World Electric Vehicle Journal

Article

All-Wheel Steering Tracking Control Method for Virtual Rail Trains with Only Interoceptive Sensors

Zhenpo Wang [1,2], Yi Zhang [1] and Zhifu Wang [1,2,*]

[1] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; wangzhenpo@bit.edu.cn (Z.W.); zhangyizdh@bit.edu.cn (Y.Z.)
[2] National Engineering Research Center of Electric Vehicles, Beijing Institute of Technology, Beijing 100081, China
* Correspondence: wangzhifu@bit.edu.cn

Abstract: A virtual rail train (VRT) is a multi-articulated vehicle as well as a novel public transportation system due to its low economic cost, environmental friendliness and high transit capacity. Equipped with all-wheel steering (AWS) and a tracking control method, the super long VRT can travel on urban roads easily. This paper proposed a tracking control approach using only interoceptive sensors with high scene adaptivity. The kinematic model was established first under reasonable assumptions when the sensor configuration was completed simultaneously. A hierarchical controller consists of a front axle controller and a rear axle controller. The former applies virtual axles theory to avoid motion interference. The latter generates a first-axle reference path with path segmentation and a data updating method to improve storage and computational efficiency. Then, a fast curvature matching rear axles control method is developed with an actuator time delay considered. Finally, the proposed approach is verified in a hardware in loop (HIL) simulation under various situations with predefined evaluation standards, which shows better tracking performance and applicability.

Keywords: virtual rail train; interoceptive sensors; tracking control; all-wheel steering

Citation: Wang, Z.; Zhang, Y.; Wang, Z. All-Wheel Steering Tracking Control Method for Virtual Rail Trains with Only Interoceptive Sensors. *World Electr. Veh. J.* **2024**, *15*, 247. https://doi.org/10.3390/wevj15060247

Academic Editor: Joeri Van Mierlo

Received: 17 May 2024
Revised: 28 May 2024
Accepted: 31 May 2024
Published: 4 June 2024

1. Introduction

Due to the rapid urbanization in China over the past two decades, traditional public transportation systems can barely meet the growing and various demands of citizens. Buses and subways are the most common low- and high-transit-capacity systems. Many debates about which to develop lasted for decades until the appearance of medium-transit-capacity systems (MTCS), which do not terminate but heat the arguments [1].

MTCSs carry a maximum of 5000~30,000 persons per hour per line, in contrast with buses (less than 5000) and subways (larger than 30,000) [2,3]. Unlike buses and subways, MTCSs do not have unified forms as well as professional names. Here, several main categories are listed, like light rail transit (LRT), monorail vehicles (straddle or suspended, SMV), low-to-medium-speed maglev trains (MLT), rubber tire trams (RTT), bus rapid transit (BRT) and VRT. Figure 1 shows the real image and Table 1 shows some key features [4–7].

LRT, SMVs and MLTs depend on expensive and complex infrastructure, even comparable to subways. They are often treated equally as simplified subways by the public. Originating from traditional tram, RTTs can only travel along a fixed guide monorail embedded in the road with extra catenary power supply. As public transportation develops and optimizes, some RTTs are terminated in cities like Shanghai, Tianjin and Zhuhai, China. BRT and VRTs can be seen as super buses that can travel any urban road with the lowest construction cost. Compared with RTTs, VRTs provide the same capacity with a lower cost. Compared with BRT, VRTs provide a higher capacity with the same cost. Outstanding cost-effectiveness has made VRTs popular for big city suburbs and middle city urban areas in recent years.

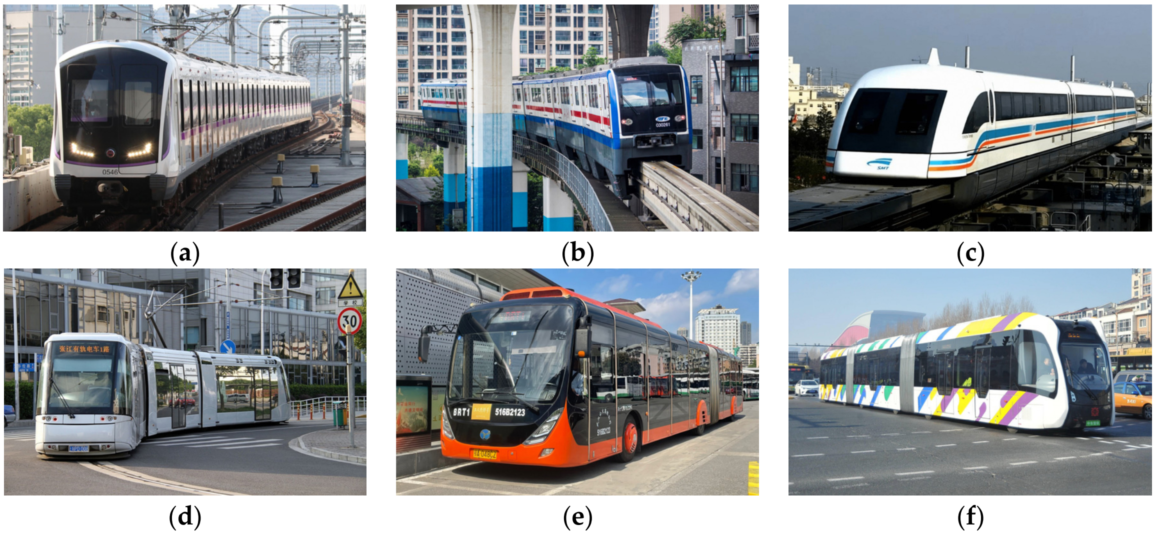

(**a**)
(**b**)
(**c**)
(**d**)
(**e**)
(**f**)

Figure 1. Various MTCSs: (**a**) LRT; (**b**) SMV (straddle); (**c**) MLT; (**d**) RRT; (**e**) BRT; (**f**) VRT.

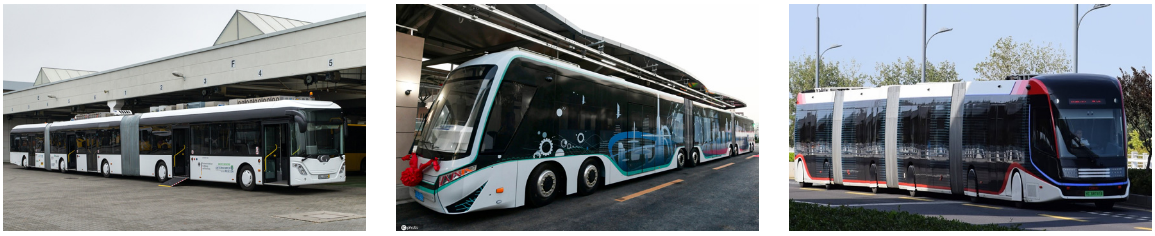
Table 1. Key features for various MTCSs.

Name	**Size (L × W, m)**	**Vehicle Marshalling**	**Unit/Line Capacity**	**Running Form (Rail, Wheel, Road)**	**Power**	**Construction Cost (million¥/km)**	**Right of Way**
Bus	12.0 × 2.5	1	80/3000	NR + RT + OG	EV/HEV	/	Mixed
Subway	120 × 2.8	4~8	1460/44,000	SR + SW + UG/ET	AC + CPS/RPS	600	Exclusive
LRT	80 × 2.6	4~6	900/25,000	SR + SW + UG/ET	AC + CPS/RPS	300	Exclusive
SMV	70 × 2.6	4~6	700/18,000	GR + RT + ET	AC + RPS	200	Exclusive
MLT	75 × 2.8	4~6	800/20,000	MR + LM + OG/ET	DC + RPS	250	Exclusive
RTT	30 × 2.5	3~4	300/12,000	GR + RT + OG	AC + CPS	100	Exclusive
BRT	18 × 2.6	2	180/7000	NR + RT + OG	EV/HEV	50	Exclusive/Mixed
VRT	32 × 2.6	3~4	300/12,000	NR + RT + OG	EV/HEV	50	Exclusive/Mixed

NR: no rail, SR: steel rail, GR: guide rail, MR: maglev rail, SW: steel wheel, RT: rubber tire, LM: linear motor, OG: on ground, UG: underground, ET: elevated track, AC: alternating current, DC: direct current, EV: electric vehicle, HEV: hybrid electric vehicle, CPS: catenary power supply, RPS: rail power supply.

Usually, a driver takes control of the first axle and the other modules will follow the path automatically as if the VRT is running on a real rail. A virtual rail has two meanings: a head module tracking predefined path and other modules tracking head module. The former is like an autonomous driving problem similar to that of regular cars and the latter is our concern. Without tracking control, VRTs cannot move anymore with off-tracking, severe tire wear, tail swing and even structure interference. This key technology is achieved by sensors, external infrastructure (not real rails) and algorithms, which are our main considered factors.

Figure 1f shows a special kind of VRT (Autonomous rail Rapid Transit (ART) produced by CRRC Zhuzhou Institute Co., Ltd., Zhuzhou, China), which is our interest for its most typical and broad applications. The structure consists of three modules, two articulated joints, six AWS axles in which head/tail axles are driven axles and two driving cabs at the head and tail without backing up. There also exist other VRTs, and we tried to define some common features including multi-articulated bodies, active steering, rubber tires and virtual rail tracking. Figure 2 shows other types of VRTs in commercial operation with the main difference described [8]. We will talk about the tracking control of various VRTs and other active steering articulated vehicles. After that, some topics are discussed, and our reach is introduced.



