation angle values track close to desired values except when such a straightening manoeuvre is performed between time 45s and 60s.



(a) Front Left and Front Right steering response

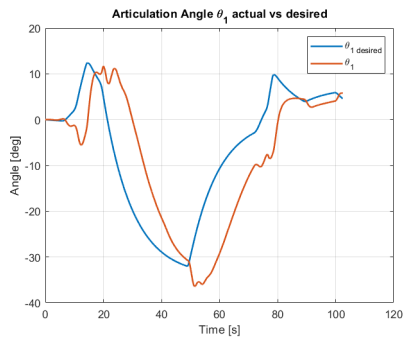


(b) Rear axle steering response

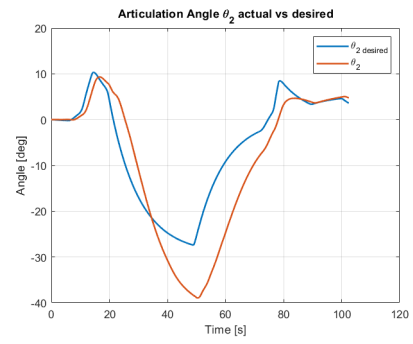


(c) Radius traversed

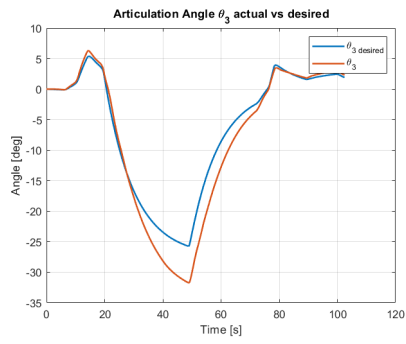
Figure 26: Performance metrics



(a) Articulation angle θ_1 response



(b) Articulation angle θ_2 response



(c) Articulation angle θ_3 response

Figure 27: Articulation angles

3.1.5 Misaligned units

Manoeuvre

An A-double in real life is rarely positioned such that all units are perfectly aligned along a straight line. Here, random steering inputs are given to the front and rear axle to deliberately misalign the vehicle units. Following this, the controller is engaged and the vehicle is reversed while being given a constant radius target of 25 metres.

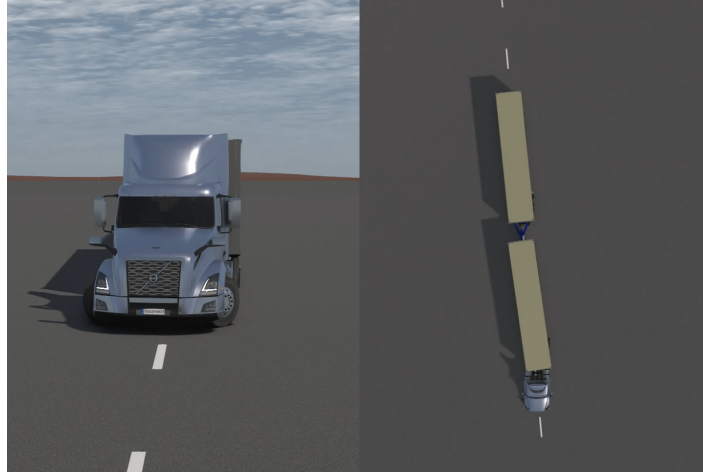
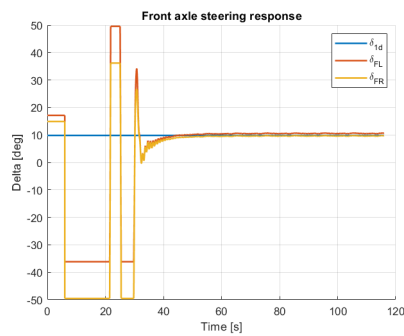


