electronics

MDPI

Article

Integrating Vehicle Positioning and Path Tracking Practices for an Autonomous Vehicle Prototype in Campus Environment

Jui-An Yang [1] and Chung-Hsien Kuo [2,*]

[1] Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 106335, Taiwan; M10707327@mail.ntust.edu.tw
[2] Department of Mechanical Engineering, National Taiwan University, Taipei 10617, Taiwan
* Correspondence: chunghsien@ntu.edu.tw; Tel.: +886-2-3366-2686

Abstract: This paper presents the implementation of an autonomous electric vehicle (EV) project in the National Taiwan University of Science and Technology (NTUST) campus in Taiwan. The aim of this work was to integrate two important practices of realizing an autonomous vehicle in a campus environment, including vehicle positioning and path tracking. Such a project is helpful to the students to learn and practice key technologies of autonomous vehicles conveniently. Therefore, a laboratory-made EV was equipped with real-time kinematic GPS (RTK-GPS) to provide centimeter position accuracy. Furthermore, the model predictive control (MPC) was proposed to perform the path tracking capability. Nevertheless, the RTK-GPS exhibited some robust positioning concerns in practical application, such as a low update rate, signal obstruction, signal drift, and network instability. To solve this problem, a multisensory fusion approach using an unscented Kalman filter (UKF) was utilized to improve the vehicle positioning performance by further considering an inertial measurement unit (IMU) and wheel odometry. On the other hand, the model predictive control (MPC) is usually used to control autonomous EVs. However, the determination of MPC parameters is a challenging task. Hence, reinforcement learning (RL) was utilized to generalize the pre-trained datum value for the determination of MPC parameters in practice. To evaluate the performance of the RL-based MPC, software simulations using MATLAB and a laboratory-made, full-scale electric vehicle were arranged for experiments and validation. In a 199.27 m campus loop path, the estimated travel distance error was 0.82% in terms of UKF. The MPC parameters generated by RL also achieved a better tracking performance with 0.227 m RMSE in path tracking experiments, and they also achieved a better tracking performance when compared to that of human-tuned MPC parameters.

Keywords: vehicle positioning; path tracking; unscented Kalman filter; model predictive control; reinforcement learning

check for updates

Citation: Yang, J.-A.; Kuo, C.-H. Integrating Vehicle Positioning and Path Tracking Practices for an Autonomous Vehicle Prototype in Campus Environment. *Electronics* **2021**, *10*, 2703. https://doi.org/10.3390/electronics10212703

Academic Editors: Wei Hua and Felipe Jiménez

Received: 6 September 2021
Accepted: 3 November 2021
Published: 5 November 2021

1. Introduction

Vehicle positioning and path tracking are essential to the development of autonomous vehicles. Vehicle positioning is responsible for providing stable, real-time, and accurate location information for navigations. The path tracking is developed to navigate the vehicle by following the desired trajectory to reduce the tracking errors. In general, the position accuracy with a conventional, stand-alone global positioning system (GPS) is approximately 0.5 m to 3.5 m on average. By applying real-time kinematic (RTK) technology, the GPS positioning accuracy can be improved to the centimeter level. However, the RTK-GPS performance is still restricted when jumping and drifting occur in an obstructed area, such as a blockage and multipath effects in an urban region with buildings and landscapes, under a bridge, or in a tunnel.

Due to the aforementioned RTK-GPS limitations, wheel odometry and inertial measurement units (IMUs) were complementarily combined with the global navigation satellite system (GNSS) to realize a stable and accurate multisensory, fusion-based vehicle positioning system. The aforementioned multisensory fusion problem is nonlinear; hence, an extended Kalman filter (EKF) or unscented Kalman filter (UKF) could be considered. However, the UKF outperforms the conventional EKF in terms of accuracy performance at a comparable level of system complexity [1]. Hence, this work used the UKF as the vehicle position estimation.

On the other hand, a commonly used model predictive control (MPC) method in a dynamic vehicle control system was further utilized in this work. The MPC controller calculates the system output according to the linear time-varying (LTV) model. Nevertheless, due to vehicle dynamics, hardware limitations, and environmental disturbances, system stability and trajectory tracking accuracy were a challenge. The MPC parameter settings are highly related to the controller performance. Practically, trial-and-error blind tuning of MPC parameters takes time and is inefficient. Therefore, applying reinforcement learning (RL) is a helpful way to produce proper MPC parameters to improve the trajectory tracking performance in terms of defining the rewards, states, and actions. Such an RL model works based on the tuning experience of the human MPC model parameters. The pre-trained MPC parameters are capable of providing the datum value rather than trial-and-error. As a consequence, the MPC parameters generated by the RL methods efficiently and effectively supported the MPC to perform an accurate path tracking performance. Such MPC performance measures were evaluated in terms of a simulation environment and a laboratory-made, full-scale electric vehicle.