(**a**) (**b**) (**c**)

Figure 2. Various VRTs: (**a**) AutoTram, five axles, one driving cab, second axle fixed as a driven axle; (**b**) Digital Rail-guided Train (DRT), four double-axle bogies; (**c**) Super Rail rapid Transit (SRT), four bodies articulated like two BRTs connected tail to tail.

Wagner proposed a kinetic feedforward–feedback (FF-FB) tracking controller for early Autotram with two semi-trailers [9]. An FF controller depends on the desired kinematics from the desired path and an FB controller uses an inverse vehicle model to suppress disturbance. This approach is extended to an improved Autotram (Figure 2a) and six-axle Autotram with the axle virtualization method [10,11]. The author also designed another extended Ackermann-steering tracking control method in which there is steady circle steering together with a first-order inertial process instead of the desired path. The simpler approach is efficient enough to execute in a vehicle electronic control unit (ECU). Wagner's extended Ackermann-steering controller was considered to steer too early, which resulted in the latter modules moving to the inner side of a curve [12]. A Time delay unit was added to the first-order inertial to overcome this problem, and it works. Peng and Xiao also proposed similar methods for ART aiming at optimizing the transient process by adding time delay or increasing the order of the inertial unit [13,14]. But the time delay predicted by the current speed may not work well when speed changes frequently, considering the amazing vehicle length. A higher-order inertial unit can also perform limited improvement compared to a first-order inertial unit.

Zhang emphasized reference path generation as the first urgent problem ignored by most researchers [15,16]. A segmentation path generation method utilizes data compression and filtering to reduce data length efficiently. Then, two controllers for front axles and rear axles are designed. The front axles controller is a coordinated steering controller. The rear axles controller is a preview feedback steering controller to realize path following. He also proposed an adaptive tracking control method for all semi-trailers based on lateral deviation and yaw error [17]. However, there is no precision analysis about the generated path, which can be as high as 0.5 m in [18]. A road line recognition-based FF-FB tracking control method for DRT is described with up to 14 cameras around the vehicle [19]. FF control is achieved by road curve and FB control uses a proportional–integral (PI) controller based on lateral deviation. Wang [20] proposed a reconfigurable dynamic model of SRT with the relationship between generalized forces and tires. Improved model predictive control (MPC) is designed to allocate generalized forces and steering angles, which are calculated from generalized forces.

Leng [21] proposed the optimal acceleration tracking control method for a gantry virtual track train (GVTT) with a locking mechanism between vehicle modules. A local tracking objective function was used to calculate the optimal acceleration and steering angles. In another work [22], he achieved the decoupling of the axles' motion and applied an adaptive preview proportional–integral–differential (PID) algorithm to tracking control.

K.-H. MOON [23,24] tested the Phileas (BRT) controller in HIL mode to acquire the real input–output response. A new tracking control method based on the test data is proposed to improve the original one. The key point is to adjust the steering center of each module according to the speed, first-axle angle and articulated angle. It also provides many engineering details to work better in the real world. Jujnovich [25] developed a combined tracking controller including different low and high speed PID controllers for heavy articulated trucks. Side-slip angles are calculated as the controller input from kinematics at a low speed or estimated from dynamics at a high speed. Oreh [26] proposed a fuzzy controller based on the desired articulation angle and speed level to steer the trailer wheels of heavy trucks. Ritzen [27] designed nonlinear kinematic control strategies for a tractor–trailer robot using feedback linearization and backstepping methods.

In previous works, virtual rail was constructed in various forms so that the desired vehicle state or state error can be obtained to develop algorithms. Interoceptive or exteroceptive sensors [28] are used, and some applications are listed below. There is one thing to make clear: items 1~3 may not only use exteroceptive sensors.

1. Computer vision-based deviation measurements need cameras installed, a clear lane line and proper lighting conditions;
2. A magnetic guidance system with magnetic markers buried in the road surface and vehicle magnetic sensors;
3. Precise navigation systems like a differential global positioning system (DGPS) or integrated navigation system (INS);
4. Estimation method based on interoceptive sensors.

Tracking control with only interoceptive sensors is more fundamental for several reasons. First, VRTs usually travel on specific lines with extra infrastructures. But the ability to run on any urban roads like buses is also necessary. Then, exteroceptive sensors are more likely affected and even fail in some scenes when interoceptive sensors can always work well. Finally, the interoceptive sensors-based tracking control method can become a critical part of the advanced approach with powerful exteroceptive sensors.

In fact, ART has the DGPS or computer vision tech inside mainly for ADAS functions and line scheduling management. Phileas relies on special road-buried magnetic markers to realize auto guiding. Siemens opti-guide TEOR uses cameras to run on its own transit-only corridors like LRT. All of the above can run out of fixed lines without DGPS, cameras or magnetic virtual rails with interoceptive sensors. The interoceptive sensors-based method somehow can be seen as a feedforward control method. It has no conflicts with the advanced feedback approach, which leads to higher performance or safety.

Interoceptive sensor-based methods are not perfect anyway. For example, dead reckoning in path generation has an inherent drift problem caused by model uncertainties and integration calculation without an external reference. Fortunately, VRT tracking control only uses short time estimation and local results, which avoid unbounded drift. Some estimation methods can perform better than dead reckoning with a high complexity like Kalman filters. Other no-path methods seem too rough and experience-dependent.

Dynamic or kinematic models are another question for tracking control. Generally, dynamic models take account of more factors, which is more valid in various conditions. However, tremendous, highly qualified model parameters make it difficult to realize in real time control, or only simplified dynamic models with low precision can be used. For more complex VRTs whose highest speed is less than 70 km/h and running scenarios are smooth motion as public vehicles, dynamic models can be used to analyze extreme responses like vehicle braking and yaw stability in the designing period. A kinematic model is more proper for developing a tracking control method for its simple structure.

The main contributions for VRT tracking control for this paper are listed below:

- Only kinematic models and interoceptive sensors are used to ensure the most fundamental driving function.
- Time delay or inertial unit optimization by experience in previous VRT control are replaced with standard process.
- Inaccurate reference paths, computational and storage pressure for ECU and even actuator delay are discussed and solved.

The paper is organized as follows. Section 2 introduces the vehicle kinematic model and possible sensor configurations. Front axles virtualization control is developed in Section 3, which simplifies the whole control structure. Section 4 realizes a reference path generation, storage and updating process as the basis of rear axles tracking control in

Section 5. In Section 6, HIL simulation and related results are shown and discussed to verify the effectiveness of our approach. Finally, some conclusions are summarized in Section 7.

2. Vehicle Kinematic

A single-track kinematic model of a VRT is proposed under the following assumptions [29]:

1. Only plane movement is considered, and vertical movement is ignored;
2. The left and right wheels can be seen as a single wheel. Wheel slip is ignored, which means that the wheel speed coincides with the wheel plane;
3. Vehicle modules are treated as rigid bodies;
4. Head and tail axles are active driven axles under the control of a traction control system with a speed coordination mechanism. Thus, only one valid vehicle speed is used;

These assumptions come from the reality that VRTs run on urban roads with solid, flattened surfaces. High-speed driving (>40 km/h) only occurs on relatively straight roads, and the turning speed is less than 30 km/h or lower on a road curvature radius less than 50 m. Under these conditions, wheel slip can be neglected, and a kinematic model can be applied with proper precision. The structure of a single-track kinematic model of a VRT is shown in Figure 3, and related parameters are listed in Table 2. The main coordinate systems used here are the global coordinate system (GCS) and vehicle coordinate system (VCS).

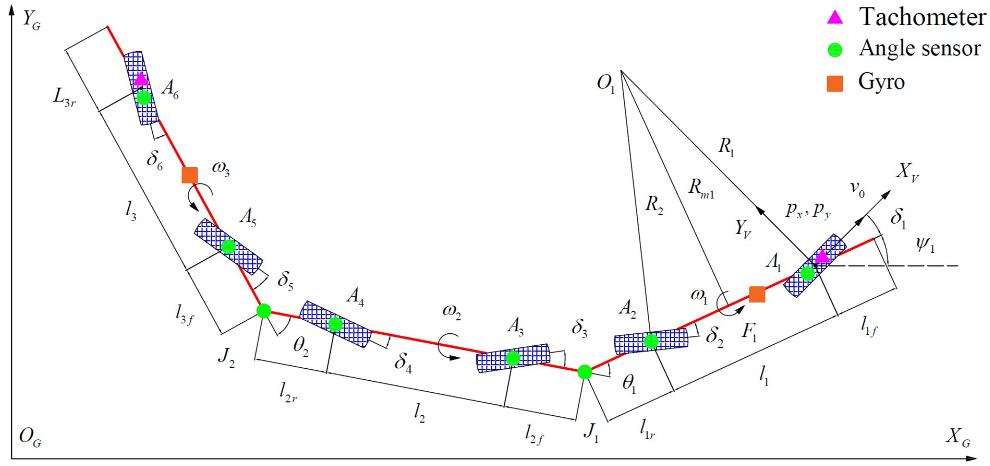


Figure 3. Single-track kinematic model structure of VRT.