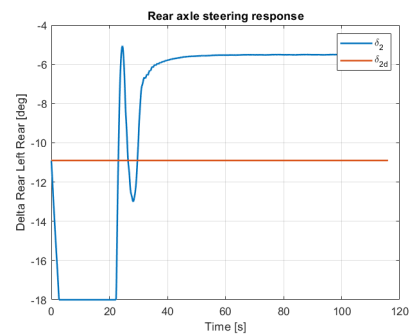
Figure 28: Misaligned units

Performance

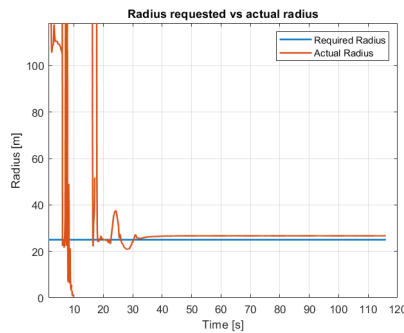
The vehicle may not always start reversing while all units are aligned without any yaw. This manoeuvre shows how the vehicle would perform in such situations. The vehicle does manage to pass this. However, due to the complexity of the manoeuvre, this test case is not completely robust, and depends on the extent of the misalignment.



(a) Front Left and Front Right steering response

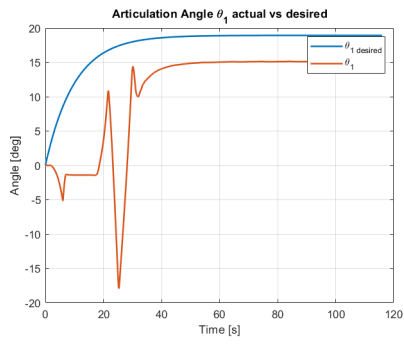


(b) Rear axle steering response

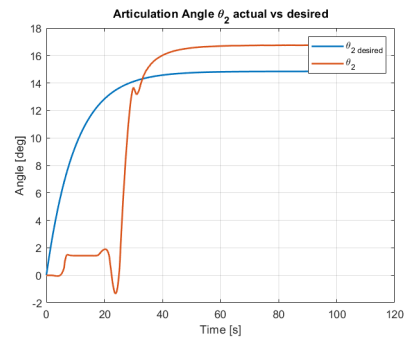


(c) Radius traversed

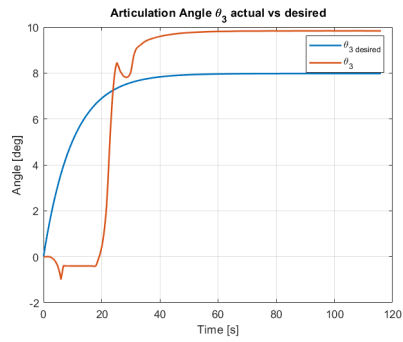
Figure 29: Performance metrics



(a) Articulation angle θ_1 response



(b) Articulation angle θ_2 response



(c) Articulation angle θ_3 response

Figure 30: Articulation angles

3.1.6 Reverse parking manoeuvre

Manoeuvre

In this test case the vehicle is to reverse into a parking slot on a cargo terminal setup while avoiding other vehicles and people. An ideal manoeuvre would be reversing close to the parking lane and initiating a sharp turn of 5-10m radius to the left.

