*energies*

MDPI

*Article*

# Model Predictive Control Based Path Tracking and Velocity Control with Rollover Prevention Function for Autonomous Electric Road Sweeper

**Yonghwan Jeong [1], Wongun Kim [2,*] and Seongjin Yim [1,*]**

[1] Department of Mechanical and Automotive Engineering, Seoul National University of Science and Technology, 232 Gongneung-ro, Nowon-gu, Seoul 01811, Korea; yh.jeong@seoultech.ac.kr
[2] Korea Institute of Industrial Technologies, Seobuk-gu, Cheonan-si 31056, Korea
* Correspondence: wgk@kitech.re.kr (W.K.); acebtif@seoultech.ac.kr (S.Y.); Tel.: +82-2-970-9011 (S.Y.)

**Abstract:** This paper presents a model predictive control (MPC)-based algorithm for rollover prevention of an autonomous electric road sweeper (AERS). For AERS, the basic function of autonomous driving is a path- and velocity-tracking control needed to make a vehicle follow given path and velocity profiles. On the other, the AERS adopts an articulated frame steering (AFS) mechanism which can make cornering behavior agile. Moreover, the tread of the AERS is narrow, and the height of the mass center is high. As a result, it is prone to roll over. For this reason, it is necessary to design a controller for path and velocity tracking and rollover prevention in order to improve maneuverability and roll safety of the AERS. A kinematic model was adopted as a vehicle one for the AERS. With the vehicle model, reference states of position and velocity were determined that are needed to make the AERS track the reference path and prevent rollover. With the vehicle model and reference states, an MPC-based motion controller was designed to optimize articulation angle and velocity commands. The load-transfer ratio (LTR) was used to measure a rollover propensity. To evaluate the proposed algorithm, a simulation was conducted for the U-turn scenario. Simulation results show that the proposed algorithm improves path tracking and prevents the rollover of the AERS.

**Keywords:** autonomous electric road sweeper; path-tracking control; velocity-tracking control; rollover prevention; model predictive control

check for updates

**Citation:** Jeong, Y.; Kim, W.; Yim, S. Model Predictive Control Based Path Tracking and Velocity Control with Rollover Prevention Function for Autonomous Electric Road Sweeper. *Energies* **2022**, *15*, 984. https://doi.org/10.3390/en15030984

Academic Editor: Daniel Chindamo

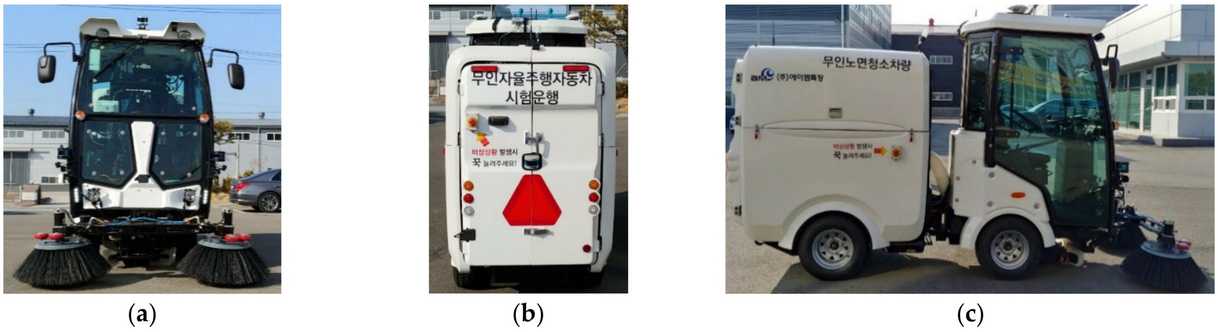
Received: 30 December 2021
Accepted: 27 January 2022
Published: 28 January 2022

## 1. Introduction

A road sweeper is a machine to clean roadsides in urban areas. The first ones appeared in the 17th century to replace human workers. The early road sweeper was drawn by horse and raked up the debris from the road. With the evolution of technologies, internal combustion engines and electric motors have been introduced as a power source for the road sweeper. The modern road sweeper appeared in the 1950s, and the structure has been maintained so far [1]. Recently, the road sweeper has become capable of cleaning small particles up to PM 1.0 or PM 2.5 [2]. Various types of sweeping machines were summarized in SAE Standard J2130 [3]. This standard classified the types of sweeping machines based on the cleaning method and the target environment. The brief history of the road sweeper can be found in Reference [4].

Research trends on road sweepers are expanding from cleaning performance to eco-friendly and autonomous driving. To enhance the sustainability of road transportation, the electrification of roads has been focused as a solution [5–7]. In the transportation sector, 82% of greenhouse gas emissions are from vehicles [8]. Generally, commercial vehicles, including road sweepers, are powered by diesel engines and have a long operating time. The amount of pollutants, compared to the number of vehicles, is higher for commercial vehicles than that of passenger ones [8]. Thus, the power source of the road sweeper has been changed from a diesel engine to electric motor with battery [9]. In addition to electrification, autonomous driving has been applied to road sweepers. The automation of road sweepers began with cleaning robots for the indoor environment [10]. By virtue of the advance of autonomous driving for passenger vehicles, various sensors have been added to the road sweeper for operations without human drivers [11–13]. The autonomous road sweeper can maximize its utilization because there are no operators. In addition, by replacing the work of sanitation workers, the injuries caused by accidents can be drastically reduced. From a functional point of view, the structure of autonomous road sweepers is more compact and efficient, due to the elimination of the cockpit. In this paper, this type of cleaning vehicle is named an autonomous electric road sweeper (AERS).

With these changes, various types of AERS have been developed to date. The first type of AERS was developed based on a truck platform by Delevo International, which is powered by a diesel engine [14]. Westfield POD and a Johnston Sweeper co-developed the AERS, named AutoSweeper, based on an electric vehicle. This AERS was designed for the purposed of cleaning airfields, not for public roads. [15]. The AERSs for public roads have been published by ENWAY, Trombia, and Boschung [16–18]. The target AERS of this paper is a small-sized road sweeper, which is similar to the Boschung's AERS. Figure 1 shows the AERS that has been in development by the Korea Institute of Industrial Technologies (KIIT) and AM Special Vehicle (AMSV) since 2019 [19]. The AERS is equipped with chassis sensors, IMU, and DGPS to measure the state of the vehicle. In addition, the actuators are also designed as x-by wire systems to implement the autonomous driving system.



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

**Figure 1.** Autonomous electric road sweeper considered in this paper: (**a**) front view, (**b**) rear view, and (**c**) side view.

A steering mechanism of the target AERS is an articulated frame steering (AFS), which has been widely used on construction, mining, and forestry vehicles [20,21]. Figure 2 shows the structure of the AFS. As shown in Figure 2, the AERS is composed of the front and rear bodies. Each body can have a different dynamic state. A heading difference of two bodies is defined as an articulation angle, $\gamma$. The articulation angle serves as a steering angle of a front-wheel-steered (FWS) vehicle. To change the articulation angle, a hydraulic cylinder is attached between the front and rear bodies. For longitudinal motion, a single driving motor is mounted on the rear axle to drive the rear wheels. A pneumatic brake is used to stop the vehicle. These actuators are developed for drivers driving and autonomous driving systems. In this study, the AERS was operated by an autonomous driving system.

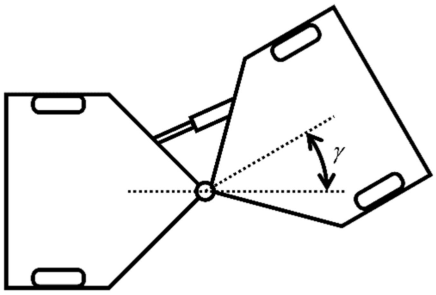


**Figure 2.** Structure of articulated frame steering.

The AERS of this paper has four special features. First, this vehicle shows enhanced maneuverability and agility, which generate larger yaw motion than FWS vehicles with an identical steering and articulation angle. Thus, the AFS provides a smaller turning diameter than FWS vehicles [20,21]. In other words, this means that a small articulation angle can generate a larger lateral acceleration than the same steering angle of an FWS vehicle. Second, this vehicle is operated on specific courses under low-speed conditions. The maximum and operation speeds of this vehicle are 20 and 5 km/h, respectively. Therefore, there are a few issues about lateral stability and snaking phenomenon, as outlined by Ref. [21]. Third, the height of the mass center is larger than the tread width. Finally, this vehicle has no suspensions. These features make this vehicle more prone to roll over. For this reason, it is necessary to design a controller for rollover prevention. Since the AERS should follow a predefined path, the steering controller with AFS cannot be used as a method for rollover prevention. Therefore, the speed controller or velocity-tracking controller developed for autonomous driving is available for roll prevention.

To implement autonomous driving on the AERS, the perception, localization, decision, motion planning, and vehicle control for typical passenger vehicles should be modified [22–25]. It is important to consider the Operational Design Domain (ODD) and dynamic characteristics of the AERS. The ODD of the AERS is reduced, compared to autonomous driving of the passenger cars. Thus, the perception, decision, and planning algorithms can be implemented on the AERS by considering fewer variables than those of road vehicles [26]. However, the dynamic behavior of AFS is quietly different from FWS vehicles [21]. Therefore, a vehicle-controller design is an important issue for the automation of the road sweeper. Therefore, this study aimed to design a control algorithm that considers the characteristics of AERS.