Table 2. Kinematic model signs and parameters.

Parameters	Description	Value
A_i	ith$(i = 1 \sim 6)$ axle	/
J_i	ith$(i = 1 \sim 2)$ hinge joint	/
δ_i	ith$(i = 1 \sim 6)$ axle steering angle	From angle sensor
θ_i	ith$(i = 1 \sim 2)$ hinge joint angle	From angle sensor
ω_i	ith$(i = 1 \sim 3)$ module angular velocity	From calculation or gyro
l_i	ith$(i = 1 \sim 3)$ module axle distance	$l_1 = l_3 = 6.0$ m, $l_2 = 6.5$ m
l_{if}	ith$(i = 1 \sim 3)$ module front overhang	$l_{1f} = 1.8$ m, $l_{2f} = l_{3f} = 2.5$ m
l_{ir}	ith$(i = 1 \sim 3)$ module rear overhang	$l_{3r} = 1.8$ m, $l_{1r} = l_{2r} = 2.5$ m
p_x, p_y, ψ_1	A_1 center position and yaw angle in GCS	/
v_0	Linear velocity of the first wheel	From tachometer

The steering angle δ_i comes from the steering by wire (SBW) system. The original measurement may not be the steering angle of the axle center, and it should be transferred according to Ackermann steering geometry. The linear velocity of the first wheel v_0 comes from a tachometer based on the driven motor rotary speed and tire size. For vehicle symmetry, there is also a tachometer on A_6, which is ignored here. A tachometer and angle sensor along with its v_0, δ_i, θ_i are the minimum sensor configurations for tracking control. But we also added two gyros for comparison with the gyro drift estimated and removed when the vehicle powers on. All angles are signed with left turning positive and right turning negative.

The kinematic state vector x is chosen as $x = [p_x, p_y, \psi_1, \theta_1, \theta_2]^T$, which can totally describe the vehicle position and attitude. The kinematic state function is expressed as

$$\dot{x} = f(x, v_0, \delta_1, \omega_1, \omega_2, \omega_3) = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\psi}_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} v_0 \cos(\delta_1 + \psi_1) \\ v_0 \sin(\delta_1 + \psi_1) \\ \omega_1 \\ \omega_1 - \omega_2 \\ \omega_2 - \omega_3 \end{bmatrix} \tag{1}$$

The latter module yaw angle ψ_2, ψ_3 is

$$\psi_2 = \psi_1 - \theta_1, \psi_3 = \psi_2 - \theta_2 \tag{2}$$

The hinge joint angle θ_1, θ_2 can be obtained directly from sensors. ω_i can be obtained as below, in which v_{ix} is the longitude velocity of the ith module. For the first module, we have $v_{1x} = v_0 \cos \delta_1$. The calculation of v_{2x}, v_{3x} based on the velocity and steering angles is described in the next chapter. Furthermore, ω_1 can also come from gyro.

$$\omega_i = \frac{v_{ix}(\tan \delta_i - \tan \delta_{i+1})}{l_i}, (i = 1 \sim 3) \tag{3}$$

The instantaneous center radius of A_1, A_2 and the whole first module are R_1, R_2, R_{m1} in the form below when $\delta_1 \neq \delta_2$. A turning radius is also signed with a left turning positive.

$$R_1 = \frac{l_1}{\cos \delta_1 (\tan \delta_1 - \tan \delta_2)}, R_2 = \frac{l_1}{-\cos \delta_2 (\tan \delta_1 - \tan \delta_2)}, R_{m1} = \frac{l_1}{\tan \delta_1 - \tan \delta_2} \tag{4}$$

According to (4), only when $\delta_2 = -\delta_1$ will A_2 have the same turning radius as well as the same path as A_1. But using this strategy directly will lead to a dangerous rear swing out problem, which should be solved later. For a VRT's low and medium traveling speeds, anti-phase or zero-phase steering can be applied, which means $\delta_1 \delta_2 \leq 0$, as the latter module [30]. We will then start designing our hierarchical controller based on front axles virtualization.

3. Front Axles Virtualization Control

There are a total of seven independent control inputs (one driven speed, six steering angles) and five state variables, which may lead to conflict. Thus, two redundant control inputs must cooperate with others or the control will fail. Intuitively speaking, the driven speed is independent and A_1, A_6 steering is relatively free to steer with less constraints compared to the axle's neighbor to hinge joints (A_2, A_3 or A_4, A_5).We can imagine a disaster when A_2, A_3 steer oppositely with a terrible internal force in the hinge joint. Thus, two redundant control inputs should be chosen from one axle separately in A_2, A_3 or A_4, A_5. In fact, the intuitive result exists in some theoretical evidence in [31,32] where the dynamic relative gain array method is used to analyze the effect of turning for every single axle. A_3, A_5 are chosen as coordinated control axles according to their relatively low effect levels for turning, which are also called virtual axles.

Virtual axles refer to axles that only provide ground support instead of a vehicle turning function, as if they can be canceled when handling the turning motion of a vehicle. According to the definition above, we can remove A_3 first and analyze the moving direction of the A_3 mount point to realize the axles' virtualization. Figure 4 shows the motion decomposition of the A_3 axles' virtualization. For convenient expression, the process is divided into two steps; the first step is for the first module in the right part of Figure 4, and the other is for the second module in the left part.

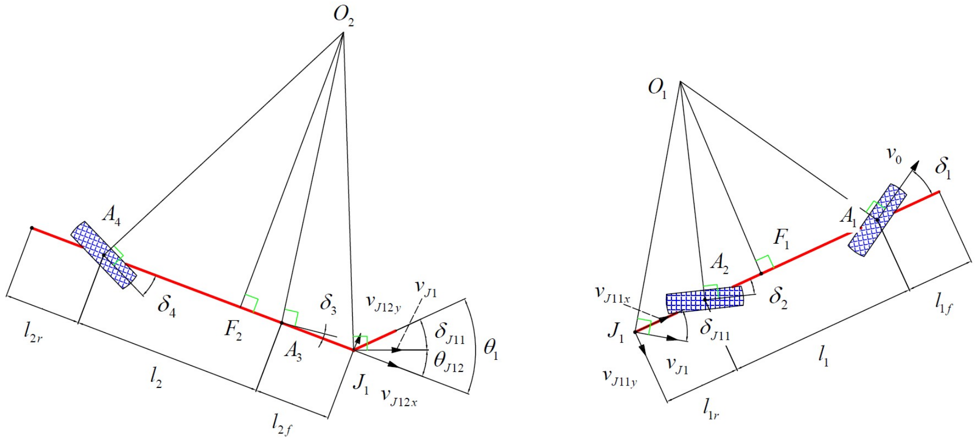


Figure 4. Axles' virtualization.

In the right part of Figure 4, v_{J1} is the joint J_1 velocity and δ_{J11} is the angle between v_{J1} and the first module, which is negative here. We have

$$\delta_{J11} = -\angle J_1O_1F_1 = -a\tan(\frac{l_{1r} + |A_2F_1|}{|O_1F_1|}) = -a\tan\left(\frac{l_{1r}}{l_1}\tan\delta_1 - \frac{l_1 + l_{1r}}{l_1}\tan\delta_2\right) \quad (5)$$

Although (5) is obtained when $\delta_1 > 0$, it can be verified as correct in other conditions. In the left part of Figure 4, δ_{J12} is the angle between v_J and the second module, expressed as

$$\delta_{J12} = \theta_1 + \delta_{J11} \quad (6)$$

Then, the rotation of the second module is determined by δ_{J12} and δ_4.The angle between the velocity on A_3 and the second module δ_{3C} should be the steering control output value of A_3.

$$\delta_{3C} = a\tan(\frac{|A_3F_2|}{|O_2F_2|}) = a\tan\left(\frac{l_2}{l_2 + l_{2f}}\tan\delta_{J12} + \frac{l_{2f}}{l_2 + l_{2f}}\tan\delta_4\right) \quad (7)$$

We can obtain δ_{5C} with a similar process on the second and third modules, in which δ_{J22} is the angle between the joint J_2 velocity and the second module and δ_{J23} is the angle between the joint J_2 velocity and the third module.

$$\delta_{J22} = -a\tan\left(\frac{l_{2r}}{l_2}\tan\delta_3 - \frac{l_2 + l_{2r}}{l_2}\tan\delta_4\right) \quad (8)$$

$$\delta_{J23} = \theta_2 + \delta_{J22} \quad (9)$$

$$\delta_{5C} = a\tan\left(\frac{l_3}{l_3 + l_{3f}}\tan\delta_{J23} + \frac{l_{3f}}{l_3 + l_{3f}}\tan\delta_6\right) \quad (10)$$

Some may notice that the calculations above only care about the velocity orientation instead of the amplitude. The reason for this is that the velocity amplitude naturally satisfies constraints with $A_2 \sim A_5$ passive wheel rotation. In order to achieve delay compensation, in a later chapter, we will talk about the velocity calculation here. v_{J11x} is the projection of v_{J1} on the first module's longitude orientation, which equals v_{1x}.

$$v_{J1} = \frac{v_{J11x}}{\cos\delta_{J11}} = \frac{v_{1x}}{\cos\delta_{J11}} \quad (11)$$

The longitude velocity of the second module $v_{2x} = v_{J12x}$ can be obtained from v_{J1} and in the same way with v_{3x}. The front axle's virtualization is completed, and control outputs are determined, which means A_3, A_5 can be removed in the following control. We also obtain access to v_{ix} based on the vehicle velocity, steering angles and joint angles.