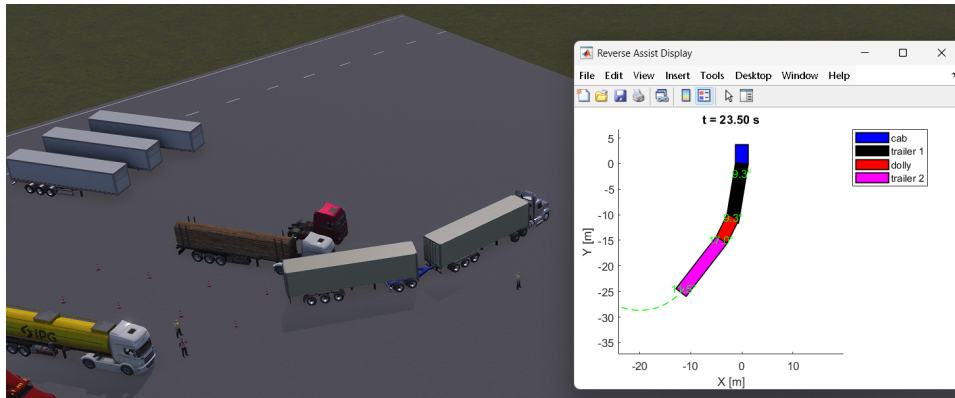


Figure 31: reverse parking scenario

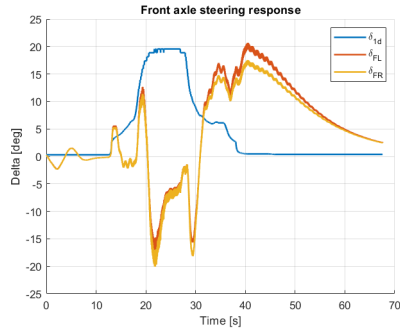
Performance

The truck is able to perform this as expected. Due to the setup of the scenario there seems to be ample time window to correct course.

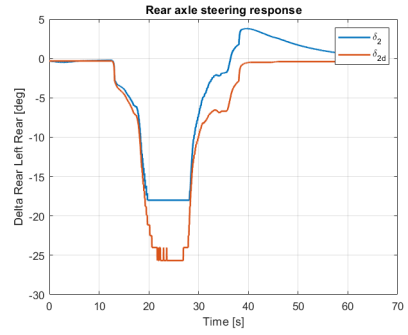
As discussed in the 90 degree reverse scenario (see 3.1.4) when going out of a turn to a straight pose the control action is slow as shown in the radius plot (Fig.34(c)).

As the requested radius gets larger, small changes in steering angles do not produce significant changes in vehicle's turning arc (unlike in cases for small turning radii) which results in a fast step-down response but a poor step-up response.

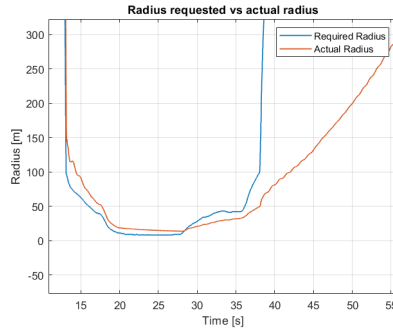
The controller is setup in a way to minimise sudden variations in the articulation angles so as to prioritise stability.