Many studies have been conducted on tractor-trailer structures capable of front-wheel steering in the design of route tracking controllers for articulated vehicles. A reversing control algorithm for the twin trailer was proposed based on a minimum set-path control. This controller is designed with a linear optimization to reduce the control effort [27]. A backward path-tracking controller was also proposed, using linearization of the vehicle model for tractor-trailers [28]. An algorithm for articulated vehicle stability was proposed that determines the target yaw moment with linear Model Predictive Control (MPC) and distributes control effort to brake actuators through another optimization [29]. In contrast to tractor-trailers, the AERS can control the articulation angle and has non-steerable wheels [30–32]. Therefore, it is difficult to utilize a controller based on FWS. Various approaches have been introduced to design a controller for the AERS with AFS. For path tracking, a kinematic model has been used to design an error dynamics-based state feedback controller [33] and an MPC-based one [34,35]. To improve the robustness, an adaptive MPC was used to consider disturbances in the path curvature [36]. A two-level tracking controller was proposed based on a kinematic-model-based MPC and a dynamic-model-based Sliding Mode Control (SMC). MPC was used to determine the posture of the articulated vehicle, and SMC was used to calculate the control input to track the desired posture. [37]. In addition, nonlinear control approaches have also been introduced to design path-tracking controllers [38–40]. A concept of potential function was utilized to generate the virtual altitude function of the terrain field to determine the steering angle [41]. However, these studies focused on the articulation angle control under the assumption that vehicle speed is constant, or there were cases where only the maximum speed limit was given [36]. In addition, the risk of rollover due to AFS is also not considered. Therefore, velocity-tracking control is needed for rollover prevention, because the steering control with AFS cannot be used for the purpose. This is because the AERS should follow the reference path to avoid collision with surrounding objects. In other words, reducing the articulation angle to prevent the rollover can cause other problems. Thus, it is necessary to design an integrated controller in order to make the AERS follow the given path and velocity profiles and prevent rollover simultaneously.

Figure 3 shows the structure of the proposed integrated control algorithm. As shown in Figure 3, the proposed algorithm uses the path information and vehicle states from the AERS model to determine control inputs. A reference-state decision module adopts a kinematic model to determine the reference position, heading, and articulation angle. In this process, a threshold for lateral acceleration is adopted for rollover prevention. An MPC-based motion controller is designed to track the reference states and satisfy constraints, including the lateral acceleration for rollover prevention. For actuator control, a low-level controller generates actuator inputs from outputs of the MPC controller. Simulation is conducted based on MATLAB/Simulink environments to show the effectiveness of the proposed algorithm.

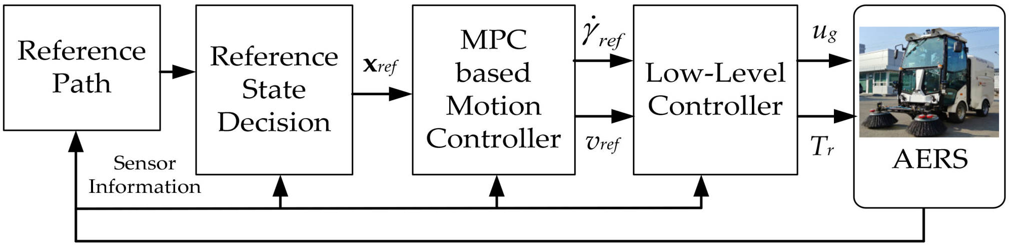


**Figure 3.** Schematic diagram of the control algorithm for AERS.

The key contributions of this paper are summarized as follows:

1. In the proposed algorithm, the MPC-based path and velocity-tracking controllers determine the control inputs, i.e., the articulation angle and rear-wheel torque, required for the AERS with AFS mechanism.
2. The reference states can be determined regardless of the road shapes and vehicle models. Thus, the reference-state decision module can be used for various types of road conditions and vehicle systems.
3. The desired velocity in the reference states is derived for rollover prevention from a predefined lateral acceleration threshold. With the desired velocity, an MPC-based velocity-tracking controller can effectively prevent the AERS from rolling over.

The remainder of the paper is organized as follows: In Section 2, the vehicle model for the AERS is built and the state equation is derived. The path and velocity-tracking control algorithms are designed in Section 3. Section 3 is composed of three subsections: reference-state decision, MPC-based controller, and low-level controller. In Section 4, we recount a simulation that was conducted to evaluate the performance of the proposed algorithm. The last section, Section 5, provides the conclusion of this paper.

## 2. Vehicle Modeling

For the articulated vehicle shown in Figure 2, it is assumed that each body has a non-steerable single wheel. Figure 4 shows the geometry of the articulated vehicle model [33–35,38]. As mentioned before, the AERS is composed of the front and rear bodies. The steering of the AERS is performed by the actuation of the hydraulic cylinder on the center joint. The hydraulic cylinder changes the articulation angle, $\gamma$. $L_f$ and $L_r$ are the distances from the center joint to the front and rear wheels, respectively. In this paper, the wheels on both sides of the axle were simplified as a single wheel, which is similar to the bicycle model that is widely used in designing path-tracking controller [33,34,38]. Positions of the front and rear wheels are denoted as $P(x_f, y_f)$ and $P(x_r, y_r)$, respectively. $\theta_f$ and $\theta_r$ are the heading angles of the front and rear bodies with respect to the global coordinate system, respectively. The velocities of the front and rear bodies can be represented by $v_f$ and $v_r$ under the no-slip assumption.

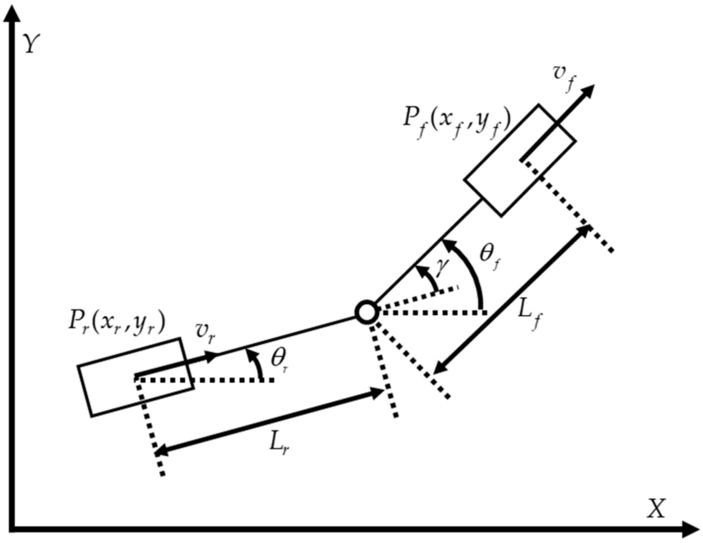


**Figure 4.** Kinematic model for the articulated vehicle.

Since the AERS operates under low-speed conditions, it is reasonable to model the behavior by using a kinematic model with an assumption of the no-slip [33,34]. Therefore, the changes of $v_f$ and $v_r$ are identical with respect to the rigid center joint. Thus, constraints on $v_f$ and $v_r$ can be applied to the positions, as given in (1).

$$\begin{cases} \dot{x}_f \sin\theta_f - \dot{y}_f \cos\theta_f = 0 \\ \dot{x}_r \sin\theta_r - \dot{y}_r \cos\theta_r = 0 \end{cases} \tag{1}$$

The kinematic model of the AERS is derived under three assumptions. First, the AERS only moves on the 2D plane with a small body slip. Second, the dynamic characteristics of the tire are negligible. Third, the articulation angle remains constant while traveling a small distance. Under these assumptions, the changes of $P(x_f, y_f)$ are represented as (2).

$$\begin{cases} \dot{x}_f = v_f \cos\theta_f \\ \dot{y}_f = v_f \sin\theta_f \end{cases} \tag{2}$$

Since the front and rear bodies are connected via the rigid center joint, the change rate of $v_f$ and $v_r$ can be regarded as identical. Thus, the relationship between $v_f$ and $v_r$ is derived as (3). By substituting $v_r$ to $v_f$ in (3), the angular velocity of the front body is calculated as follows (4):

$$\begin{cases} v_f = v_r \cos\gamma + \dot{\theta}_r L_r \sin\gamma \\ v_r = \dot{\theta}_f L_f \sin^{-1}\gamma + \dot{\theta}_r L_r \tan^{-1}\gamma \end{cases} \tag{3}$$

$$\dot{\theta}_f = \frac{v_f \sin\gamma + L_r \dot{\gamma}}{L_f \cos\gamma + L_r} \tag{4}$$

The vectors of the state and input are defined as (5) and (6), respectively. In the input vector (6), $v_{des}$ and $\dot{\gamma}_{des}$ are the velocity and the articulation angular rate inputs, respectively. With the definitions of the state and input vectors, the state equation of the kinematic model for the front body is obtained as (7) from (2) and (4) [34].

$$\mathbf{x} = \begin{bmatrix} x_f & y_f & \theta_f & \gamma \end{bmatrix}^T \tag{5}$$

$$\mathbf{u} = \begin{bmatrix} v_{des} & \dot{\gamma}_{des} \end{bmatrix}^T \tag{6}$$

$$\begin{bmatrix} \dot{x}_f \\ \dot{y}_f \\ \dot{\theta}_f \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} \cos\theta_f & 0 \\ \sin\theta_f & 0 \\ \frac{\sin\gamma}{L_f\cos\gamma+L_r} & \frac{L_r}{L_f\cos\gamma+L_r} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{des} \\ \dot{\gamma}_{des} \end{bmatrix} \tag{7}$$

Based on the state variables of the front body and the articulation angle, the state variables of the rear body are obtained as (8) [34]. This kinematic vehicle model was utilized for the reference-state decision module and the design of MPC-based motion controller.

$$\begin{cases} x_r = x_f - L_f\cos\theta_f - L_r\cos\theta_r \\ y_r = y_f - L_f\sin\theta_f - L_r\sin\theta_r \\ \theta_r = \theta_f - \gamma \end{cases} \tag{8}$$

## 3. Controller Design

As shown in Figure 3, the proposed algorithm for path- and velocity-tracking control is composed of the three modules: reference-state decision, MPC-based controller, and low-level controller. The reference-state decision module provides the reference states, which are tracked by the MPC-based controller. The MPC-based controller determines the desired articulation angular rate and velocity needed to make the AERS follow the reference states. The low-level controller determines the actuator inputs to follow the desired articulation angular rate and velocity. The details of the proposed algorithm are discussed in the following subsection.