The rest of the paper is organized as follows. Section 2 surveys the related works. The methods regarding the system architecture, vehicle model, implementation of the UKF-based position estimation, and the RL-based MPC algorithm are discussed in Section 3. In Section 4, the simulation of the proposed system and experiments on the evaluations of the position estimator and RL-based MPC trajectory tracking with a full-scale EV are elaborated. Finally, the conclusion of the proposed study and future works are presented in Section 5.

2. Related Works

This paper first surveys the related works within vehicle positioning. In general, a stand-alone GPS could suffer from a signal mismatch or failure. In addition, inaccurate GPS positioning cannot be directly applied to autonomous vehicle driving purposes unless extra efforts are made, such as image-based lane detection methods [2]. RTK-GPS provides a center centimeter level, and it has been widely used in low-speed (1 Hz) surveying and mapping systems. With the RTK (fixed mode), the position error may be less than 10 cm by following the radiotechnical commission for maritime (RTCM) service standards. Furthermore, the strength of the signal must be larger than 40 dB, and it is expected to receive 16 satellites normally to meet the lowest requirements [3].

Practically, the RTK-GPS is basically composed of a fixed base station and a rover to reduce the rover's positioning error. Hence, communication between the base station and the rover must be established. An RF module is convenient; however, the disadvantage of utilizing RF modules is that the transmission distance might be limited by the rated power or environment interference. Hence, the stability of signal transmission using RF modules is a challenge [4].

When applying RTK-GPS as a solution to autonomous driving, low-evaluation satellites may suffer from larger atmospheric errors. Practically, implementation with a Kalman filter (KF) estimation could obtain integer ambiguities that allow individuals to be corrected by all ambiguity parameters in practical applications [5]. Moreover, the performance of RTK-GPS may be limited when driving in an interferential environment. The performance evaluation method could be performed in terms of statistically comparing the traveled path by using the mean absolute error and the root mean square (RMS) error, as addressed in [6].

Network instability and component failure are concerns of applying RTK-GPS in practical usages. Hence, Um et al. [7] proposed a redundancy solution, and their system could switch the master system to one of the available slave systems to deal with such network or component failures. In addition, the GPS accuracy may drop due to the multipath error that could occur during signal reflections among large amounts of objects. Therefore, the GSA message was used to provide position dilution of precision (PDOP) measurements according to the geometric constellation of GPS satellites [8,9].

Based on the NMEA (National Electrical Manufacturers Association) GST instructions, the standard deviations of latitude and longitude values are known. Such information could be applied for EKF-based visual odometry and GPS sensor fusion solutions. Once the measurement data are convincing enough, the visual odometry position can be corrected by GPS [10]. Furthermore, the transformation matrix of GPS can be determined to transform the local tangent plane (LTP) coordinates to world coordinates [11]. Within the considerations of the RTK carrier phase ambiguities and the pseudorange multipath errors, the state prediction could be more robust and precise [12]. Odometry methods are categorized into five main types, i.e., wheel, inertial, laser, radar, and visual. Most of the research applied multiple sensors to fuse or optimize the estimation of vehicle position [13].

In general, accumulative error is a critical factor of odometry. Hence, Mikov et al. [14] integrated inertial MEMS (Micro-Electromechanical Systems) and odometry solutions for land vehicle localizations. When the GNSS (Global Navigation Satellite System) signal is available, it can become the odometry correction factor to fix the position. The characteristics of GNSS and IMU are complementary, and they are fused in terms of KFs. Their results achieved a higher sampling rate and a high bandwidth by utilizing an online error state covariance estimation [15]. In addition to the EKF, the UKF is suitable for solving nonlinear problems. The advantage of using the UKF is that it does not need to calculate the Jacobian matrix and apply Taylor expansion. Hence, the UKF eventually results in a higher precision of estimation.

In addition to vehicle positioning, the related works within path tracking problems are also discussed. The single-track kinematics model is simple, and it is often applied to the MPC controller design. The optimal steering wheel angle and longitudinal force can be obtained by using the standard quadratic cost function based on the design constraints [16–18]. Under different speed ranges, the prediction model needs to be robust enough to describe the overall vehicle motion behaviors; otherwise, the prediction error will significantly increase [19].

Tang et al. [20] presented the lane changing control method in highway environments. Their study applied MPC as a controller for lane changing path planning. A multi-constraint model predictive control algorithm and optimization for lane changing were proposed. By determining the Q and R cost coefficients in the MPC model, the objective function can be minimized after the iterations. Assuming that the path tracking controller and driver behavior can be synthesized, the robustness and performance of driver-oriented maneuvers, such as cutting in and overtaking, can be further improved [21,22].