(a) Front Left and Front Right steering response

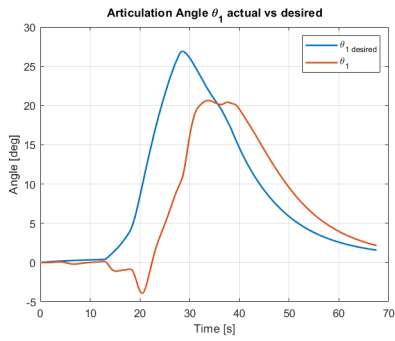


(b) Rear axle steering response

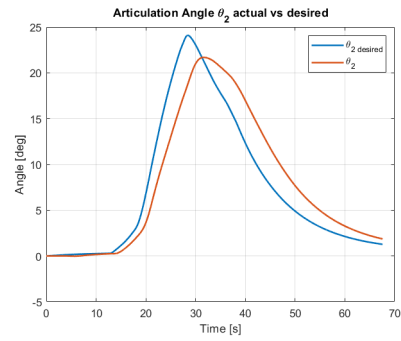


(c) Radius traversed

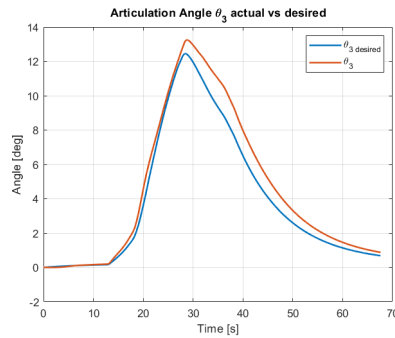
Figure 32: Performance metrics



(a) Articulation angle θ_1 response



(b) Articulation angle θ_2 response



(c) Articulation angle θ_3 response

Figure 33: Articulation angles

3.2 Real vehicle test and CAN communication

3.2.1 Signal reception (successful)

Using the `CAN Receive` and `CAN Unpack` blocks, the Steering wheel angle signal was successfully decoded and displayed in a `Scope`. The measured angle followed the actual angle in real time.

3.2.2 Steering request transmission (pending)

Under the same test conditions, steering request messages were generated in Simulink according to the DBC definition and transmitted via the `CAN Transmit` block. However, no observable steering response was produced by the vehicle.

Likely cause (gateway configuration limitation). According to information from the REVERE laboratory, the current test vehicle is a new electric truck that recently replaced an older diesel platform. The gateway module installed on the electric truck currently offers only three interfaces, and some of the CAN channels may not be connected to the network where the steering ECU resides. As a result, although the steering request messages are sent onto the bus, they may not be forwarded by the gateway to the steering controller.

The laboratory plans to swap the gateway from the original diesel truck onto the new electric platform in order to restore the full steering-control interface. After this modification, the steering request transmission from Simulink to the real vehicle will be re-tested.

4 Test day at Hällered Proving Ground

On 21 October 2025, we had the opportunity to visit Volvo's Hällered Proving Ground and perform some manoeuvres, both forward and reverse with REVERE's diesel semi-truck before it was retired for good.

We processed the GNSS and inertial measurement data recorded during the test and generated relevant visualisation results. The visualisation of Skidpad Team 1 test data is as below.