### *3.1. Reference-State Decision Module*

The reference states for the MPC are determined at each sampling time of the controller. The reference-state decision module uses the relative position between the reference path and the AERS. Thus, the global coordinates of the reference path are transformed to local ones by using the position and heading of the AERS. The proposed algorithm is different from conventional path-tracking one in such a way that the representative path-tracking methods, such as Stanley, are based on lateral and heading errors [42]. Furthermore, the error-dynamics-based approaches require the assumption on the geometry of the reference path to define a time derivative of the errors [42]. To overcome the limitations of path-tracking error-based approaches, the reference-state decision and the reference tracking controller are designed separately. As a result, it is possible to use the reference path without any assumptions. Moreover, the longitudinal motion of the AERS is considered simultaneously when determining the reference states. Therefore, the reference state for the lateral and longitudinal motion can be simultaneously determined.

The components of the reference states are identical with the state vector (5). In other words, the longitudinal and lateral positions, heading, and articulation angular rate are determined based on the geometric relationship between the AERS and the reference path [43]. Figure 5 shows a desired path decision procedure based on the reference path. In this study, the desired path is the path to converge into the reference path. The reference path is the path that AERS should follow. In Figure 5, the reference path is marked with a black dotted line. The desired path is defined as the path connecting the center of the front axle, $P(x_0, y_0)$, and the target point, $P(x_1, y_1)$, which is marked as a red solid line in Figure 5. To define the target point, the preview point is determined by calculating a preview distance, $L_p$, which is proportional to the velocity of the vehicle with gain, $k$. The constant $k$ was used because the ODD is limited to low-speed conditions. The closest point on the reference path to the preview point is defined as the target point, $P(x_1, y_1)$.

The desired path is used to calculate the required lateral motion needed to align the AERS to the reference path. The desired path is modeled as a 2nd order polynomial and defined as (9) in the local coordinate of the front body, because a path-generation algorithm for autonomous vehicles is generally defined with respect to the front of the vehicle, where the sensor is attached. The coefficient of the $y(x)$ is determined by (10) from the position and heading constraints at $P(x_0, y_0)$ and the position constraint at $P(x_1, y_1)$ [43]. A curvature, $\kappa_{des}$, at $P(x_0, y_0)$ is calculated from the desired path as (11). With the definition of the curvature, the desired yaw rate $\dot{\theta}_{des}$ is obtained as (12). In (12), $v_{des}$ is the desired velocity to prevent rollover of the AERS. As mentioned in the introduction, the maximum speed of the AERS is 20 km/h. Therefore, the desired yaw rate can be determined directly from the curvature of the desired path. Equation (12) is the explicit relationship between the desired yaw rate and the reference path, which is the key contribution of this paper. By virtue of the relationship, the desired yaw-rate tracking and the reference-path tracking are connected to each other.

$$y(x) = a_2x^2 + a_1x + a_0 \tag{9}$$

$$\begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} {x_0}^2 & x_0 & 1 \\ {x_1}^2 & x_1 & 1 \\ 2x_0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y'(x_0) \end{bmatrix} \tag{10}$$

$$\kappa_{des} = \frac{y''(x_0)}{\left\{1 + y'(x_0)^2\right\}^{1.5}} \tag{11}$$

$$\dot{\theta}_{des} = k_{des} \cdot v_{des} \tag{12}$$

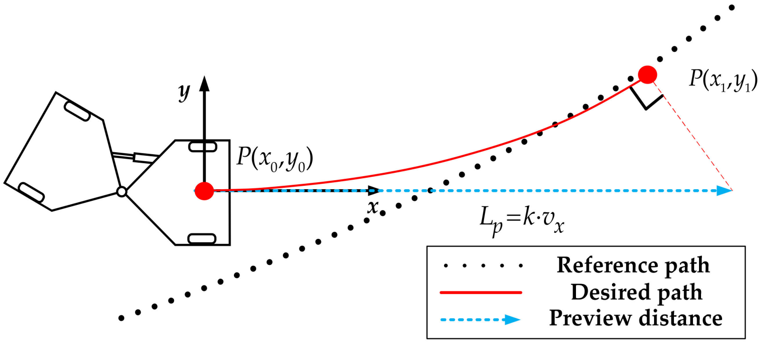


**Figure 5.** Desired path decision based on reference path.

As mentioned above, the coordinate system for the path generation and tracking algorithm was matched. Therefore, if the yaw rate of the front body is equal to the desired one, the AERS follows the reference path. Based on (4), the desired articulation angular rate $\dot{\gamma}_{des}$ is defined as (13).

$$\dot{\gamma}_{des} = \frac{1}{L_r}\left\{\dot{\theta}_{des}\left(L_f \cos\gamma + L_r\right) - v_f \sin\gamma\right\} \tag{13}$$

Generally, the speed of the AERS is set as constant. However, due to the AFS mechanism, the AERS is prone to roll over under low speed if a large articulation angle is applied. For this reason, a rollover prevention function is needed for the AERS. Lateral acceleration is known to be the most important factor in vehicle rollover [44]. If the lateral acceleration is maintained as to be as low as possible, the vehicle rollover cannot occur. Thus, it is necessary to limit the speed of the AERS by the lateral acceleration threshold for rollover prevention.

For longitudinal motion, the desired velocity needed for rollover prevention is derived from the definition of lateral acceleration [45]. The relationship between the lateral acceleration and the longitudinal velocity is given as (14). In (14), $v_{xi}$, $v_{yi}$, and $\theta_i$ are the longitudinal and lateral velocities, and the heading angle of the front and rear bodies, respectively. Moreover, $a_{yi,measured}$ is the measured lateral acceleration of the front and rear bodies. From (14), the relationship between the desired acceleration, $a_{yi,des}$, and the desired velocity, $v_{des}$, is obtained as (15). In (15), $a_{yi,des}$ is a parameter that is set by a controller designer considering the rollover prevention function of the velocity tracking controller. Thus, the desired velocity $v_{xi,des}$ needed for rollover prevention is derived as (16) from (14) and (15) [42]. If the threshold $a_{yi,des}$ is set to higher values, then the desired velocity will be maintained as a constant value, the maximum velocity. On the contrary, if $a_{yi,des}$ is set to lower values, then the desired velocity will decrease in order to prevent rollover. The desired velocity $v_{xi,des}$ is used in calculating the desired yaw rate, as given in (12).

$$\dot{v}_{yi} = a_{yi,measured} - v_{xi}\dot{\theta}_i, \quad i = f, r \tag{14}$$

$$a_{yi,des} = \dot{v}_{yi} + v_{xi,des}\dot{\theta}_i, \quad i = f, r \tag{15}$$

$$v_{xi,des} = \frac{1}{\dot{\theta}_i}\left\{a_{yi,des} - \left(a_{yi,measrued} - v_{xi}\dot{\theta}_i\right)\right\} \tag{16}$$

From (16), two desired velocities are obtained based on the dynamic states and the desired lateral acceleration of the front and rear bodies. A vehicle model for controller design is the kinematic model of (7), but the vehicle model used in the simulation is composed of the 7-DOF model as given in [21]. Thus, the signals needed to calculate the desired velocity can be measured. The proposed algorithm can set the limit of the lateral acceleration for each body, i.e., $a_{yi,des}$, separately considering the parameter and payload. Since the AERS has similar parameters for each body, the $a_{yi,des}$ for each body is set into an identical value. After calculating $v_{xi,des}$, the minimum between $v_{xf,des}$ and $v_{xr,des}$ is used as $v_{des}$.