Optimization-based autonomous racing RC (Remote Control) cars were also carried out in [23]. That study combined the multitask problem as a nonlinear optimization problem (NLP) for contouring control. The proposed control system was created to build local approximations of NLPs in convex quadratic programs (QPs) at each time step. The algorithm solved the constrained infinite horizon optimal control problem at each time step and iteration. Moreover, an approximated safe set was added to satisfy the real-time computation and to prevent the occurrence of insoluble solutions [24]. The recursive least square algorithm and first-order Lagrange interpolation can be used to further estimate the cornering stiffness and road friction [25].

On the other hand, MPC is highly dependent on the prediction model. The properties of MPC and reinforcement learning (RL) are quite different but complementary, such as model requirements, robustness, stability, feasibility, etc. [26]. The RL applied the concept of the Markov decision process. Based on this assumption, the whole model can be simplified and the next state can be estimated [27].

The combination of RL and MPC was successfully applied to connected and automated vehicles at intersections [28]. The trained coordinate policy in a simulation setting and proximal policy optimization were applied. Their results demonstrated that the computing time and traffic efficiency outperformed other works. Another work [29] was proposed by Tange et al. Their study proposed model predictive control based on the deep RL approach. By evaluating the control performance and learning convergence, their discrete-valued input deep RL method achieved a better performance than direct P-based and PI-based RL methods.

Finally, parameter tuning could be applied by RL methods in various fields. A segmented and adaptive RL approach was proposed in [30] to provide successful noise reduction. By searching possible solution spaces, all the coefficient values can be well tuned in a short learning time. Such an RL-based parameter tuning concept is also applied in this paper to find the MPC parameters.

3. Methods

3.1. System Architecture

The overall system is primarily composed of a UKF-based position estimator and an RL-based MPC (RLMPC) controller design. The Robot Operating System (ROS) system was used in this work to develop all the programs. The first part addressed the UKF-based vehicle positioning system, and stable and accurate vehicle position information was further used for path tracking. The overall operation flowchart of this study is shown in Figure 1.

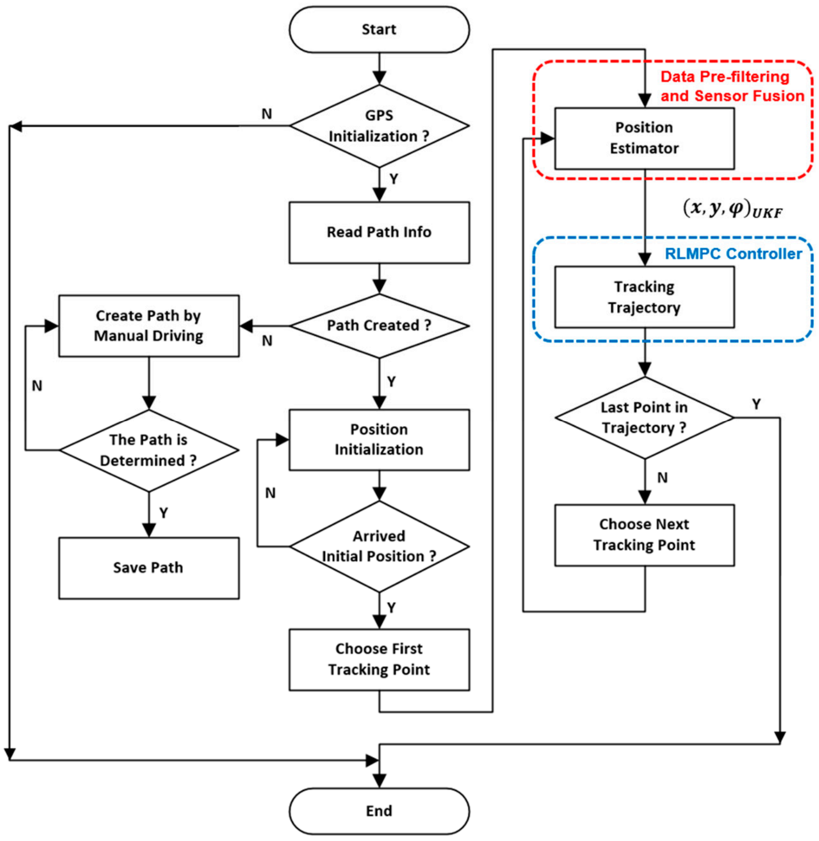

Figure 1. Overall operation flowchart.

When the operator intends to start the autonomous driving function, the initialization of the GPS device needs to be ensured. Once the RTK correction signal is received and set, the electric vehicle (EV) is ready for operation. It is noted that the 4G communication modules are convenient to provide a stable signal transmission channel between the base station and the rover.