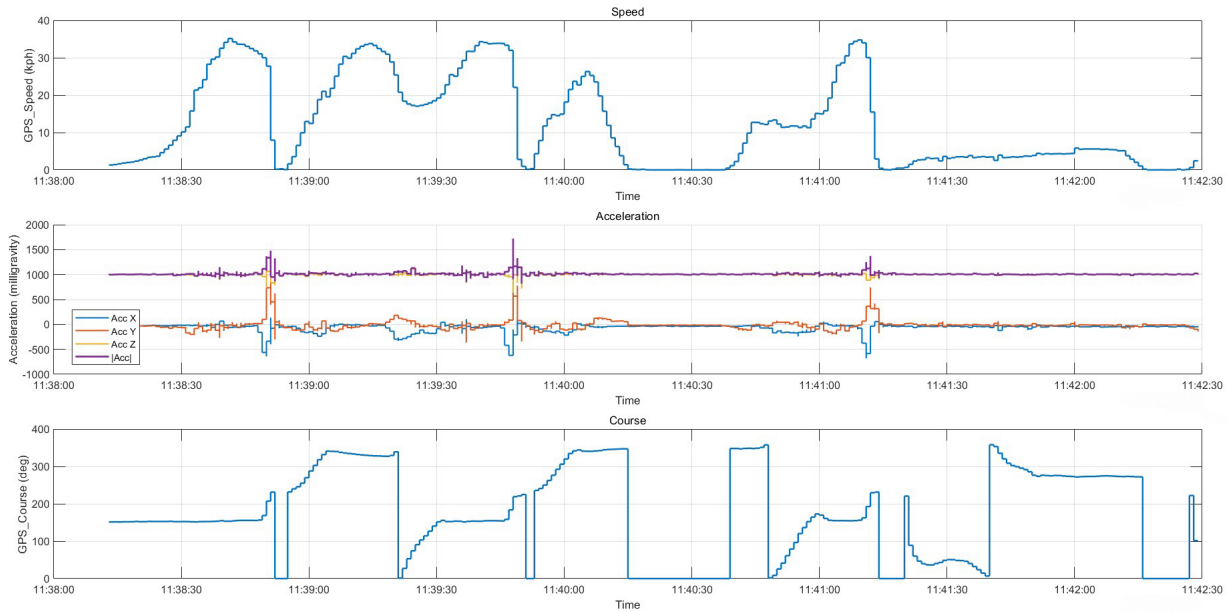


Figure 34: Speed/Acc/Course vs Real Time

The velocity curve exhibits multiple acceleration-deceleration cycles, indicating that the vehicle repeatedly entered, exited, and turned on the test track. Since the manoeuvres were largely performed at low speeds, the three-axis acceleration signals are generally stable, with limited spikes only appearing at moments when the vehicle turns or changes its radius of curvature; the heading angle curve shows periodic changes, consistent with the directional changes of the vehicle being steered and reversed in various ways on the skidpad.

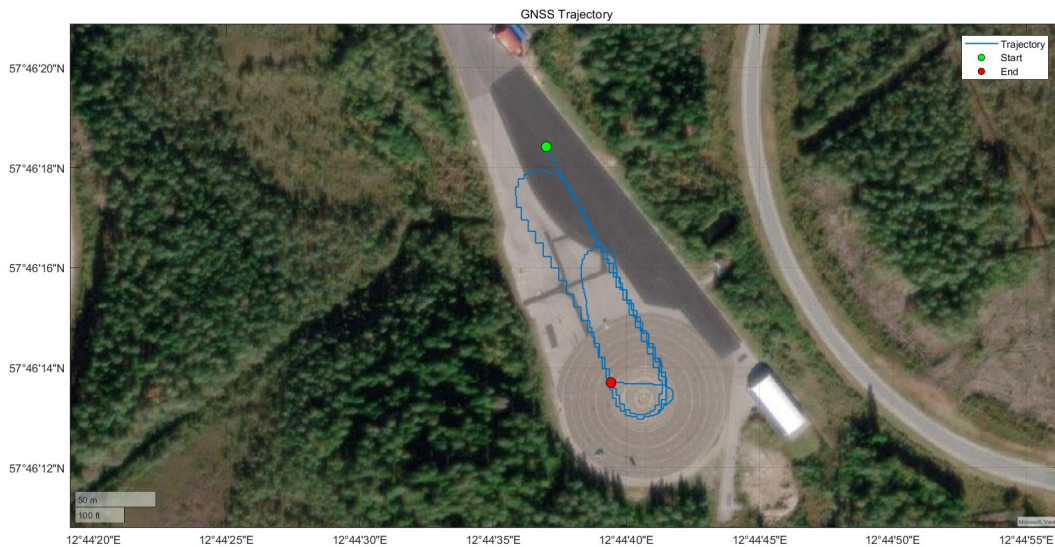


Figure 35: GNSS Trajectory

The corresponding GNSS trajectory on a satellite basemap is shown in Figure 35. By overlaying the latitude and longitude data onto the base map of the Hällered test range, it is clear that the vehicle repeatedly runs

along a typical elliptical path, with the starting and ending positions consistent with the velocity and heading changes in the time-domain data in Figure 34.

Relevance to the rest of the project

The results indicate that the test process and data analysis methods we used are reliable for plotting the trajectory and motions of a real vehicle. This method can be replicated with an A-double at a later stage, both to verify that the TruckMaker/Simulink model is close to reality and for further use by the manufacturer for parameter tuning and so on.

5 Conclusion and Future Work

5.1 Conclusion

This project investigated a driver-assist concept for low-speed reverse steering of long combination vehicles, with a specific focus on A-double configurations. A complete desktop simulation framework was developed by integrating TruckMaker vehicle models with MATLAB/Simulink-based control, visualisation, and human-machine interface modules. The core idea was to support the driver during reversing manoeuvres by combining rear-axle steering with an intuitive knob-based input for desired path curvature or turning radius.