4. Path Generation and Storage

Some reference or desired characters should be provided to execute A_2, A_4, A_6 tracking control. Most of the existing algorithms take the reference path in discrete coordinate points or just take the steady circle state with an unreliable time delay and transition method. The reference path is often chosen as the history behavior of A_1, A_2 or even J_1, which is precise enough with exteroceptive sensors. Then, the matching points are used to extract a state error like lateral or yaw deviation. But things change for interoceptive sensors, whose integral calculation (dead reckoning) leads to drift as time goes on. Although we are only concerned with about a 30 m length of the history behavior, the drift is also too large to ignore. Researchers proposed an optimal estimation method like Kalman filters to suppress drift, which indeed works with increasing calculation costs. Furthermore, the efficiency storage of the reference path is also as important as the algorithm complexity for the vehicle's ECU.

In this part, we propose a new path generation and storage method for curvature matching instead of deviation calculation. It is based on the assumption that drift exists but cannot change suddenly for its integral calculation. The matching point is relatively precise and the curvature is what we want. Path segmentation and a first input first output (FIFO) queue are used to solve the data storage and processing. All of the methods above are expected to execute the algorithm in the vehicle's ECU. Here, the A_1 center history path is chosen as the reference path, so the first three state variables are used in model discretization with a cycle period $\Delta t = 0.01s$.

$$\begin{array}{l} \psi_{1/k} = \psi_{1/k-1} + \omega_{1/k}\Delta t \\ p_{x/k} = p_{x/k-1} + v_{0/k}\cos(\delta_{1/k} + \psi_{1/k})\Delta t \\ p_{y/k} = p_{y/k-1} + v_{0/k}\sin(\delta_{1/k} + \psi_{1/k})\Delta t \end{array} \tag{12}$$

Starting with the given initial values, Formula (16) can run cycle by cycle, which is called dead reckoning. The path we need is the nearest time interval history data whose total length can cover $A_1 \sim A_6$ at least. The direct use of (16) leads to an uncertain data length and a huge amount of history data. Flexible data length maintenance is difficult, and the storage of every single point is unnecessary. The path segmentation data structure and storage are used here with a fixed data length and less calculation. The path segmentation length threshold L_s is set as $0.3m$, which is a compromise for the algorithm complexity, storage capacity and precision. The segmentation number N_s (fixed FIFO length) is 100, which is larger than the quotient between the $A_1 \sim A_6$ distance and L_s. Each element in FIFO consists of four values, e.g., $E_i = \left[S_i, c_i, p_{xsi}, p_{ysi}\right]^T (i = 1 \sim 100)$, in which i represents the ith element, S_i is the real segmentation length, c_i is the mean segmentation curvature and p_{xsi}, p_{ysi} is the segmentation end point coordinate. The algorithm process is described in Figure 5 with the following steps:

1. Initialization is completed as shown below when the VRT is in a straight parking state. $S_{0/0}, B_{0/0}$ are the segmentation curve length and curvature integral variable. $N_{0/0}$ represents the integral times. $\omega_{1/0}, \delta_{1/0}, v_{0/0}$ are initialized with sensor measurements and calculations.

$$\begin{array}{l} p_{x/0} = 0, p_{y/0} = 0, \psi_{1/0} = 0; \\ E_i = \left[S_i, c_i, p_{xi}, p_{yi}\right]^T = \left[L_s, 0, -iL_s, 0\right]^T \\ S_{0/0} = 0, B_{0/0} = 0, N_{0/0} = 0 \end{array} \tag{13}$$

2. At time k when $v_{0/k} = 0$, nothing needs to be done but to enter the next cycle period with $k \to k+1$. Otherwise, if $v_{0/k} > 0$, Formulas (16) and (18) are carried out, where $c_{1/k}$ is the running curvature of A_1.

$$\begin{array}{l} S_{0/k} = S_{0/k-1} + v_{0/k}\Delta t \\ B_{0/k} = B_{0/k-1} + c_{1/k} = B_{0/k-1} + \frac{\omega_{1/k}}{v_{0/k}} \\ N_{0/k} = N_{0/k-1} + 1 \end{array} \tag{14}$$

3. Step 2 continues with $k \to k+1$ in every cycle period until $S_{0/k} > L_s$ and it is time to execute the FIFO update operation. Dequeue operation removes element E_{100} and other elements shift to the nest position in the sequence. E_0 with the data below replaces E_1 to complete the enqueue operation.

$$\begin{array}{l} E_0 = \left[S_0, c_0, p_{xs0}, p_{ys0}\right]^T = \left[S_{0/k}, \frac{B_{0/k}}{N_{0/k}}, p_{x/k}, p_{y/k}\right]^T \\ E_i \to E_{i+1} (i = 0 \sim 99) \end{array} \tag{15}$$

4. Resetting $S_{0/k}, c_{0/k}, n_{0/k}$ to zero and jumping to step 2 with $k \to k+1$ to start the next cycle period.

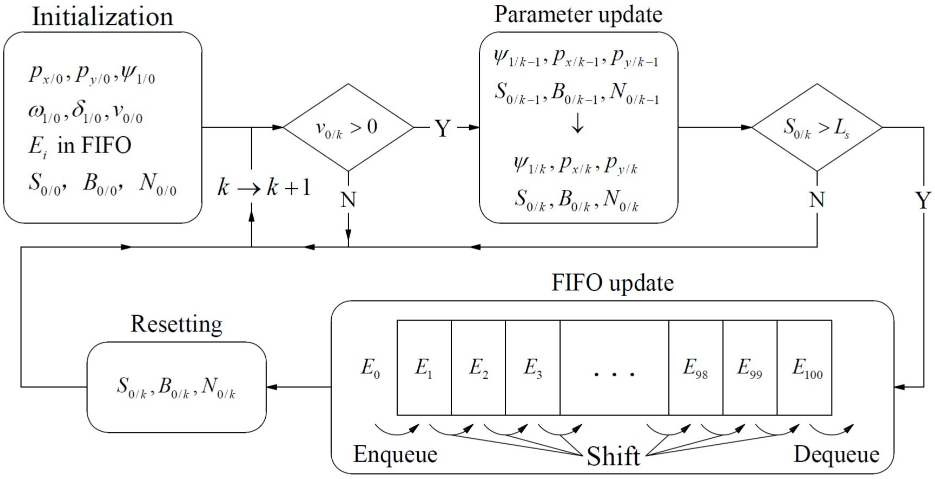

Initialization
$p_{x/0}, p_{y/0}, \psi_{1/0}$
$\omega_{1/0}, \delta_{1/0}, v_{0/0}$
E_i in FIFO
$S_{0/0}$, $B_{0/0}$, $N_{0/0}$
$v_{0/k} > 0$
Y
N
$k \to k+1$
Parameter update
$\psi_{1/k-1}, p_{x/k-1}, p_{y/k-1}$
$S_{0/k-1}, B_{0/k-1}, N_{0/k-1}$
↓
$\psi_{1/k}, p_{x/k}, p_{y/k}$
$S_{0/k}, B_{0/k}, N_{0/k}$
$S_{0/k} > L_s$
N
Y
FIFO update
Resetting
$S_{0/k}, B_{0/k}, N_{0/k}$
E_0
E_1
E_2
E_3
. . .
E_{98}
E_{99}
E_{100}
Enqueue
Shift
Dequeue

Figure 5. Path generation and storage algorithm.

The generated reference path is a local path stored in a finite fixed data length with data updates and an abandon mechanism. The path information helps us to realize rear axles tracking control in the next chapter.

5. Rear Axles Tracking Control

The match point is defined as the nearest point in the reference path for A_2, A_4, A_6, which should be determined first. The coordinate of A_2, A_4, A_6 is $p_{ix}, p_{iy} (i = 2, 4, 6)$, with time k omitted, which can be obtained with a chain calculation. The coordinates of A_4 are listed here.

$$\begin{array}{l} p_{4x} = p_{2x} - l_{1r}\cos(\psi_1) - (l_{2f} + l_2)\cos(\psi_2) \\ p_{4y} = p_{2y} - l_{1r}\sin(\psi_1) - (l_{2f} + l_2)\sin(\psi_2) \end{array} \tag{16}$$

The formula above provides the current position of rear axles without SBW actuator delay considered. SBW is a complex hydro mechatronic system with an obvious delay character between the control command and execution in place. Thus, the current position of the rear axles is not enough, and we should predict the position $\hat{p}_{ix}, \hat{p}_{iy} (i = 2, 4, 6)$ after delay time T_d.

$$\begin{array}{ll} \hat{p}_{ix} = p_{ix} + \frac{v_{jx}}{\cos(\delta_i)} T_d \cos(\psi_j + \delta_i) & \\ & i = 2, 4, 6, j = \frac{i}{2} \\ \hat{p}_{iy} = p_{iy} + \frac{v_{jx}}{\cos(\delta_i)} T_d \sin(\psi_j + \delta_i) & \end{array} \tag{17}$$

Then, the distance between $\hat{p}_{ix}, \hat{p}_{iy} (i = 2, 4, 6)$ and any point in the reference path can be obtained to find the match point. Traversal is the naivest approach, and here, we develop a heuristic search method to improve efficiency. The basic idea comes from the distance between axles, which can provide tough search information to narrow the search interval in Figure 6, where the start search and end search points should be determined.