Autonomous driving also must ensure that the desired navigation path is created. If the path file is not found, a vehicle mission path file can be created either by recording the manual drive path by receiving the UKF vehicle positions or by planning. After the mission path file is loaded, the EV can start the navigation mission at any arbitrary location near the path start location of the trip. The mission task is divided into two parts: The first part is to autonomously drive from the vehicle initial position to the desired path start location; the second part is the path mission part, and the RLMPC controller is capable of tracking the desired trajectory from the path start location to the path destination location by maintaining minimum tracking error. The details of UKF position estimation and RLMPC path tracking will be explained in detail in Sections 3.3 and 3.4.

3.2. Vehicle Modeling

To evaluate the performance of vehicle positioning and path tracking, a full-scale, laboratory-made electric vehicle (EV) was used for experiments on the campus of National Taiwan University of Science and Technology (NTUST). A kinematic bicycle model was applied since the steering geometry and the control maneuvers can only fit at low speeds and low lateral acceleration in the campus experimental setting. The EV for experiments is shown on the left-hand side of Figure 2. The dimensions of the EV were 3.57 m (L) $\times$ 1.55 m (W) $\times$ 1.87 m (H), and the vehicle was 1300 kg in weight. The maximum speed was 30 km/h to meet the campus driving setting. The RTK-GPS receiver was installed on the top center of the vehicle. The rotary encoders were mounted inside two rear motors, and the IMU was fixed in front of the EV. It is noted that two 2-D outdoor usage lidars placed inside the right front and left rear of the front and rear bumpers were used to form a 360° detection of obstacles for collision avoidance. Because reactive obstacle avoidance was not the purpose of this paper, the vehicle temporarily stopped autonomous navigation when any obstacles were detected from two LiDars within a 2 m range for safety.

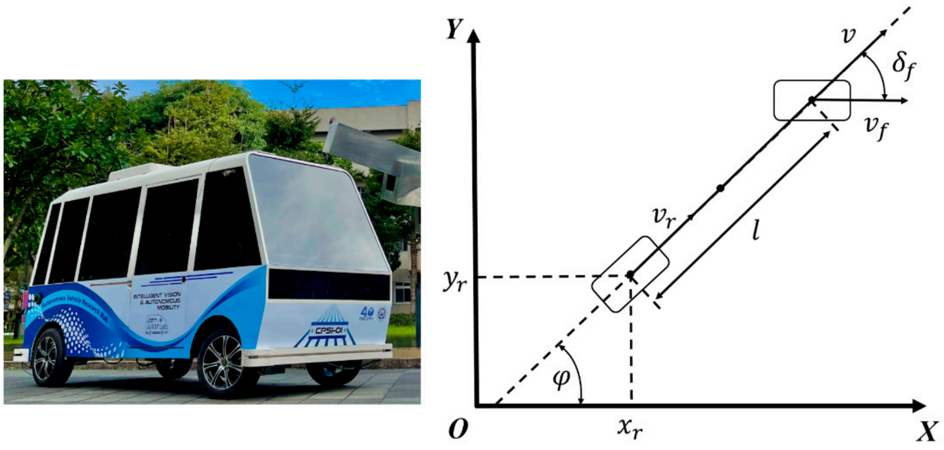

Figure 2. A laboratory-made EV for experiments in this paper (**left-hand-side**); the kinematic bicycle model for illustrating the kinematics (**right-hand-side**).

Based on the EV model, the inertial reference frames OXY, (x_f, y_f), and (x_r, y_r) were the coordinates at the front axle and rear axle, respectively, as shown on the right-hand side of Figure 2, where φ is the vehicle heading angle, $v = v_r$, and it is the magnitude of the velocity vector of the center mass, v_f and v_r are the center velocities of the front axle and rear axle, respectively, l_f and l_r are the distances to the front axle and rear axle from the center of mass, respectively, l is the distance from the front axle to the rear axle, and δ_f is the steering angle. It was assumed that there was no slip between the road and the tire for low-speed campus driving. The velocity model can be obtained in Equations (1) and (2).

$$\begin{cases} \dot{x}_r = v_r \cos\varphi \\ \dot{y}_r = v_r \sin\varphi \end{cases} \tag{1}$$

$$\begin{cases} R = v_r/\omega \\ \delta_f = \tan^{-1}(l/R) \end{cases} \tag{2}$$

From Equations (1) and (2), the vehicle kinematic model can be derived in Equation (3)

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \tan\delta_f/l \end{bmatrix} v_r \tag{3}$$

The kinematic bicycle model can be denoted in a generalized form $\dot{\xi}_{\text{kin}} = f_{\text{kin}}(\xi_{\text{kin}}, u_{\text{kin}})$. The state is $\xi_{\text{kin}} = [x_r,\ y_r, \varphi]^T$ and the control vector is $u_{\text{kin}} = [v_r,\ \delta_f]^T$.