The reference states for a prediction horizon are determined by integrating (7), using $\dot{\gamma}_{des}$ and $v_{des}$. The reference-state decision at $k$-th step is summarized as (17) and (18). In (18), $\Delta T$ is the sampling time of the proposed MPC controller, and $k$ is a discrete-time index.

$$\begin{bmatrix} \dot{x}_f(k) \\ \dot{y}_f(k) \\ \dot{\theta}_f(k) \\ \dot{\gamma}(k) \end{bmatrix} = \begin{bmatrix} \cos\theta_f(k) & 0 \\ \sin\theta_f(k) & 0 \\ \frac{\sin\gamma(k)}{L_f\cos\gamma(k)+L_r} & \frac{L_r}{L_f\cos\gamma(k)+L_r} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{des}(k) \\ \dot{\gamma}_{des}(k) \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} x_f(k) \\ y_f(k) \\ \theta_f(k) \\ \gamma(k) \end{bmatrix} = \begin{bmatrix} x_f(k-1) \\ y_f(k-1) \\ \theta_f(k-1) \\ \gamma(k-1) \end{bmatrix} + \begin{bmatrix} \dot{x}_f(k) \\ \dot{y}_f(k) \\ \dot{\theta}_f(k) \\ \dot{\gamma}(k) \end{bmatrix} \Delta T \tag{18}$$

### *3.2. MPC-Based Motion Controller*

The state Equation (7) is nonlinear, which increases a computational burden when directly applied to MPC. Generally, a nonlinear MPC (NMPC) utilizes a nonlinear optimization in order to find optimal inputs, considering a nonlinear model, cost function, and constraints. If the target system shows a nonlinear behavior, NMPC should be used to determine appropriate inputs even if a large amount of computation is required [46,47]. However, as mentioned before, the ODD of the ARES is limited to low-speed conditions. Thus, the behavior of the ARES can be modeled by a linear-time-varying model obtained by local linearization [44,45]. Based on the state and input vectors, as given in (5) and (6), the nonlinear-state equation is locally linearized as (19) with respect to the current states and the previous inputs. In (19), $f(\mathbf{x}, \mathbf{u})$ represents the nonlinear state equation of (7). In this paper, it is assumed that all states are observable, so the output equation can be omitted. In (19), the system matrix $\mathbf{A}(t)$ and the input matrix $\mathbf{B}(t)$ are derived as (20) and (21), respectively.

$$\begin{aligned} \dot{\mathbf{x}}\,(t) &= \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \\ &= \left(\left.\frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\mathbf{x}(t),\mathbf{u}(t-1)}\right)\mathbf{x}(t) + \left(\left.\frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{x}(t),\mathbf{u}(t-1)}\right)\mathbf{u}(t) \end{aligned} \tag{19}$$

$$\mathbf{A}(t) = \begin{bmatrix} 0 & 0 & a_{13} & 0 \\ 0 & 0 & a_{23} & 0 \\ 0 & 0 & 0 & a_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$a_{13} = -v_{des}\sin\theta_f,\ a_{23} = v_{des}\cos\theta_f$$
$$a_{34} = \frac{v_{des}\cos\gamma}{L_f\cos\gamma + L_r} + \frac{L_f\sin\gamma(v_{des}\sin\gamma + L_r\dot{\gamma})}{(L_f\cos\gamma + L_r)^2} \tag{20}$$

$$\mathbf{B}(t) = \begin{bmatrix} \cos\theta_f & 0 \\ \sin\theta_f & 0 \\ \frac{\sin\gamma}{L_f\cos\gamma + L_r} & \frac{L_r}{L_f\cos\gamma + L_r} \\ 0 & 1 \end{bmatrix} \tag{21}$$

The time-varying linear system in (19) is discretized as (22) by the first-order difference in implementing a digital controller [48].

$$\mathbf{x}(k+1) = \mathbf{A}_d(k)\mathbf{x}(k) + \mathbf{B}_d(k)\mathbf{u}(k)$$
$$\begin{cases} \mathbf{A}_d(k) = \mathbf{I} + \mathbf{A}(t)\Delta T \\ \mathbf{B}_d(k) = \mathbf{B}(t)\Delta T \end{cases} \tag{22}$$

The cost function is defined to minimize the reference state tracing error and the control effort. In other words, a reference state tracking error $\widetilde{\mathbf{x}}(k)$ is introduced as (23) to define the cost function [49,50].

$$\widetilde{\mathbf{x}}(k) = \mathbf{x}(k) - \mathbf{x}_{ref}(k) \tag{23}$$

Considering the lateral behavior of the AERS, the usage of the articulation angular rate should be minimized in order to avoid an unnecessary actuation. The control effort for the longitudinal motion is minimized by maintaining the vehicle speed as the set speed, $v_{set}$. Thus, the cost for the input vector is modified to consider the different nature of the lateral and longitudinal actuation. In this paper, a reference input error, $\widetilde{\mathbf{u}}\ (k)$, is defined as (24).

$$\widetilde{\mathbf{u}}\ (k) = \begin{bmatrix} v_{des} \\ \dot{\gamma}_{des} \end{bmatrix} - \begin{bmatrix} v_{set} \\ 0 \end{bmatrix} \tag{24}$$

Based on the definitions of $\widetilde{\mathbf{x}}\ (k)$ and $\widetilde{\mathbf{u}}\ (k)$, the cost function is defined as (25). In (25), $T_p$ is the prediction horizon of the proposed MPC problem. Moreover, $\rho$ and $\varepsilon$ are a weight factor and a slack variable for the constraint on longitudinal acceleration, respectively. Therefore, the slack variable was introduced into the constraint on longitudinal acceleration. The last term, $\rho\varepsilon$, in (25) is introduced to avoid a failure of the convergence of optimization [49]. The proposed MPC can find a suboptimal solution when an optimal solution does not exist within the prediction horizon. In other words, if the optimal solution cannot be obtained considering only the hard constraints, the slack variable serves to obtain the solution with minimal violation of the soft constraint. For this reason, $\rho\varepsilon$ improves the optimization performance of the MPC in various conditions. Since the goal of this study was to prevent the rollover of the AERS, the priority is to control lateral behavior. **Q** and **R** are defined as (26), and they are the diagonal weighting matrices for the reference tracking and control effort, respectively. In (26), $Q_x$, $Q_y$, and $Q_\theta$ are the weights on the tracking errors of the longitudinal and lateral positions, and the heading angle, respectively. $Q_\gamma$ is the weight on the tracking error of the articulation angle. Since the articulation angle is directly changed by the articulation angular rate, $Q_\gamma$ is set as a zero to avoid situations where input is considered twice in cost. In (26), $R_v$ and $R_{\dot{\gamma}}$ are the weights on the velocity and the articulation angular rate inputs, respectively.

$$J = \sum_{k=1}^{T_p/\Delta T}\left[\widetilde{\mathbf{x}}_{ref}(k)^T\mathbf{Q}\,\widetilde{\mathbf{x}}_{ref}(k) + \widetilde{\mathbf{u}}\ (k)^T\mathbf{R}\,\widetilde{\mathbf{u}}\ (k)\right] + \rho\varepsilon \tag{25}$$

$$\mathbf{Q} = \begin{bmatrix} Q_x & & & \\ & Q_y & & \\ & & Q_\theta & \\ & & & Q_\gamma \end{bmatrix}, \mathbf{R} = \begin{bmatrix} R_v & \\ & R_{\dot{\gamma}} \end{bmatrix} \quad (26)$$

The constraints are classified into three categories: state, dynamic, and input constraints. The current states of the AERS are used as an initial constraint to obey the dynamics of the plant model. This is given as (27). Since the proposed problem is defined in the local coordinate system of the front body, the longitudinal and lateral positions and the heading angle are set to zero. The fourth element of (27) is defined as the measured articulation angular rate from the sensor model of the simulation environment. A constraint one terminal state is removed to enlarge the stable domain for the given initial constraint [51].

$$\mathbf{x}(1) = \begin{bmatrix} 0 & 0 & 0 & \dot{\gamma}_{meas} \end{bmatrix}^T \quad (27)$$

For dynamic constraints, the discrete linear model of (22) is used as equality constraints, as given in (28). In (28), $N_p$ is the prediction step of the MPC problem, which is calculated from (29).

$$\begin{cases} \mathbf{x}(2) = \mathbf{A}_d(1)\mathbf{x}(1) + \mathbf{B}_d(1)\mathbf{u}(1) \\ \qquad\vdots \\ \mathbf{x}\left(N_p\right) = \mathbf{A}_d\left(N_p - 1\right)\mathbf{x}\left(N_p - 1\right) + \mathbf{B}_d\left(N_p - 1\right)\mathbf{u}\left(N_p - 1\right) \end{cases} \quad (28)$$

$$N_p = T_p/\Delta T \quad (29)$$

Inequality constraints for the state and input vectors are defined as (3). In addition, the constraint on the longitudinal acceleration is introduced. Since the proposed MPC is based on the discrete model, the inequality constraint on acceleration is defined as (31). In (31), $a_{x,\min}$ and $a_{x,\max}$ are the minimum and maximum accelerations, which are derived to provide the smooth operation of the AERS, respectively. The parameters of the AERS and the MPC-based controller are summarized in Table 1.

**Table 1.** Parameters for AERS and MPC-based controller.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $L_f$ | 0.8 m | $L_r$ | 1.0 m |
| $Q_x$, | 1 | $\Delta T$ | 0.1 s |
| $Q_y$ | 5 | $N_p$ | 20 |
| $Q_\theta$ | 40 | $\gamma_{\min}$ | −10 deg |
| $Q_\gamma$ | 0 | $\gamma_{\max}$ | 10 deg |
| $R_v$ | 1 | $v_{\min}$ | 0 m/s |
| $R_{\dot{\gamma}}$ | 10 | $v_{\max}$ | 5 m/s |
| $a_{x,\min}$ | $-1$ m/s$^2$ | $\dot{\gamma}_{min}$ | −30 deg/s |
| $a_{x,\max}$ | 1 m/s$^2$ | $\dot{\gamma}_{max}$ | 30 deg/s |

The proposed MPC-based controller ensures closed-loop stability. The proposed linear MPC uses a simple linear model that can be stabilized with bounded input. In addition, since the state variable was defined as an error from the reference state, the domain of the terminal state can be defined as a bounded region. Therefore, the proposed algorithm satisfies the condition of the asymptotic stable of the linear MPC [52].