A kinematic single-track model was adopted to capture the essential low-speed behaviour of the articulated vehicle, and an LQR-based feedback controller was implemented to stabilise articulation angles and achieve robust path tracking during reverse motion. Simulation results from multiple manoeuvre scenarios, including straight-line reversing, constant-radius reversing at both large and small radii, and misaligned initial conditions, demonstrate that the controller is able to maintain stable articulation behaviour and closely follow the desired trajectory. These results indicate that the proposed control structure is suitable for low-speed reversing tasks of long combination vehicles under idealised kinematic assumptions.

In addition, significant effort was devoted to the design and evaluation of the driver input interface. Both a virtual knob and two generations of a physical knob were developed and successfully integrated into the simulation environment. The second hardware version, based on a continuous potentiometer and adaptive mapping of input to turning radius, provided a more intuitive and precise control experience, especially in tight-radius manoeuvres. This highlights the importance of human-machine interaction design in driver-assist systems, beyond control performance alone.

Finally, the feasibility of extending the desktop simulator toward real-vehicle application was explored through CAN communication with a test vehicle at the REVERE lab. While vehicle speed signals were successfully received in real time, steering request transmission could not yet be validated due to gateway configuration limitations on the current vehicle platform. Nevertheless, the successful reception of signals confirms that the overall communication framework and Simulink-CAN interface are correctly designed and ready for further validation once the hardware configuration is updated.

Overall, the project demonstrates that a combined approach of rear-axle steering, feedback control, and intuitive driver input has strong potential to improve the ease of reversing manoeuvres for long combination vehicles. The developed simulation and HIL-ready framework provides a solid foundation for future refinement, higher-fidelity modelling, and full real-vehicle implementation.

5.2 Future work

5.2.1 Evaluate alternative control strategies like MPC

In the future, it is essential to review and implement advanced control strategies such as Model Predictive Control (MPC). MPC can optimise system states over a prediction horizon, thus enhancing the vehicle's ability to adapt to complex scenarios and improve overall control performance. Comparative studies using simulations and experiments can be conducted to assess the dynamic response, robustness, and disturbance rejection capabilities (e.g., to road changes or load variations) of different controllers. Furthermore, integrating MPC with other methods such as adaptive or robust control could be explored to further enhance performance under uncertainties.

5.2.2 Dynamic model refinement

The current dynamic model the controller is based on relies on several simplifications, especially regarding tire and chassis dynamics. It considers a fully kinematic one track model. Future work should incorporate more complex and accurate tire models (e.g., Magic Formula or Pacejka models) and detailed suspension system models to raise the fidelity of both simulation and control. This involves more precise modelling of lateral stiffness, the effects of scrubbing due to the presence of multiple axles, and the influence of vertical load on tire behaviour. Model parameter identification and calibration tailored to specific vehicles and varying operating conditions could also be explored.

5.2.3 Articulation angle estimation

Accurate estimation of articulation angles is crucial, especially for the last trailer. The easiest way to do this is of course to use angle sensors, but future research can focus on developing real-time and high-precision angle estimation algorithms using, say, IMU data from each unit.

5.2.4 Improvements to the visualisation display

If during real vehicle setup and testing the truck is equipped with radar or lidar, the sensed objects or point cloud could be added in the display window to provide context of the surroundings to the driver.

Making the script leaner and more refined would reduce the computational load, potentially making it possible to sample the inputs at a higher rate and running the TruckMaker simulations in real time instead of 0.7x-0.9x of real time like we managed in our trials.

Adding any additional features to be shown on the display after feedback from actual drivers, such as wheel guiding lines, would also be beneficial.