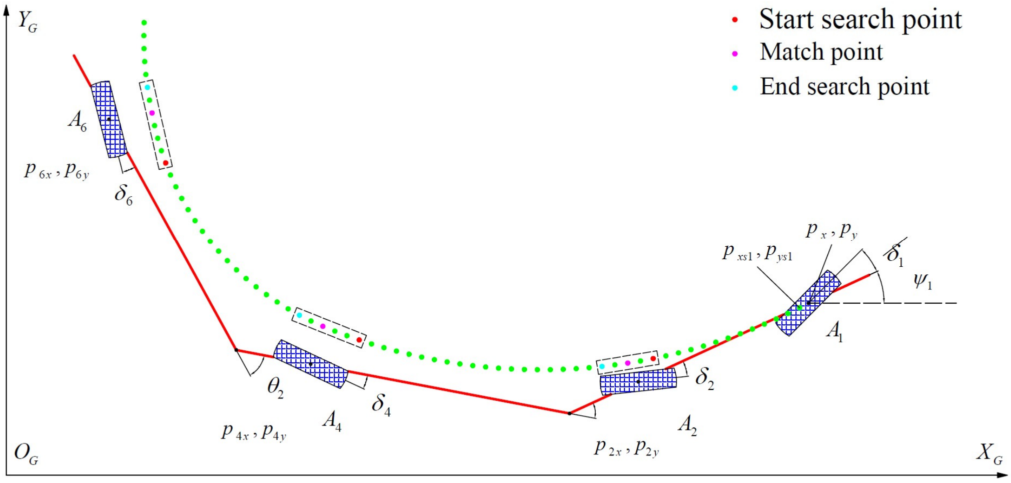


Figure 6. Heuristic search in the reference path.

The curve length must be larger than the straight length between axles. Let us take A_2 for example; the curve length between A_2 and its match point is larger than l_1.The index of the start and end points is defined as $N_{is}, N_{ie}, (i = 2, 4, 6)$ in the reference path points. N_{is} is obtained from l_{1i}, the segmentation curve length S_i as well as the actuator delay.

$$N_{is} = \underset{N_{is}}{\operatorname{argmin}} \left((S_0 + \sum_{m=1}^{N_{is}} S_m) > (l_{1i} - v_{jx} T_d) \right), i = 2, 4, 6, j = \frac{i}{2} \tag{18}$$

Three factors should be considered for the end search point, including the maximum path estimation error, maximum tracking error and real length error between the arclength and straight length. The maximum path estimation error means a dead reckoning error, which is larger from A_2, A_4, A_6. Here, we take 0.5 m, 0.75 m, 1.0 m for A_2, A_4, A_6 according to experience. The maximum tracking error is 0.5 m based on the road width (3.75 m) and vehicle width (2.65 m). The final factor is difficult to estimate for complex vehicle motion. Thus, we also take 0.5 m, 0.75 m, 1.0 m for A_2, A_4, A_6. By summing the factors, we obtain the interval search length for A_2, A_4, A_6, which is 1.5 m, 2.0 m, 2.5 m. By dividing versus L_s and rounding up, the integers as indexes are obtained.

$$N_{2e} = N_{2s} + 5, N_{4e} = N_{4s} + 7, N_{6e} = N_{6s} + 9 \tag{19}$$

The final match point indexes for A_2, A_4, A_6 are N_{2m}, N_{4m}, N_{6m}, with N_{2m} expressed below.

$$N_{im} = \underset{N_{im}}{\operatorname{argmin}} \left(\sqrt{(\hat{p}_{ix} - p_{xN_{im}})^2 + (\hat{p}_{iy} - p_{yN_{im}})^2} \right), N_{im} \in [N_{is}, N_{ie}], i = 2, 4, 6 \tag{20}$$

We just search a total of 5 + 7 + 9 + 3 = 24 points instead of 300 points to accelerate the algorithm. The relationship of the steady circle state of A_2, A_4, A_6 in Figure 7 is discussed below to access the final control output.

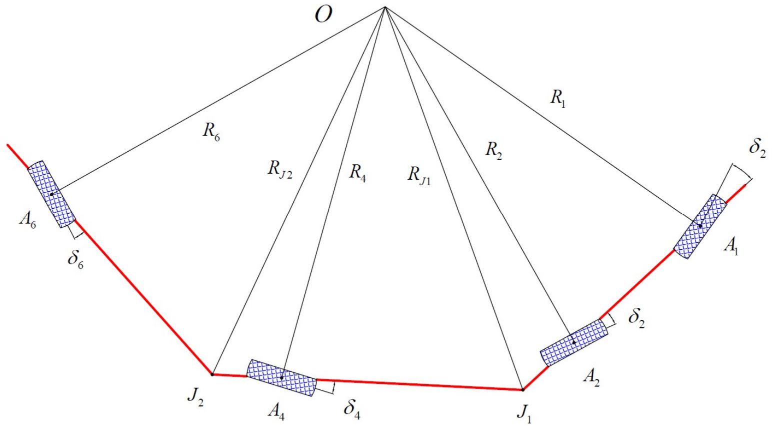


Figure 7. Steady circle state.

First, it us assumed that $R_1 = R_2 = R_4 = R_6 = R_s > 0$, and we already have $\delta_{2s} = -\delta_{1s}$ for A_2.

$$\delta_{2s} = -\delta_{1s} = -\arcsin(\frac{l_1}{2R_s}) \tag{21}$$

According to cosine theorem,

$$R_{J1}^2 = R_s^2 + l_{1r}^2 - 2R_s l_{1r}\cos(\frac{\pi}{2} - \delta_{2s}) = R_s^2 + l_{1r}^2 + l_{1r}l_1 \tag{22}$$

Then, for A_4, we can obtain δ_{4s} below.

$$\delta_{4s} = \arccos\left(\frac{R_s^2 + (l_2 + l_{2f})^2 - R_{J1}^2}{2R_s(l_2 + l_{2f})}\right) - \frac{\pi}{2} \tag{23}$$

The same applies for A_6 and δ_{6s}.

$$R_{J2}^2 = R_s^2 + l_{2r}^2 - 2R_s l_{2r}\sin(\delta_{4s}) \tag{24}$$

$$\delta_{6s} = \arccos\left(\frac{R_s^2 + (l_3 + l_{3f})^2 - R_{J2}^2}{2R_s(l_3 + l_{3f})}\right) - \frac{\pi}{2} \tag{25}$$

For $R_s < 0$,we just replace R_s with $|R_s|$ and take the negative result of δ_{is}. For $R_s = \infty$, which means traveling straightly, $\delta_{is} = 0$. δ_{is} is only relevant to the steady circle radius R_s, which is convenient in designing our rear axle tracking control. For rear axles, we already have the curvature stored in match points, which can be transferred into the instantaneous turning radius directly. Replacing the steady circle radius R_s with the corresponding instantaneous radius, we obtain the final control outputs for rear axles. For A_2, the sign of the match point curvature is out of consideration.

$$\delta_{2c} = -\arcsin(\frac{l_1 c_{N_{2m}}}{2}) \tag{26}$$

For A_4, when $c_{N_{2m}} > 0$,

$$\delta_{4c} = \begin{cases} \arccos\left(c_{N_{4m}}\frac{(l_2+l_{2f})^2-l_{1r}^2-l_{1r}l_1}{2(l_2+l_{2f})}\right) - \frac{\pi}{2}, c_{N_{4m}} > 0 \\ \frac{\pi}{2} + \arccos\left(c_{N_{4m}}\frac{(l_2+l_{2f})^2-l_{1r}^2-l_{1r}l_1}{2(l_2+l_{2f})}\right), c_{N_{4m}} < 0 \\ 0, c_{N_{4m}} = 0 \end{cases} \tag{27}$$

The result for δ_{6c} is similar, which is neglected here. The whole controller architecture is shown in Figure 8.