### *3.3. Low-Level Controller*

For the path- and velocity-tracking control, steering and driving/braking are needed. In the AERS, the steering is performed by changing the articulation angle with the moment generated by the hydraulic cylinder at the center joint. Given a reference articulation angle or angular rate, the steering moment should be determined for the AERS to follow the reference. The traction and braking torques are generated from the electric motor connected to the rear axle. Given a reference velocity profile, the motor torque should be determined for the AERS to follow the reference. For this purpose, the low-level controller is designed.

The low-level controller is used to actuate the AERS. The actuator inputs, i.e., the steering moment and the tracking/braking torque, are needed to make $v_{x,f}$ and $\dot{\gamma}$ follow the $v_{des}$ and $\dot{\gamma}_{des}$ are determined by PID controllers. Since the AERS has a rigid center joint, the speed controller can only consider one of the bodies when defining a speed error. In this paper, the speed of the front body is used to design the PID controller for the required torque decision. The low-level controllers for steering and velocity tracking are designed as (32). In (32), $u_\gamma$ and $T_r$ are the steering moment applied at the center joint and the traction/braking torque, respectively. Since the desired states for the PID controller are bounded and defined by considering the current state of the AERS, the output of the PID controller is calculated without causing a sudden change in the behavior of the vehicle.

$$\begin{cases} u_\gamma = K_{ps}e_\gamma + K_{is}\int e_\gamma dt + K_{ds}\dot{e}_\gamma, & e_\gamma = \dot{\gamma}_{des} - \dot{\gamma} \\ T_r = K_{pv}e_v + K_{iv}\int e_v dt + K_{dv}\dot{e}_v, & e_v = v_{des} - v_f \end{cases} \tag{30}$$

## 4. Simulation Results

The simulation was conducted with the reference-state decision module and the MPC-based motion controller for the AERS in order to evaluate the performance of the proposed algorithm and compare it with base algorithms. The MPC-based motion controller in the proposed algorithm was implemented by using CVXGEN, which is a solver for a convex optimization problem [53].

### *4.1. Vehicle Model for Simulation*

The simulation environment was established on the MATLAB/Simulink with the nonlinear bicycle model of the AERS. Figure 6 shows the free-body diagram of the nonlinear bicycle model used for the simulation. This adopted the Dugoff tire model in calculating the longitudinal and lateral tire forces, which were used to derive the equation of motion for the front and rear bodies [21]. The steering moment generated by the hydraulic cylinder at the center joint was determined by (32), with the consideration of the torsional stiffness and damping. The traction/braking torque of the rear wheel obtained from (32) was applied through the wheel dynamics [21]. It should be emphasized that the vehicle model used for the MPC-based controller was different from that used for the simulation.

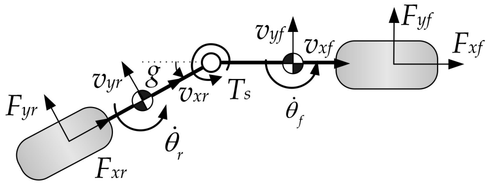
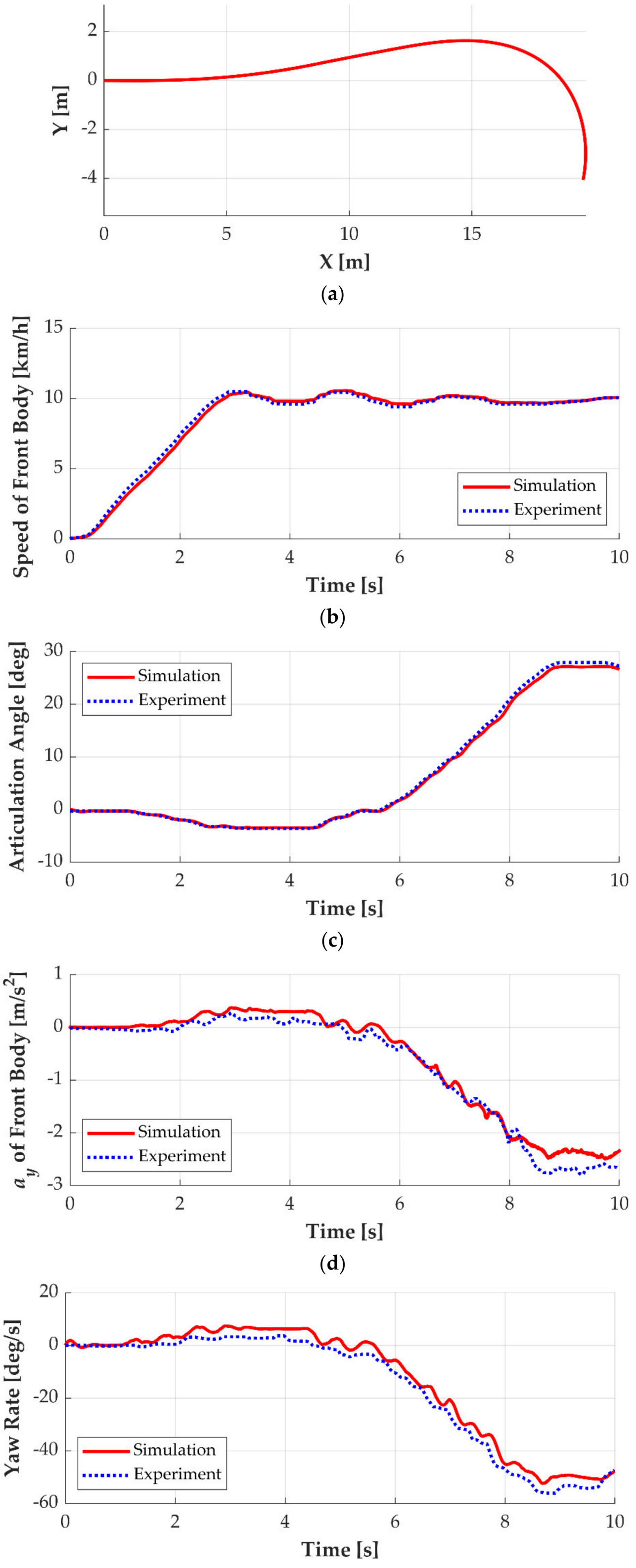


**Figure 6.** Nonlinear bicycle model used for simulation.

The experiment was conducted with the real AERS, as given in Figure 1, on a proving ground. With the experimental data, the nonlinear bicycle model was validated, and the performance of the low-level controller was checked. Figure 7 shows the comparison between simulation and experimental results. Figure 7a shows the trajectory of the experimental data. This trajectory is obtained by increasing the articulation angle to the maximum value until just before the rollover occurs. In this case, the lateral acceleration is about 2.7 m/s$^2$. Figure 7b,c show the tracking performance of the low-level controller. As shown in this figure, the low-level controller followed the reference articulation angle and speed commands obtained from the experiments well. As shown in Figure 7d,e, the simulation results of the nonlinear bicycle model were well fitted to the experimental data. Therefore, it can be concluded that the nonlinear bicycle model is valid enough to be used in the simulation with the proposed algorithm.



(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

**Figure 7.** Comparison between the simulation results of the nonlinear vehicle model and experimental results: (**a**) trajectory of the simulation result, (**b**) velocity tracking performance of the low-level controller, (**c**) articulation angle tracking performance of the low-level controller, (**d**) lateral accelerations of the front body from simulation and experiment, and (**e**) yaw rates of the front body from simulation and experiment.

### *4.2. Rollover Measure*

One of the objectives of this paper is to prevent a rollover through velocity planning. Among the indices to measure the rollover propensity of a vehicle, the load transfer ratio (LTR) was adopted as an index for the likelihood of rollover. LTR is the ratio of the load difference between the left and right wheels to the sum of all-wheel loads [54]. Since the AERS has two bodies, LTR is defined for the front and rear bodies, respectively. With the vertical force of each tire, LTRs of the front and rear bodies are calculated as (33). If the vertical force of the left or right wheels becomes zero, LTR is equal to 1. In this case, it can be regarded that the rollover occurs.

$$\mathrm{LTR}_i = \frac{|F_{zi,L} - F_{zi,R}|}{F_{zi,L} + F_{zi,R}}, \qquad i = f, r \tag{31}$$

### *4.3. Base Algorithms*

Three base algorithms were used to compare the performance of the path and velocity tracking and rollover prevention with the proposed algorithm. Pure pursuit and Stanley methods, which are frequently used as a path tracker for FWS vehicles, were adopted as the first and second base algorithms, respectively. The details of pure pursuit and the Stanley method were presented in Reference [42]. To consider the unique steering mechanism of the AERS, a model-free path tracker, which does not require vehicle parameters, was introduced as the third base algorithm [43]. The model-free path tracker was designed with the preview model and the simplified lateral dynamics. The simplified lateral dynamics assume the relationship between the yaw rate and the steering input as a first-order delay model. The lateral dynamics is defined as (32). In (32), $\dot{\theta}$ and $\delta$ are the yaw rate and steering angle, respectively. $\lambda$ and $\tau$ are the yaw rate gain and yaw rate delay, respectively. Equation (32) does not require the vehicle parameter. Thus, the model-free approach can be used regardless of the vehicle type if an appropriate adaptation law for $\dot{\theta}$ is adopted. The yaw rate error, $e_{\dot{\theta}}$, between the desired and real yaw rates was used to formulate the adaptation law for $\lambda$, as given in (33). In (33), $k$ is the gain parameter. With the estimate of $\lambda$, the steering angle is determined as (34). In (34), $\dot{\theta}_{des}$ is the desired yaw rate. In this paper, the desired yaw-rate decision method of the proposed algorithm in Section 3.1 was also adopted for the model-free path tracker.