5.2.5 Real vehicle testing and deployment

After model refinement and simulation validations, the algorithms and systems need to be transferred to real vehicles for practical testing. In the first step, this can focus on continuing with the models prepared for the desktop simulator, and merely replacing the simulation step in TruckMaker with data from the real vehicle, with communication to and from Simulink via the CAN modules prepared above. The architecture, then, will be as below:

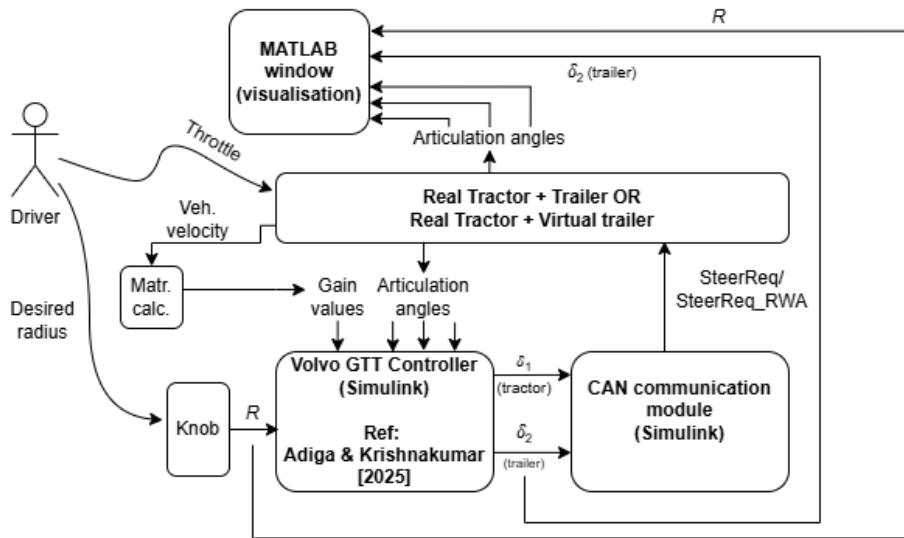


Figure 36: Model architecture with a real vehicle in the loop

When deployed to a test vehicle like those available at REVERE, the approaches used during the Hällered test can be used to plot the trajectory of the tractor, from which differences between simulation and physical system performance can be identified and addressed. This real-world data can be used for further algorithm improvement, model calibration, and new feature development (such as adaptive behaviour or fault detection), thus paving the way for actual deployment.

When deployed in a production vehicle, the control algorithms will need to be condensed into much more concise code in the vehicle's motion control unit, and the knob will also need to be integrated with the vehicle's CAN network. It will also need ergonomic redesign to make it more intuitive for drivers. Finally, the trailer we have modelled - with three actively steerable axles - is also a hypothetical device which we have explored without comment on its feasibility; this is left to the manufacturer to explore.

References

- [1] Nathan Van De Wouw et al. “Active trailer steering for robotic tractor-trailer combinations”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 4073–4079.
- [2] Jakob Roempke and Chang Liu. *Trailer backup assist using steer-by-wire*. Master’s degree thesis at Kungliga tekniska högskolan, Stockholm. 2024.
- [3] Joseph M Raad and Donald Jacob Mattern. *Trailer backup assist system with adaptive steering angle limits*. US Patent 9,522,699. Dec. 2016.
- [4] Qiushi Wang. *Design and validation of active trailer steering systems for articulated heavy vehicles using driver-hardware-in-the-loop real-time simulation*. Master’s degree thesis at the University of Ontario Institute of Technology, Oshawa, 2015.
- [5] Zhaohui Ge, Fredrik Bruzelius, and Bengt Jacobson. “Long Combination Vehicles Reverse Strategies Based on Articulation Angle Gradient”. In: *AVEC 2024*. Lecture Notes in Mechanical Engineering. Springer, 2024, pp. 707–713. DOI: 10.1007/978-3-031-70392-8_100.
- [6] Pavan Kumar Adiga Nagaraj and Niveditha Krishnakumar. *Reverse assistance steering control for long combination vehicles*. Master’s degree thesis at Chalmers tekniska högskola, Göteborg. 2025. URL: <https://odr.chalmers.se/items/bc7944f8-cd20-4abe-babe-b688fa401742>.