3.3. UKF-Based Position Estimation

In this subsection, the position estimator consisted of two steps: (1) data prefiltering and (2) multisensory fusion. To accomplish the prefiltering of the poor data point, the covariance matrices were defined as the inverse matrix of the covariance matrix of RTK (C_{rtk}) and the covariance matrix of odometry (C_{odm}). The vectors collected from stand-alone RTK-GPS and odometry were $X_{\text{rtk}} = (x_{\text{rtk}}, y_{\text{rtk}})^T$ and $X_{\text{odm}} = (x_{\text{odm}}, y_{\text{odm}})^T$, respectively. The sampling rates of the RTK and odometry were 1 Hz and 10 Hz, respectively. Due to the different sampling rates of the sensors, orientation jumping occurred severely. Hence, in this paper, the Mahalanobis distances M_p and M_φ are defined independently as Equations (4) and (5). In addition, the covariance matrices are defined in Equations (6) and (7).

$$M_p(t) = \sqrt{(X_{\text{rtk}}(t) - X_{\text{odm}}(t))^T(\Sigma C_{\text{rtk}} + \Sigma C_{\text{odm}})^{-1}(X_{\text{rtk}}(t) - X_{\text{odm}}(t))} \tag{4}$$

$$M_\varphi(t) = \sqrt{(\varphi_{\text{rtk}}(t) - \varphi_{\text{odm}}(t))^T(\sigma_{\varphi\text{rtk}}{}^2 + \sigma_{\varphi\text{odm}}{}^2(t))^{-1}(\varphi_{\text{rtk}}(t) - \varphi_{\text{odm}}(t))} \tag{5}$$

$$C_{\text{rtk}} = \begin{bmatrix} \sigma_{\text{xrtk}}{}^2 & 0 \\ 0 & \sigma_{\text{yrtk}}{}^2 \end{bmatrix} = \begin{bmatrix} 0.36 & 0 \\ 0 & 0.36 \end{bmatrix};\ C_{\text{odm}} = \begin{bmatrix} \sigma_{\text{xodm}}{}^2 & \sigma_{\text{xyodm}} \\ \sigma_{\text{xyodm}} & \sigma_{\text{yodm}}{}^2 \end{bmatrix} = \begin{bmatrix} 3.1 & 1.2 \\ 1.2 & 3.1 \end{bmatrix} \tag{6}$$

$$\sigma_{\varphi\text{rtk}} = 1.5;\ \sigma_{\varphi\text{odm}} = 0.5 \tag{7}$$

Within the RTK data, the covariance matrix C_{rtk} and the standard deviation $\sigma_{\varphi\text{rtk}}$ were obtained by the fixed-point experiments, and they were compared with the GGA as well as the VTG message from the NMEA standard. Practically, it was impossible to obtain an accurate position when the vehicle was traveling. This is the reason why the statistical data were measured by fixed points. Meanwhile, the covariance matrix C_{odm} and standard deviation $\sigma_{\varphi\text{odm}}$ were based on the experiments that drove through a fixed distance, and they were calculated from the recorded data.

Moreover, M_p and M_φ are the measurement errors, which need to be lower than the threshold values of $M_{p,\text{thres}}$ and $M_{\varphi,\text{thres}}$. The validation criterion depends on the threshold values, and a lower M_{thres} results in stricter judgment. During the iteration, once the measured error is lower than M_{thres}, the sample point X_{odm} will be inputted to the UKF framework. Otherwise, the sample point will be ignored. In this paper, the threshold values for the proposed EV are $M_{p,\text{thres}} = 4.75$ and $M_{\varphi,\text{thres}} = 0.6$. Both settings were determined by empirical knowledge in this model. By applying the above rules, the pre-filtered data points can be divided into four types, which are described as follows.

Case 1: If $M_p < M_{p,\text{thres}}$ and $M_\varphi < M_{\varphi,\text{thres}}$, the correction set of X_{pf} will take into account both position and orientation data from odometry, where X_{pf} and the covariance matrix of measurement noise R_{ukf} are defined as in Equations (8) and (9):

$$X_{\text{pf}} = (x_{\text{odm}}(t), y_{\text{odm}}(t), \varphi_{\text{odm}}(t))^T \tag{8}$$

$$R_{\text{ukf}} = \text{diag}\left({\sigma_{\text{xodm}}}^2,\ {\sigma_{\text{yodm}}}^2, {\sigma_{\varphi\text{odm}}}^2\right) \tag{9}$$