$$\dot{\theta} = \frac{\lambda}{\tau s + 1}\delta \tag{32}$$

$$\hat{\lambda} = -k \cdot \delta \cdot e_{\dot{\theta}} \tag{33}$$

$$\delta = \frac{1}{\hat{\lambda}}\left(\dot{\theta}_{des} + \tau \cdot \ddot{\theta}_{des}\right) \tag{34}$$

For the longitudinal motion, the desired velocity for the base algorithms was defined by considering the curvature of the single point on the reference path. This point was used to determine the steering input. The pure pursuit and model-free steering controller used a look-ahead point in order to calculate the road curvature. The look-ahead point was selected from the reference path in the same way as when the proposed algorithm defined the desired path. The Stanley method uses a closed point on the reference path in order to calculate the curvature. With the curvature of the reference point, the same relationship between the lateral acceleration and the curvature in Section 3.1 was used to determine the desired velocity.

### *4.4. Simulation Results of the Proposed and Base Algorithms*

To evaluate the performance and effectiveness of the proposed algorithm, the closed-loop steering was performed on a U-shaped path with a radius of 4 m. The reference path was composed of straight line and arc segments, and the road curvature changed discontinuously at the junction point of two segments. The discontinuous change of the road curvature caused the sudden change of the control input, thus increasing the possibility of the rollover. The set speed of the AERS was set to 4 m/s. Under this set speed on the U-shaped path, the AERS can easily roll over. Therefore, for rollover prevention, the desired lateral acceleration of the MPC controller was set to 1.0 m/s$^2$ based on the experimental data. The simulation results of the proposed and base algorithms are presented in Figures 8 and 9. In addition, Table 2 shows the mean, standard deviation (SD), and a maximum of the lateral and heading errors. The maximum of the traction torque, lateral acceleration, and LTR are summarized in Table 2.

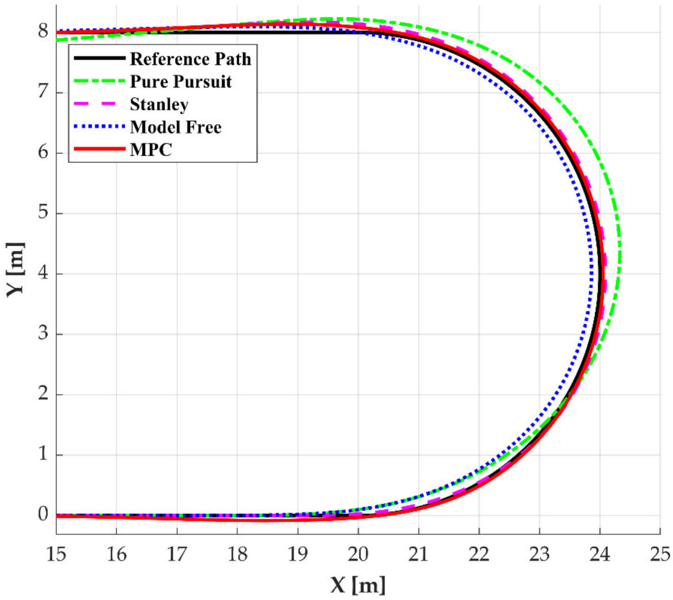
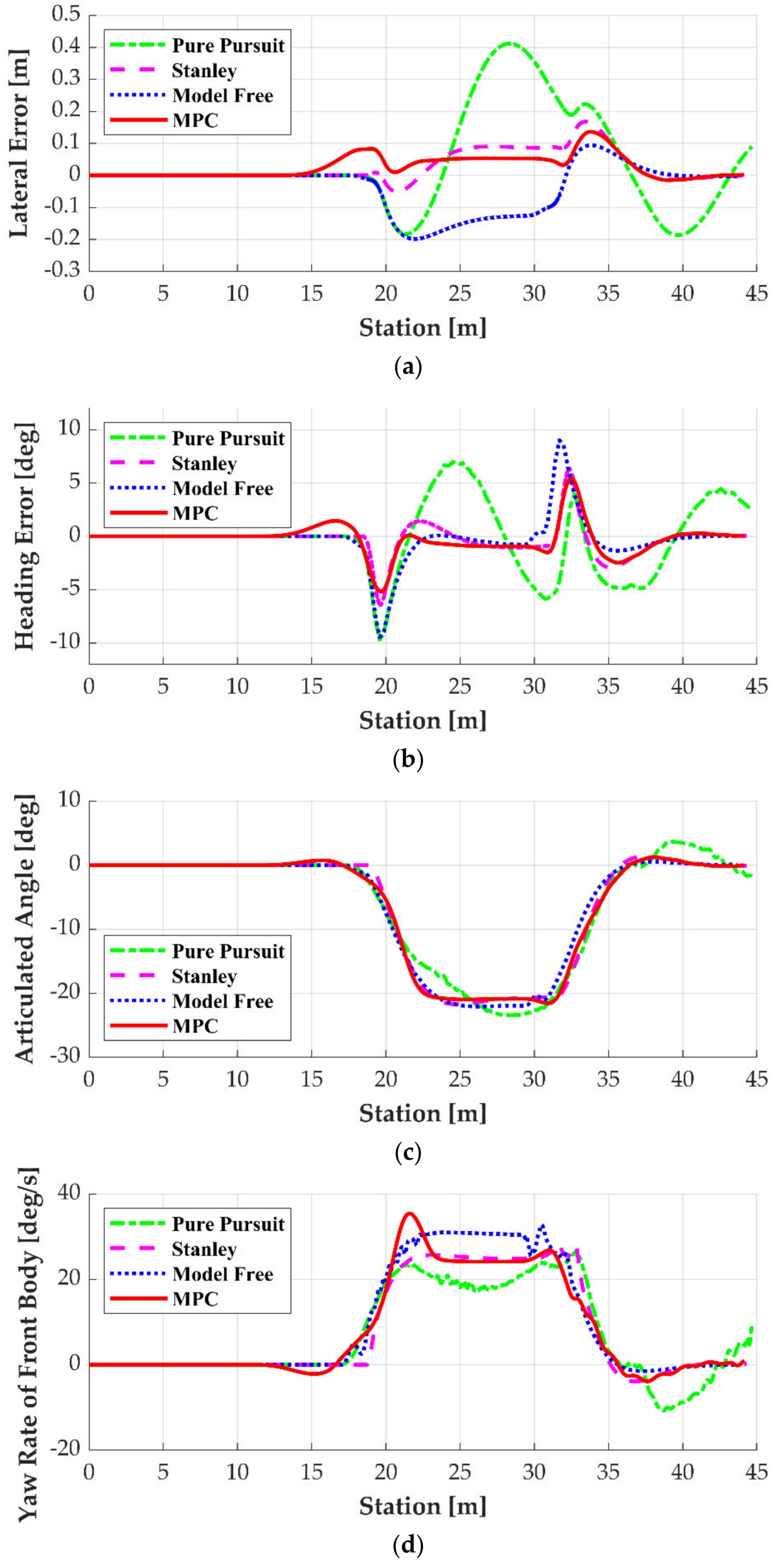


**Figure 8.** Trajectories of each algorithm.

Figure 8 shows the trajectories for each algorithm with the reference path. As shown in Figure 8, the AERS followed the reference path regardless of the control algorithms. The path-tracking controllers, except for the pure pursuit method, showed a small tracking error. As shown in Figure 9a, the lateral error of the pure pursuit method increased up to 0.41 m. Moreover, a large overshoot occurred in the lateral error when the vehicle entered the straight section after cornering. This phenomenon occurred the same in the heading angle error shown in Figure 9b. Meanwhile, the model-free controller followed the reference path better than the pure pursuit method. However, the maximum lateral errors of these two methods increased by twice that of the Stanley method and the proposed algorithm. In terms of the position and heading angle errors, the Stanley method and the proposed algorithm gave a similar performance. As shown in Figure 9c,d, these two methods showed a similar history of the articulation angle and the yaw rate. However, since the proposed algorithm applied the articulation angle input earlier than the base algorithms, the magnitude of the errors was reduced.

Since the desired lateral accelerations were set to the identical value, i.e., 1.0 m/s$^2$, for all the methods, the minimum values of the speed were all similar. As shown in Figure 9e, the speed of the AERS decreased to below 2 m/s. However, the proposed algorithm changed the speed more smoothly than the base algorithms. This caused a decrease in the usage of the traction torque. This is because the proposed algorithm considered the future reference path within the prediction horizon. Thus, the traction torque was applied earlier than base algorithms before entering and exiting the corner. As shown in Figure 9f, the maximum torque was reduced up to 16.7% compared to that of the Stanley method. A severe change of the traction torque resulted in an increase in the lateral acceleration and LTR to 3 m/s$^2$ and 1 or more, respectively. Therefore, it can be confirmed that rollover can easily occur even at a low speed of 4 m/s, which is consistent with the aforementioned characteristics of the AERS. As shown in Figure 9g,h, the maximum lateral acceleration and LTR were reduced to 1.532 $m/s^2$ and 0.433, respectively.