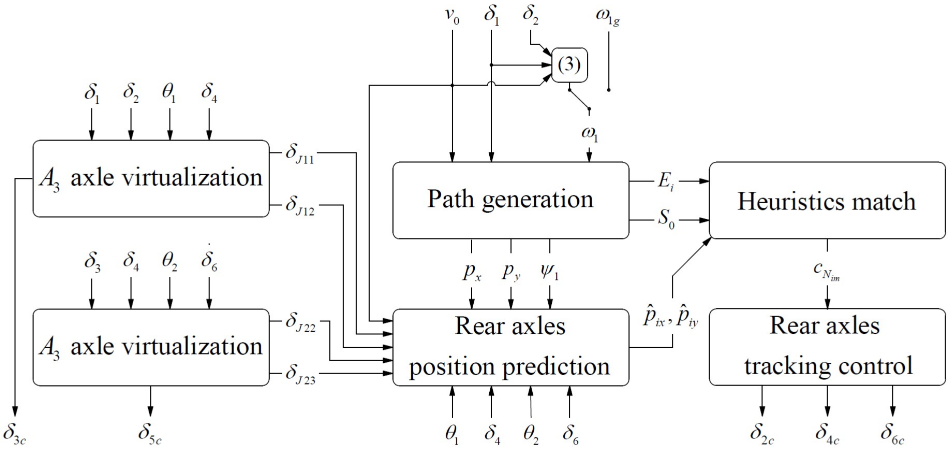

v_0
δ_1
δ_2
ω_{1g}
(3)
ω_1
δ_1 δ_2 θ_1 δ_4
A_3 axle virtualization
δ_{J11}
δ_{J12}
Path generation
E_i
S_0
Heuristics match
δ_3 δ_4 θ_2 $\dot{\delta}_6$
p_x p_y ψ_1
$c_{N_{im}}$
A_3 axle virtualization
δ_{J22}
δ_{J23}
Rear axles position prediction
$\hat{p}_{ix}, \hat{p}_{iy}$
Rear axles tracking control
δ_{3c}
δ_{5c}
θ_1 δ_4 θ_2 δ_6
δ_{2c} δ_{4c} δ_{6c}

Figure 8. Controller architecture.

Variables in Figure 8 with a single link are input/output variables, while others are intermediate variables. (3) means using Equation (3) and ω_{1g} means ω_1 from gyro. There exists a toggle switch for choosing the angular velocity sources. In fact, the reference path stores the curvature rather than the radius, whose purpose is to avoid an infinite radius. A small threshold can be used in engineering to judge whether the curvature is zero. When the vehicle speed is more than 40 km/h, the latter axles are locked to 0 by a soft threshold like Phileas' steering attenuation strategy in [23,24]. Although using a steady circle radius to derive the conclusion, the control method applies the instantaneous radius, which is updated every time for every rear axle. Any transit process like time delay and an inertial link are naturally considered in the reference path. And even the rear axles' actuator delay is also included to improve the vehicle response.

6. Simulation and Results

An HIL simulation is constructed to verify the effectiveness and efficiency of the algorithm proposed above. The hardware refers to an ECU with an NXP MPC5748 processer, in which a single core is used to execute our algorithm in 0.01 s of cycle time. The vehicles' dynamic model runs on the real-time simulation machine. Controller area networks (CAN) are used to transmit input/output information between two facilities. The simulation is shown in Figure 9.

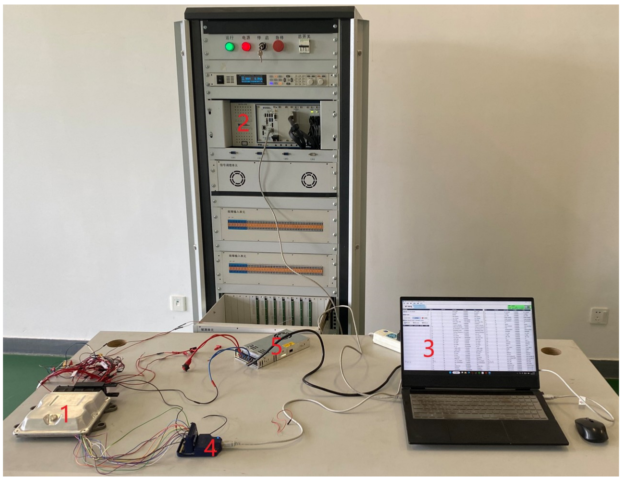


Figure 9. HIL simulation (1. ECU, 2. Simulation machine, 3. Laptop, 4. Debugger, 5. 24 V power).

The reference path is shown in Figure 10, including the combination of straight lines, a single curve and a continuous changing curve, which is representative in the real world.

The first axle is controlled by a preview driver model whose path is considered as a real reference path. The initial simulation results at 15 km/h with no gyro are shown in Figure 11.

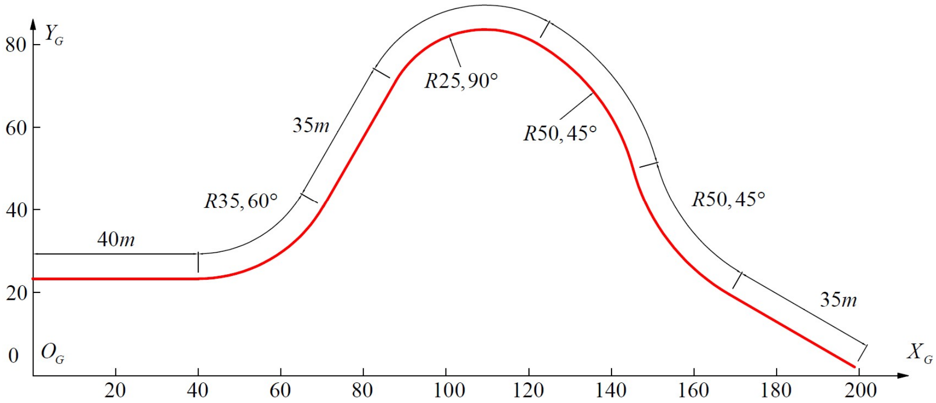


Figure 10. Reference path.

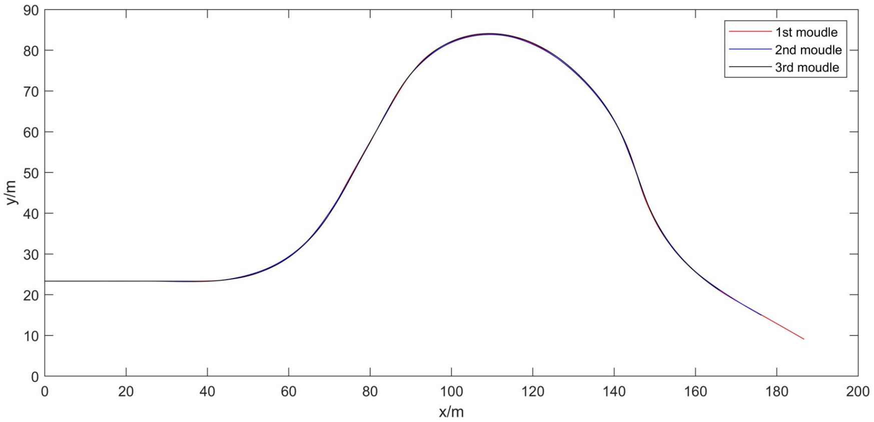


Figure 11. Initial simulation result.

We will first talk about reference path generation precision for both no-gyro and gyro conditions. Figure 12a shows that the maximum lateral estimation error is about 1.6 m for no gyro and 0.8 m for a gyro situation after running about 120 m. The yaw error is about 2° for no gyro and 1° for a gyro situation. As the time goes on, both the lateral and yaw estimation error will be out of control (absolute error). Then, we align three paths at a fixed point (x = 110, for example), and the result is shown in Figure 12b (relative error). The maximum lateral estimation error is about 0.37 m for no gyro and 0.23 m for a gyro situation at the end of the VRT. The relative error just considers the error about 30 m along the vehicle length, and it is under control from diverging. We did not execute alignment, and it is accomplished within the rear axles' control implicitly.

The tracking control evaluation indicators are defined below:

- Lateral deviation: a module geometric center path lateral deviation to the first module. This indicator shows the overall module tracking deviation.
- Swept path width: a vehicle external contour envelope, which is positively correlated with the lateral deviation, module yaw angle and turning radius and is more comprehensive. It is obtained by the path of single-module external rectangle contour corner points and side middle points, which consist of six paths for a single module. The road margin width here is 3.75 m.
- The result of no gyro is shown in Figures 13 and 14, with a running speed of 15 km/h.

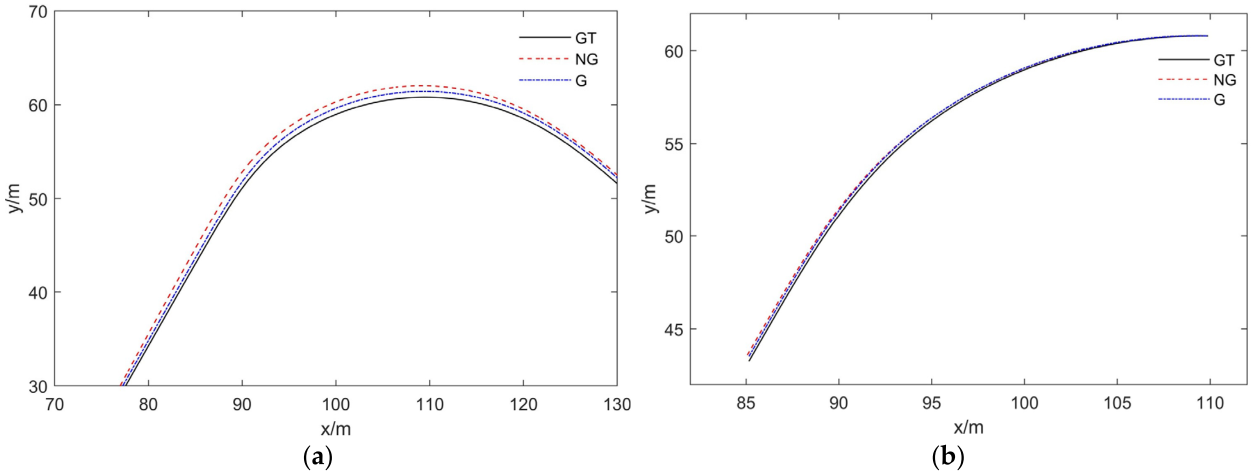


Figure 12. Path estimation results (GT: ground truth, NG: no gyro, G: gyro): (**a**) Absolute error; (**b**) Relative error (after alignment).