Case 2: If $M_p < M_{p,\text{thres}}$ and $M_\varphi \geq M_{\varphi,\text{thres}}$, the correction set of X_{pf} will only take the position data from odometry, where X_{pf} and the covariance matrix of measurement noise R_{ukf} are defined as in Equations (10) and (11):

$$X_{\text{pf}} = (x_{\text{odm}}(t), y_{\text{odm}}(t), \varphi_{\text{lukf}}(t))^T \tag{10}$$

$$R_{\text{ukf}} = \text{diag}\left({\sigma_{\text{xodm}}}^2,\ {\sigma_{\text{yodm}}}^2, 0\right) \tag{11}$$

It is noted that $\varphi_{\text{lukf}}(t)$ is the estimated orientation of the UKF from the last iteration.
Case 3: If $M_p \geq M_{p,\text{thres}}$ and $M_\varphi < M_{\varphi,\text{thres}}$, the correction set of X_{pf} will only take the orientation data from odometry, where X_{pf} and the covariance matrix of measurement noise R_{ukf} are defined as in Equations (12) and (13):

$$X_{\text{pf}} = (x_{\text{lukf}}(t), y_{\text{lukf}}(t), \varphi_{\text{odm}}(t))^T \tag{12}$$

$$R_{\text{ukf}} = \text{diag}\left(0,\ 0, {\sigma_{\varphi\text{odm}}}^2\right) \tag{13}$$

It is noted that $x_{\text{lukf}}(t)$ and $y_{\text{lukf}}(t)$ are the components of the estimated position of the UKF from the last iteration.
Case 4: If $M_p \geq M_{p,\text{thres}}$ and $M_\varphi \geq M_{\varphi,\text{thres}}$, the vehicle will not take the odometry measurement data as the correction input of the UKF.

After the above processes, the pre-filtered data point X_{pf} can be obtained, and it will be used as the observation of UKF frameworks.

It is noted that the UKF worked at 10 Hz in this work. Based on the unscented transformation (UT), the use of sigma points to describe the nonlinear system would not sacrifice the high-order terms that make the system more precise. Compared with the EKF, the UKF can achieve higher efficiency and marvelous performance without calculating the Jacobian matrix and applying the Tylor expansion. The UKF iteration includes prediction and update processes. The UKF is briefly introduced [1]. Assume that there is a discrete-time, nonlinear system, as in Equation (14).

$$X_{k+1} = \Gamma(X_{k,k}, Q_k);\ Y_k = G(X_k, R_k) \tag{14}$$

where Γ and G represent the dynamic model of the nonlinear systems, and they are assumed to be known, X_k is the unobserved state of the system, $_k$ is the system input, Y_k is the only observed state, Q_k is the process noise, and R_k is the measurement noise of the system observation. When the variable X with dimension L goes through a nonlinear system $Y = G(X)$, X has the mean $\overline{X}$ and the covariance P_X. Then, a matrix χ called a sigma matrix with size 2L + 1 is formed. The sigma point χ_i and the corresponding weights W_i can be defined as follows:

$$\lambda = \alpha^2(L+\kappa) - L;\ \chi_k^{(0)} = \overline{X} \tag{15}$$

$$\chi_k^{(i)} = \overline{X} + \left(\sqrt{(L+\lambda)P_X}\right)_i, i = 1, 2, \ldots, L;\ \chi_k^{(i)} = \overline{X} - \left(\sqrt{(L+\lambda)P_X}\right)_{i-L}, i = L+1, \ldots, 2L \tag{16}$$

$$W_0^{(m)} = \frac{\lambda}{(L+\lambda)}, W_0^{(c)} = \frac{\lambda}{(L+\lambda)} + (1-\alpha^2+\beta),\ W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)}, i = 1, 2, \ldots, 2L \tag{17}$$

where λ is the scaling parameter, α can be used to determine the spread of the sigma point around $\overline{X}$ and is usually set to a small positive value such as 0.01, β is used to combine prior knowledge of the distribution of X, κ is a secondary scaling parameter that is set to 0, and $\left(\sqrt{(L+\lambda)P_X}\right)_i$ is the i-th row of the matrix square root that predicts the sigma point with the transformation matrix Γ. Based on the weights of each sigma point, the predicted mean $\overline{X}_{k|k}^{-}$ and the predicted covariance matrix $P_{k|X}^{-}$ with the process noise R_k can be obtained.

$$\overline{X}_{k|k}^{-} = \sum_{i=0}^{2L} W_i^{(m)} \chi_{k|k-1}^{(i)} \tag{18}$$

$$P_{k|X}^{-} = \sum_{i=0}^{2L} W_i^{(m)} \left(\chi_{k|k-1}^{(i)} - \overline{X}_{k|k}^{-}\right)\left(\chi_{k|k-1}^{(i)} - \overline{X}_{k|k}^{-}\right)^T + R_k \tag{19}$$