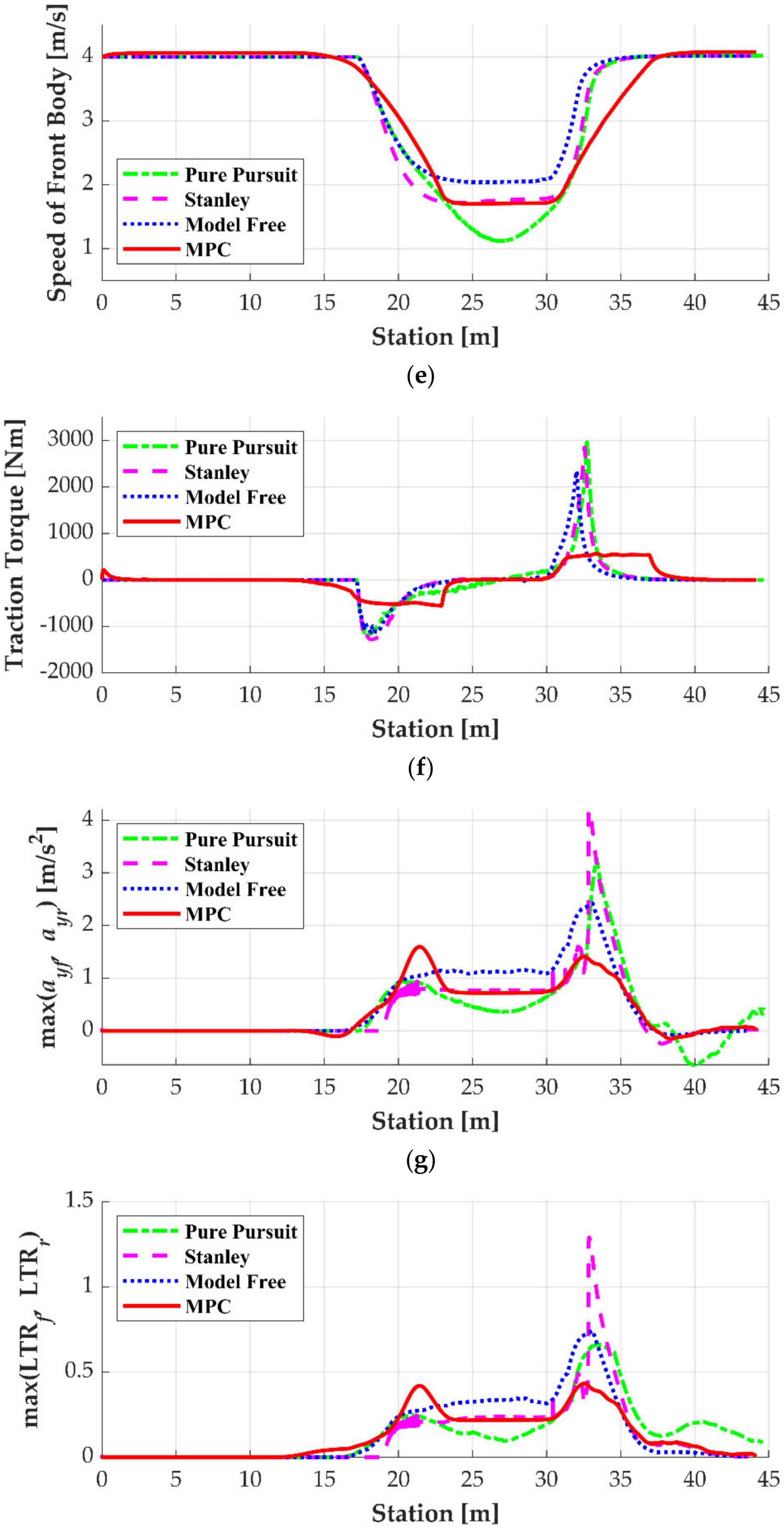
(**a**)

(**b**)

(**c**)

(**d**)

**Figure 9.** *Cont.*



(**e**)

(**f**)

(**g**)

(**h**)

**Figure 9.** Simulation results of each algorithm: (**a**) lateral error, (**b**) heading angle error, (**c**) articulation angle, (**d**) yaw rate of the front body, (**e**) speed of the front body, (**f**) traction/braking torques, (**g**) maximum lateral acceleration of the front and rear bodies, and (**h**) maximum of LTRs of the front and rear bodies.

**Table 2.** Performance comparison between controllers.

| Criteria | Pure Pursuit | Stanley | Model Free | MPC |
|---|---|---|---|---|
| Mean (SD) of lateral error (m) | 0.151<br>(0.144) | 0.041<br>(0.044) | 0.066<br>(0.070) | 0.036<br>(0.032) |
| Max of position error (m) | 0.411 | 0.168 | 0.198 | 0.136 |
| Mean (SD) of heading error (degree) | 2.816<br>(2.513) | 0.861<br>(1.258) | 0.989<br>(2.014) | 0.942<br>(1.156) |
| Max of heading error (degree) | 9.688 | 6.442 | 9.452 | 5.410 |
| Max of traction torque (Nm) | 2963 | 2853 | 2339 | 476.5 |
| Max of $a_y$ (m/s$^2$) | 3.277 | 4.204 | 2.422 | 1.532 |
| Max of LTR | 0.661 | 1.290 | 0.743 | 0.433 |

In addition, the lateral acceleration and LTR when entering and exiting the corner showed similar results compared to the base algorithms. This is because MPC can prepare the change of the road curvature in advance. However, the base algorithms consider a single point on the reference path when determining the desired articulation angle and velocity. This led to heavy usage of acceleration and resulted in deteriorating the stability of the AERS. In other words, when entering the corner, the path-tracking error increased and the risk of the rollover was not high. However, when exiting the corner, the ARES controlled by base algorithms rapidly converged into a straight road. Thus, the lateral acceleration and LTR increased to the risky level of rollover. Therefore, in order to safely drive on the U-shaped road with the base algorithms, the restriction on driving speed should be imposed more severely. This lowering of the operating speed reduces the efficiency of road sweepers. In short, the proposed algorithm is designed to determine the desired articulation angle and velocity while previewing the reference path. Therefore, the proposed algorithm achieved the best performance in terms of path and velocity tracking and rollover prevention as compared to the base algorithms.

## 5. Conclusions

This paper presented the path- and velocity-tracking control algorithm for rollover prevention of the AERS. The ODD of the AERS is limited to low-speed conditions. For this reason, the AERS was modeled with the kinematic model. With the vehicle model, the proposed algorithm was designed with three modules. The reference-state decision module determined the reference states needed to make the AERS follow the reference path and prevent rollover. The maximum lateral acceleration was considered to plan the desired velocity for rollover prevention. With the vehicle model and the reference states, the MPC-based motion controller was designed to determine the articulation angular rate and velocity inputs. The low-level controller determined the actuator input to follow the outputs of the MPC controller. The simulation was conducted to evaluate the effectiveness of the proposed algorithm compared with base algorithms. The simulation results showed that the proposed algorithm improved the path- and velocity-tracking performance and prevented the rollover of the AERS. Further research can include the implementation of the proposed algorithm on real AERS and the test in real driving situations.