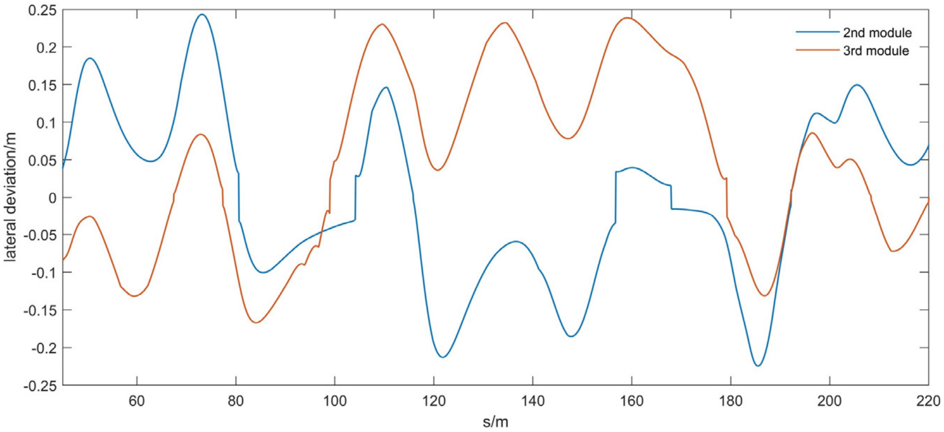


Figure 13. Lateral deviation (horizontal axis s means path length).

The maximum lateral deviation is within 0.25 m for the latter modules. The maximum swept path width is 3.60/3.36/3.18 for R25/R35/R50, respectively, which is always within the road margin. The maximum value occurs when the road curvature changes. In order to test the effect of the delay time T_d, we removed the prediction part (Formula (21)) in the same simulation. Then, the maximum lateral deviation increased to 0.32 m, which means that the prediction part indeed improved the control effect.

We then simulated different speed conditions without gyro. The maximum speed was set to 20/25/30 km/h for R25/R35/R50 turning to ensure ride comfort. The minimum speed was set to 10 km/h for all situations. The maximum lateral deviations are all within 0.25 m~0.3 m, which is better than those of Autotram or ART, with lateral deviations above 0.4 m. The results will be more uncertain when turning with speed change. Our approach can handle it for the match point changes when speed change. The maximum swept path results are listed below in Table 3.

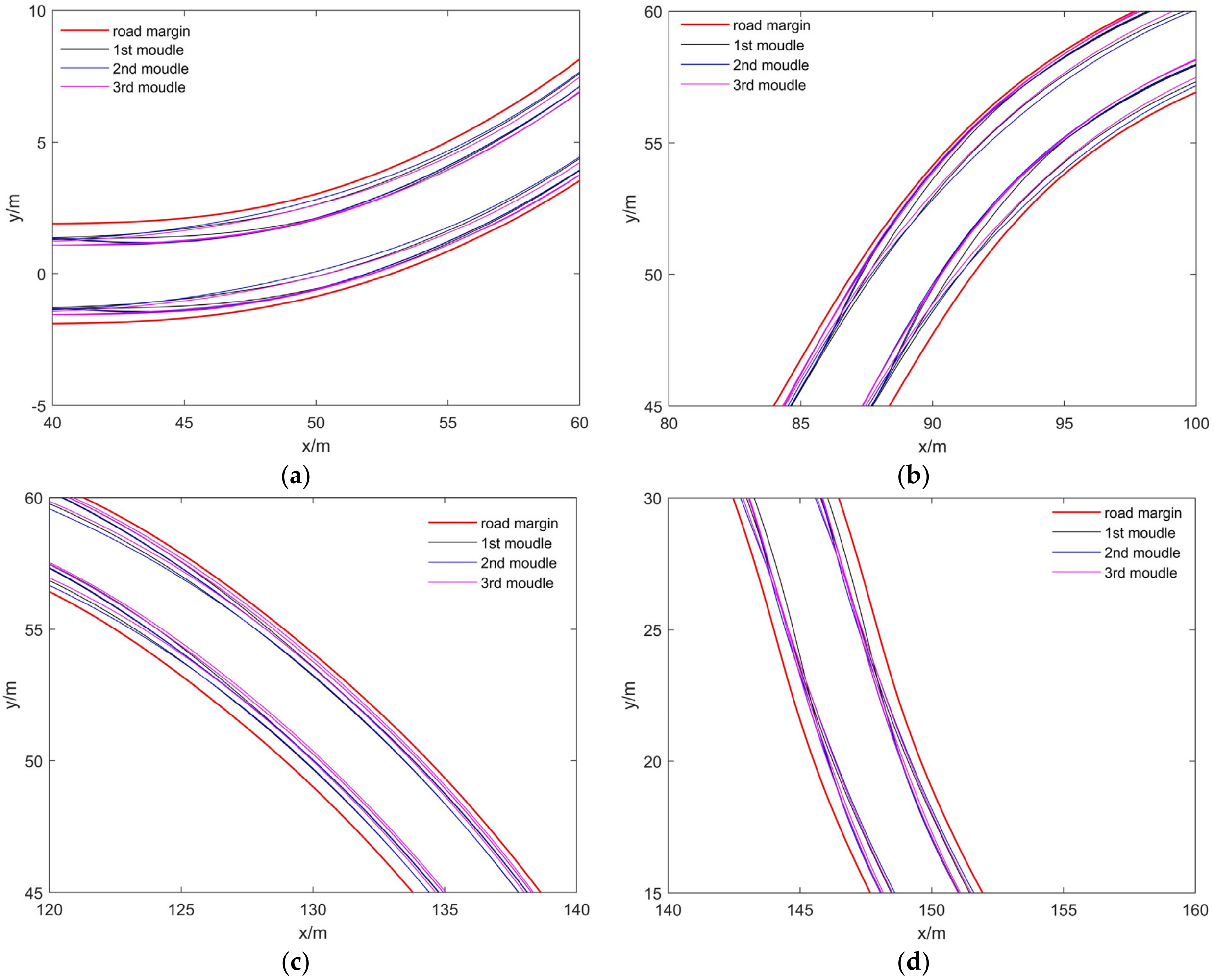


Figure 14. Swept path: (**a**) Straight line to the R35 curve; (**b**) Straight line to the R25 curve; (**c**) R25 curve to the R50 curve; (**d**) R50 curve to the inverse R50 curve.

Table 3. Swept path width for different situations.

Speed (km/h)	R25	R35	R50
10	3.56 m	3.33 m	3.15 m
15	3.60 m	3.36 m	3.18 m
20	3.66 m	3.43 m	3.20 m
25	-	3.52 m	3.24 m
30	-	-	3.34 m

Although the maximum lateral deviation did not change significantly, the maximum swept path width increased as the speed rose. The reason for this is that the prediction part only works for rear axles, leaving inadequate compensation for front axles actuator time delay. This leads to a little swing to the latter module, causing a larger swept path width. These adverse effects will become weaker as the road curvature decreases because of the change rate decreasing. In real applications, a large road curvature also owns a wider road margin, which grantees long vehicle trafficability.

The results using gyro as an angular velocity source did not differ significantly compared with those of a no-gyro situation for the small curvature estimation error. The minimum sensor configuration is enough for VRT tracking control, which consists of axle steering angle sensors, hinge joint angle sensors and tachometers. Gyro is optional, which can barely improve the control effect.

7. Conclusions

The tracking control approach proposed in this paper is totally based on a kinematic model and interoceptive sensors. Kinematic model parameters are easy to achieve, and interoceptive sensors lead to more independence for outer infrastructures. Some conclusions and prospects are listed below:

1. A minimum interoceptive sensor configuration is proposed for VRT. However, it is worth using gyros as a redundancy solution for vehicle safety. Furthermore, fault diagnosis and fault-tolerant control can also be developed. The method proposed here can be seen as a feedforward control method, essentially. Furthermore, advanced and complex feedback methods with more exteroceptive sensors can be developed based on this approach to realize high performance, even for an advanced driving assistance system (ADAS).
2. Kinematic estimation, segment storage and a queue update strategy are applied in path generation and storage. Fixed-length reference path maintenance is possible, and data quantity handling in a single time cycle can be compressed to a low level. A heuristic search method avoids 90% of invalid search points in a reference path. All efforts make it efficient enough to run on a real vehicle ECU in a 0.01 s time cycle, with engineering computation available.
3. The path estimation error is analyzed where a curvature instead of a coordinate is used to overcome the effect of the low precision of an estimated path. Few experience values are needed in controller design compared to the time delay or inertial unit design in Autotram or ART. That means the method can be easily modified when the vehicle structure changes without tremendous tests to obtain proper experience values.
4. Rear axles actuator time delay is considered and compensated, which works well in simulations. Although the quick response of rear axles helps to improve the time delay in front axles indirectly, the front axles actuator time delay is expected to be solved in the future, as well as sensor or CAN bus delay.
5. The final results are better than those of Autotram and ART in lateral deviation, and the swept path width is always within the road margin, which verifies the effectiveness of our approach.
6. Despite the advantage above, the model used here is simple, and lots of uncertainties are not considered. We proposed a curvature threshold and speed soft threshold here to improve the performance. More necessary if-then rules from simulation or vehicle testing need to be added in case of extreme conditions. We still cannot find a simple and effective enough dynamic model for tracking control like the classical 2-DOF lateral vehicle dynamic control model for cars. This will be the focus area for VRT tracking control in the future.