Moreover, the calculated sigma point will propagate through the nonlinear function G. The approximation of the measurement means $\overline{Y}_{k|k}^{-}$ based on the predicted state is indicated in Equation (20):

$$\overline{Y}_{k|k}^{-} = \sum_{i=0}^{2L} W_i^{(m)} Y_{k|k-1}^{(i)} \tag{20}$$

The measurement covariance matrix $P_{\overline{Y}_k \overline{Y}_k}$ with measurement noise Q_k and the covariance matrix $P_{X_k Y_k}$ of the cross-correlation measurement for Y are estimated by using the weighted mean and the covariance of the posterior sigma point, as indicated in Equations (21) and (22):

$$P_{\overline{Y}_k \overline{Y}_k} = \sum_{i=0}^{2L} W_i^{(c)} \left(Y_{k|k-1}^{(i)} - \overline{Y}_{k|k}^{-}\right)\left(Y_{k|k-1}^{(i)} - \overline{Y}_{k|k}^{-}\right)^T + Q_k \tag{21}$$

$$P_{X_k Y_k} = \sum_{i=0}^{2L} W_i^{(c)} \left(\chi_{k|k-1}^{(i)} - \overline{X}_{k|k}^{-}\right)\left(Y_{k|k-1}^{(i)} - \overline{Y}_{k|k}^{-}\right)^T \tag{22}$$

Finally, the system updates the mean of the system state and its covariance matrix and then calculates the Kalman gain K_k

$$K_k = P_{X_k Y_k} {P_{\overline{Y}_k \overline{Y}_k}}^{-1} \tag{23}$$

$$\overline{X}_{k|k} = \overline{X}_{k|k}^{-} + K_k\left(Y_k - \overline{Y}_{k|k}^{-}\right) \tag{24}$$

$$P_{k|X} = P_{k|X}^{-} - K_k P_{\overline{Y}_k \overline{Y}_k} {K_k}^T \tag{25}$$

Assume that the driving surface is a plane; hence, the vehicle motion state and input can be expressed as:

$$Y_k = \begin{bmatrix} x_{\text{rtk},k} \\ y_{\text{rtk},k} \\ \varphi_{\text{rtk},k} \end{bmatrix},\ X_k = X_{pf},\ \mho_k = \begin{bmatrix} D_k \\ \Delta\varphi_k \end{bmatrix} \tag{26}$$

where D_k, $\Delta\varphi_k$, and the state equation are defined as follows:

$$D_k = \sqrt{(d_x)^2 + \left(d_y\right)^2} \tag{27}$$

$$\Delta\varphi_k = \varphi_k - \varphi_{k-1} \tag{28}$$

$$\begin{bmatrix} x_k \\ y_k \\ \varphi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + D_k \cos(\varphi_k + \Delta\varphi_k) \\ x_{k-1} + D_k \sin(\varphi_k + \Delta\varphi_k) \\ \varphi_{k-1} + \Delta\varphi_k \end{bmatrix} \tag{29}$$

The number of the input state is 3. As a consequence, $L = 3$, $\kappa = 0$, and $\alpha = 0.01$. According to the Gaussian distribution, $\beta = 2$ is optimal; hence, $\lambda = -2.9997$. The definitions of the process noise matrix Q_k and measurement noise R_k are shown as follows:

$$Q_k = \begin{bmatrix} {\sigma_{\text{lat}}}^2 & \sigma_{\text{lat}}\sigma_{\text{lon}} & \sigma_{\text{lat}}\sigma_{\varphi\text{rtk}} \\ \sigma_{\text{lat}}\sigma_{\text{lon}} & {\sigma_{\text{lon}}}^2 & \sigma_{\varphi\text{rtk}}\sigma_{\text{lon}} \\ \sigma_{\varphi\text{rtk}}\sigma_{\text{lat}} & \sigma_{\varphi\text{rtk}}\sigma_{\text{lon}} & {\sigma_{\varphi\text{rtk}}}^2 \end{bmatrix} \tag{30}$$

$$R_k = R_{\text{ukf}} \tag{31}$$

The standard deviations of the latitude, longitude, and orientation are obtained from the GST message, and they are compared with the position dilution of precision (PDOP) from the GSA message. If the PDOP is greater than the PDOP_{avg} (i.e., =1.5), the standard deviation will stay with the values equaling 0.6 for latitude and longitude and 1.5 for orientation. The values are obtained by experiments; otherwise, the standard deviation will be dynamic with the GST message. After the completion of the UKF framework definition, the position estimator can provide robust positioning ability by fusing the RTK-GPS signal and IMU/odometry.