**Author Contributions:** The key idea was devised by Y.J., W.K. and S.Y. Equation development and simulation design were implemented by the Y.J. Verification and simulation of the control method were performed by Y.J. and S.Y. This manuscript was drafted by the Y.J. The revised version and final submission were written by Y.J., W.K. and S.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Promotion of Innovative Businesses for Regulation-Free Special Zones funded by the Ministry of SMEs and Startups (MSS, Korea) (No. P0012726).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Peel, G.; Michielen, M.; Parker, G. Some aspects of road sweeping vehicle automation. In Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Como, Italy, 8–12 July 2001.
2. Chang, Y.-M.; Chou, C.-M.; Su, K.-T.; Tseng, C.-H. Effectiveness of street sweeping and washing for controlling ambient TSP. *Atmos. Environ.* **2005**, *39*, 1891–1902. [CrossRef]
3. *Surface Vehicle Standard J2130*; Self-Propelled Sweepers and Cleaning Equipment. SAE Inc.: Warrendale, PA, USA, 1977.
4. Xu, H.; Xiao, J.; Feng, Y. Development and research status of road cleaning vehicle. *J. Phys. Conf. Ser.* **2020**, *1626*, 012153. [CrossRef]
5. Boulanger, A.G.; Chu, A.C.; Maxx, S.; Waltz, D.L. Vehicle electrification: Status and issues. *Proc. IEEE* **2011**, *99*, 1116–1138. [CrossRef]
6. Fjelstul, J. Vehicle electrification: On the "green" road to destination sustainability. *J. Destin. Mark. Manag.* **2014**, *3*, 137–139. [CrossRef]
7. Chen, F.; Taylor, N.; Kringos, N. Electrification of roads: Opportunities and challenges. *Appl. Energy* **2015**, *150*, 109–119. [CrossRef]
8. Fast Facts U.S. Transportation Sector GHG Emissions. Available online: https://www.epa.gov/greenvehicles/archives-fast-facts-us-transportation-sector-greenhouse-gas-emissions (accessed on 29 December 2021).
9. Budich, R.; Hübner, M. Vehicle simulation of an electric street sweeper for substitution analysis. In Proceedings of the 16. Internationales Stuttgarter Symposium, Stuttgart, Germany, 15–16 March 2016.
10. Prassler, E.; Ritter, A.; Schaeffer, C.; Fiorini, P. A short history of cleaning robots. *Auton. Robot.* **2000**, *9*, 211–226. [CrossRef]
11. Prassler, E.; Schwammkrug, D.; Rohrmoser, B.; Schmidl, G. A robotic road sweeper. In Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000.
12. Jeon, J.; Jung, B.; Koo, J.C.; Choi, H.R.; Moon, H.; Pintado, A.; Oh, P. Autonomous robotic street sweeping: Initial attempt for curbside sweeping. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 8–10 January 2017.
13. Li, J. *Research on Intelligent Road Sweeper Path Planning and Dynamic Monitoring System Based on Machine Vision and Internet of Things Technology*; SAE Technical Paper 2020-01-5108; SAE International: Warrendale, PA, USA, 2020.
14. Channel News Asia. Self-Driving Vehicles for Road Sweeping to Go on Trial. Available online: https://www.channelnewsasia.com/singapore/self-driving-vehicles-road-sweeping-to-go-on-trial-nea-894756 (accessed on 21 March 2019).
15. Westfield. AutoSweeper. Available online: https://westfieldavs.com/westfield-autosweep/ (accessed on 25 October 2021).
16. ENWAY. Autonomous Munical Sweeper. Available online: https://www.enway.ai/outdoor-sweeper (accessed on 25 October 2021).
17. Trombia. Trombia Free. Available online: https://trombia.com/free/ (accessed on 25 October 2021).
18. Boschung. Urban-Sweeper S2.0 Autonomous. Available online: https://www.boschung.com/product/urban-sweeper-s2-0-autonomous/ (accessed on 25 October 2021).
19. Korea IT News. Gwangju-Si and Korea Institute of Industrial Technology to Introduce Two Unmanned Special Vehicles Next Month. Available online: https://english.etnews.com/20201118200004 (accessed on 20 November 2020).
20. Dou, F.; Liu, W.; Huang, Y.; Liu, L.; Meng, Y. Modeling and path tracking for articulated steering vehicles. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017.
21. Lei, T.; Wang, J.; Yao, Z. Modelling and stability analysis of articulated vehicles. *Appl. Sci.* **2021**, *11*, 3663. [CrossRef]
22. Rahimi, A.; He, Y. A review of essential technologies for autonomous and semi-autonomous articulated heavy vehicles. In Proceedings of the Canadian Society for Mechanical Engineering International Congress, Charlottetown, PE, Canada, 21–24 June 2020.
23. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.; Mutz, F.; et al. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]
24. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
25. Yaqoob, I.; Khan, L.U.; Kazmi, S.A.; Imran, M.; Guizani, N.; Hong, C.S. Autonomous driving cars in smart cities: Recent advances, requirements, and challenges. *IEEE Netw.* **2019**, *34*, 174–181. [CrossRef]
26. Schwarting, W.; Alonso-Mora, J.; Rus, D. Planning and decision-making for autonomous vehicles. *Annu. Rev. Control Robot. Auton. Syst.* **2018**, *1*, 187–210. [CrossRef]
27. Liu, X.; Madhusudhanan, A.K.; Cebon, D. Experimental Evaluation of Minimum Swept-Path Control for Autonomous Reversing of Articulated Vehicles. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020.
28. Kolb, J.; Nitzsche, G.; Wagner, S.; Röbenack, K. Path Tracking of Articulated Vehicles in Backward Motion. In Proceedings of the 2020 24th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 8–10 October 2020.

29. Zhang, Y.; Khajepour, A.; Hashemi, E.; Qin, Y.; Huang, Y. Reconfigurable model predictive control for articulated vehicle stability with experimental validation. *IEEE Trans. Transp. Electrif.* **2020**, *6*, 308–317. [CrossRef]
30. Xia, G.; Zhao, M.; Tang, X.; Wang, S.; Zhao, L. Linear reversing control of semi-trailer trains based on hitch angle stable and feasible domain. *Control Eng. Pract.* **2020**, *104*, 104625. [CrossRef]
31. Jing, J.; Maroli, J.M.; Bin Salamah, Y.; Hejase, M.; Fiorentini, L.; Özgüner, Ü. Control method designs and comparisons for tractor-trailer vehicle backward path tracking. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019.
32. Zhao, H.; Liu, Z.; Li, Z.; Zhou, S.; Chen, W.; Suo, C.; Liu, Y.H. Modelling and dynamic tracking control of industrial vehicles with tractor-trailer structure. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019.
33. Yongming, B.; Meng, Y.; Xiaojun, F.; Xiahui, W. Kinematics and path following control of an articulated drum roller. *Chin. J. Mech. Eng.* **2017**, *30*, 888–899.
34. Nayl, T.; Nikolakopoulos, G.; Gustafsson, T. Effect of kinematic parameters on MPC based on-line motion planning for an articulated vehicle. *Robot. Auton. Syst.* **2015**, *70*, 16–24. [CrossRef]
35. Nayl, T.; Nikolakopoulos, G.; Gustfsson, T. Switching model predictive control for an articulated vehicle under varying slip angle. In Proceedings of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, 3–6 July 2012.
36. Shi, J.; Sun, D.; Qin, D.; Hu, M.; Kan, Y.; Ma, K.; Chen, R. Planning the trajectory of an autonomous wheel loader and tracking its trajectory via adaptive model predictive control. *Robot. Auton. Syst.* **2020**, *131*, 103570. [CrossRef]
37. Liu, Z.; Yue, M.; Guo, L.; Zhang, Y. Trajectory planning and robust tracking control for a class of active articulated tractor-trailer vehicle with on-axle structure. *Eur. J. Control* **2020**, *54*, 87–98. [CrossRef]
38. Nayl, T.; Nikolakopoulos, G.; Gustafsson, T.; Kominiak, D.; Nyberg, R. Design and experimental evaluation of a novel sliding mode controller for an articulated vehicle. *Robot. Auton. Syst.* **2018**, *103*, 213–221. [CrossRef]
39. Sampei, M.; Tamura, T.; Kobayashi, T.; Shibui, N. Arbitrary path tracking control of articulated vehicles using nonlinear control theory. *IEEE Trans. Control Syst. Technol.* **1995**, *3*, 125–131. [CrossRef]
40. Bolzern, P.; DeSantis, R.M.; Locatelli, A. An input-output linearization approach to the control of an n-body articulated vehicle. *J. Dyn. Syst. Meas. Control* **2001**, *123*, 309–316. [CrossRef]
41. Gao, Y.; Cao, D.; Shen, Y. Path-following control by dynamic virtual terrain field for articulated steer vehicles. *Veh. Syst. Dyn.* **2020**, *58*, 1528–1552. [CrossRef]
42. Snider, J.M. *Automatic Steering Methods for Autonomous Automobile Path Tracking*; Technical Report CMU-RITR-09-08; Robotics Institute: Pittsburgh, PA, USA, 2009.
43. Jung, C.; Kim, H.; Son, Y.; Lee, K.; Yi, K. Parameter adaptive steering control for autonomous driving. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014.
44. Yoon, J.; Kim, D.; Yi, K. Design of a rollover index-based vehicle stability control scheme. *Veh. Syst. Dyn.* **2007**, *45*, 459–475. [CrossRef]
45. Yoon, J.; Yim, S.; Cho, W.; Koo, B.; Yi, K. Design of an unified chassis controller for rollover prevention, manoeuvrability and lateral stability. *Veh. Syst. Dyn.* **2010**, *48*, 1247–1268. [CrossRef]
46. Chen, W.-H.; Ballance, D.J.; O'Reilly, J. Model predictive control of nonlinear systems: Computational burden and stability. *IEE Proc. Control Theory Appl.* **2000**, *147*, 387–394. [CrossRef]
47. Yang, K.; Liu, Y.; Liu, Y.; He, X.; Ji, X. A linear time-varying MPC method for vehicle path-following assistance based on steering torque. In Proceedings of the IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017.
48. Ozaki, T. A local linearization approach to nonlinear filtering. *Int. J. Control* **1993**, *57*, 75–96. [CrossRef]
49. Liang, Y.; Li, Y.; Khajepour, A.; Zheng, L. Holistic adaptive multi-model predictive control for the path following of 4WID autonomous vehicles. *IEEE Trans. Veh. Technol.* **2020**, *70*, 69–81. [CrossRef]
50. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 566–580. [CrossRef]
51. Limón, D.; Alamo, T.; Salas, F.; Camacho, E.F. On the stability of constrained MPC without terminal constraint. *IEEE Trans. Autom. Control* **2006**, *51*, 832–836. [CrossRef]
52. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814. [CrossRef]
53. Mattingley, J.; Boyd, S. CVXGEN: A code generator for embedded convex optimization. *Optim. Eng.* **2021**, *13*, 1–27. [CrossRef]
54. Yim, S. Design of a preview controller for vehicle rollover prevention. *IEEE Trans. Veh. Technol.* **2011**, *60*, 4217–4226. [CrossRef]