Author Contributions: Conceptualization, methodology, funding acquisition: Z.W. (Zhenpo Wang); theoretical derivation, simulation, original draft preparation: Y.Z.; review and editing, resources, supervision: Z.W. (Zhifu Wang). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China under grant number 2021YFB2500900.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, M. Bus versus Rail: Meta-Analysis of Cost Characteristics, Carrying Capacities, and Land Use Impacts. *Transp. Res. Rec.* **2009**, *2110*, 87–95. [CrossRef]
2. Vuchic, V.R. *Urban Transit Systems and Technology*; Wiley: Hoboken, NJ, USA, 2007.

3. Tsai, Y.L. A Decision Support System for the Choice of Medium-Capacity Transit System. Master's Thesis, Nanyang Technological University, Singapore, 2010.
4. He, X. Application and Prospect of Straddle Monorail Transit System in China. *Urban Rail Transit* **2015**, *1*, 26–34. [CrossRef]
5. Wirasinghe, S.C.; Kattan, L.; Rahman, M.M.; Hubbell, J.; Thilakaratne, R.; Anowar, S. Bus Rapid Transit—A Review. *Int. J. Urban Sci.* **2013**, *17*, 1–31. [CrossRef]
6. Ingvardson, J.B.; Nielsen, O.A. Effects of New Bus and Rail Rapid Transit Systems—An International Review. *Transp. Rev.* **2018**, *38*, 96–116. [CrossRef]
7. Novales, M.; Orro, A.; Conles, E.; Anta, J. Medium-Capacity Transit Systems: Some Reflections about Making the Right Choice. In *WIT Transactions on State of the Art in Science and Engineering*; Pratelli, A., Ed.; WIT Press: Billerica, MA, USA, 2014; Volume 1, pp. 85–96.
8. Leng, H.; Ren, L.; Ji, Y. Path-Following Control Strategy for Gantry Virtual Track Train Based on Distributed Virtual Driving Model. *Veh. Syst. Dyn.* **2023**, *62*, 85–113. [CrossRef]
9. Wagner, S.; Zipser, S.; Bartholomaeus, R.; Baeker, B. A Novel Two DOF Control for Train-Like Guidance of Multiple Articulated Vehicles. In *Volume 4: 7th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B and C, Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, San Diego, CA, USA, 30 August–2 September 2009*; ASMEDC: San Diego, CA, USA, 2009; pp. 1989–1997.
10. Wagner, S.; Nitzsche, G.; Huber, R. Advanced Automatic Steering Systems for Multiple Articulated Road Vehicles. In *Volume 13: Transportation Systems, Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition, San Diego, CA, USA, 15–21 November 2013*; American Society of Mechanical Engineers: San Diego, CA, USA, 2013; p. V013T14A003.
11. Wagner, S.; Nitzsche, G. Advanced Steer-by-Wire System for Worlds Longest Busses. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; IEEE: Rio de Janeiro, Brazil, 2016; pp. 1932–1938.
12. Zeng, H.; Che, C.; Xie, J.; Jia, S. Research on Optimized Model and Simulation of Virtual Orbit Tramcar Tracking Control. *Comput. Simul.* **2021**, *38*, 89–92+492.
13. Peng, J.; Feng, J.; Xiao, L.; Ren, X.; Wang, S.; Liu, X. Research on Autonomous Guidance and Track Following Technology of Autonomous-rail Rapid Tram. *Control. Inf. Technol.* **2020**, *1*, 27–31.
14. Xiao, L.; Wang, K.; Zhou, S.; Ma, S. An Intelligent Multiple-Articulated Rubber-Tired Vehicle Based on Automatic Steering and Trajectory Following Method. *J. Vib. Control.* **2022**, *28*, 507–519. [CrossRef]
15. Zhang, D.; Yang, C.; Zhang, W.; Cheng, Y. A Novel Tracking Control Method for the Distributed-Drive and Active-Steering Articulated Virtual Rail Train. *Proc. Inst. Mech. Eng. Part I J. Syst. Control. Eng.* **2022**, *236*, 418–440. [CrossRef]
16. Zhang, D.; Yang, C.; Zhang, W. A New Active-Steering Control System for the Running Control of the Articulated Virtual Rail Train. In *Advances in Dynamics of Vehicles on Roads and Tracks II*; Orlova, A., Cole, D., Eds.; Lecture Notes in Mechanical Engineering; Springer International Publishing: Cham, Switzerland, 2022; pp. 802–814.
17. Zhang, Z.; Yang, C.; Zhang, W. An Adaptive Robust Path-Tracking Control Algorithm of Multi-Articulated and Redundantly Actuated Virtual Track Train. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2023**. [CrossRef]
18. Huber, R.; Jaekel, J.; Wagner, S.; Zipser, S. Track Estimation for Multi-Axle Steered Truck-Trailer Vehicles. In Proceedings of the 12th Mechatronics Forum Biennial International Conference, Zurich, Switzerland, 28–30 June 2010; pp. 294–301.
19. Zhang, R.; Shen, G.; Wang, W. Research on the Path-Following Control of Rubber-Tire Trackless Train with Bogies. *J. Mech. Sci. Technol.* **2022**, *36*, 1113–1121. [CrossRef]
20. Wang, Z.; Lu, Z.; Wei, J.; Qiu, X. Research on Virtual Track Train Path-Tracking Control Based on Improved MPC and Hierarchical Framework: A Reconfigurable Approach. *Appl. Sci.* **2023**, *13*, 8443. [CrossRef]
21. Leng, H.; Ren, L.; Ji, Y. Cascade Modular Path Following Control Strategy for Gantry Virtual Track Train: Time-Delay Stability and Forward Predictive Model. *IEEE Trans. Veh. Technol.* **2022**, *71*, 6969–6983. [CrossRef]
22. Leng, H.; Ren, L.; Ji, Y. Analysis Methodology of Compatibility between Motion Control and Mechanical Architecture of a Newly Designed Gantry Virtual Track Train and the Path-Tracking Control Strategy. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2022**, *236*, 6985–7005. [CrossRef]
23. Moon, K.-H.; Lee, S.-H.; Chang, S.; Mok, J.-K.; Park, T.-W. Method for Control of Steering Angles for Articulated Vehicles Using Virtual Rigid Axles. *Int. J. Automot. Technol.* **2009**, *10*, 441–449. [CrossRef]
24. Lee, S.H.; Park, T.W.; Moon, K.H.; Chung, K.H.; Kim, K.J. A Study on the Manoeuvrability Analysis of the Bi-Modal Tram Using the All-Wheel Steering Electronic Control Unit. *Veh. Syst. Dyn.* **2010**, *48*, 113–131. [CrossRef]
25. Jujnovich, B.A.; Cebon, D. Path-Following Steering Control for Articulated Vehicles. *J. Dyn. Syst. Meas. Control.* **2013**, *135*, 031006. [CrossRef]
26. Oreh, S.T.; Kazemi, R.; Azadi, S. A New Desired Articulation Angle for Directional Control of Articulated Vehicles. *Proc. Inst. Mech. Eng. Part K J. Multi-Body Dyn.* **2012**, *226*, 298–314. [CrossRef]
27. Ritzen, P.; Roebroek, E.; Van De Wouw, N.; Jiang, Z.-P.; Nijmeijer, H. Trailer Steering Control of a Tractor–Trailer Robot. *IEEE Trans. Contr. Syst. Technol.* **2016**, *24*, 1240–1252. [CrossRef]
28. Barfoot Timothy, D. *State Estimation for Robotics*; Cambridge University Press: Cambridge, UK, 2024.
29. Michalek, M.M.; Patkowski, B.; Gawron, T. Modular Kinematic Modelling of Articulated Buses. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8381–8394. [CrossRef]

30. Hiraoka, T.; Nishihara, O.; Kumamoto, H. Automatic Path-Tracking Controller of a Four-Wheel Steering Vehicle. *Veh. Syst. Dyn.* **2009**, *47*, 1205–1227. [CrossRef]
31. Kim, Y.C.; Yun, K.H.; Min, K.D. Automatic guidance control of an articulated all-wheel-steered vehicle. *Veh. Syst. Dyn.* **2014**, *52*, 456–474. [CrossRef]
32. Han, L.; Ren, L.; Ji, Y. Transverse statically indeterminate stress virtual rail train subsection coordinated steering tracking control method. Chinese Patent 115048714A, 13 September 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.