3.4. Reinforcement Learning-Based Model Predictive Control

When designing the EV trajectory tracking controller, the prediction model needs to be robust enough to describe the overall dynamics of the system. In addition, the system model also needs to be simple enough, allowing the optimization problem to be computed in real time. In this paper, the prediction model and the quadratic cost function focus on a linear time-varying (LTV) model as the validation criterion. The vehicle state equation $\dot{X}$ and its reference $\dot{X}_r$ used in the MPC controller are shown as follows:

$$\dot{X}_r = f(X_r, u_r),\ \dot{X} = f(X, u) \tag{32}$$

In Equation (32), f is a state function. By applying the Tylor expansion, the state-space model for the MPC can be expressed as:

$$\dot{\widetilde{X}} = A(t)\widetilde{X} + B(t)\widetilde{u} \tag{33}$$

where

$$A(t) = \begin{bmatrix} 0 & 0 & -v_r \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & v_r \cos\varphi & \sin\varphi & 0 & 0 \\ 0 & 0 & 0 & \tan\delta_f / l & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ f'(x) & -1 & v_r e\cos(e\varphi)dt & \sin(e\varphi)dt & 0 & v_r \cos(e\varphi)dt \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},\ B(t) = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ \tan\delta_f / l & v_r / l \cos^2\delta_f \\ 1 & 0 \\ \sin(e\varphi)dt & 0 \\ 0 & 0 \end{bmatrix} \tag{34}$$

$\widetilde{X} = [x\ y\ \varphi\ v_r\ cte\ e\varphi]^T$; $\widetilde{u} = \left[v_r\ \delta_f\right]^T$; and v_r, cte, and $e\varphi$ are the reference velocity, tracking error, and orientation error, respectively. In addition, the variables in Equation (34) refer to the kinematic bicycle model indicated in Figure 2. It is noted that this equation is derived in terms of the partial differentiation from the state vectors and control vectors, which applied a basic single-trail bicycle model and the parameters indicated in the paragraph. Moreover, Equation (33) needs to be discretized since the MPC algorithm is only used for the discrete control system, as shown in Equation (35). It is noted that the forward Euler discretization was applied in our proposed algorithm.

$$\widetilde{X}(k+1) = A_{k,t}\widetilde{X}(k) + B_{k,t}\widetilde{u}(k) \tag{35}$$

With the sampling time T, $A_{k,t} = I + TA(t)$ and $B_{k,t} = TB(t)$. On this basis, the new state space Equation (37) can be defined with the transformation shown in Equation (36):

$$\xi(k|t) = \begin{bmatrix} \widetilde{X}(k|t) \\ \widetilde{u}(k-1|t) \end{bmatrix} \tag{36}$$

$$\begin{cases} \xi(k+1|t) = \widetilde{A}_{k,t}\xi(k|t) + \widetilde{B}_{k,t}\Delta u(k|t) \\ \eta(k|t) = \widetilde{C}_{k,t}\xi(k|t) \end{cases}, \text{ and} \tag{37}$$

$$\widetilde{A}_{k,t} = \begin{bmatrix} A_{k,t} & B_{k,t} \\ 0_{m\times n} & I_m \end{bmatrix}, \widetilde{B}_{k,t} = \begin{bmatrix} B_{k,t} \\ I_m \end{bmatrix}, \widetilde{C}_{k,t} = \begin{bmatrix} C_{k,t} & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{38}$$

Assume that N_p (i.e., =8) and N_c (i.e., =7) are the prediction horizon and control horizon of the system. To simplify the calculation, we assume that $A_{k,t} = A_t$, $B_{k,t} = B_t$, and $k = 1, 2, \ldots, t + N - 1$. Hence, the prediction formula can be expressed as:

$$Y(t) = \Psi\xi(t|t) + \Theta\Delta U(t) \tag{39}$$

where

$$\begin{gathered} Y(t) = \left[\eta(t+1|t)\ \eta(t+2|t)\cdots\eta(t+N_p|t)\right]^T,\ \Psi = \left[\widetilde{C}_t\widetilde{A}_t\ \widetilde{C}_t\widetilde{A}_t{}^2\cdots\widetilde{C}_t\widetilde{A}_t{}^{N_p}\right]^T, \\ \Theta = \begin{bmatrix} \widetilde{C}_t\widetilde{B}_t & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \widetilde{C}_t\widetilde{A}_t{}^{N_p-1}\widetilde{B}_t & \cdots & \widetilde{C}_t\widetilde{A}_t{}^{N_p-N_c-1}\widetilde{B}_t \end{bmatrix}, \\ \Delta U(t) = \left[\Delta u(t|t)\ \Delta u(t+1|t)\cdots\Delta u(t+N_c|t)\right]^T \end{gathered} \tag{40}$$

To guarantee that the autonomous vehicle can track the reference trajectory fast and smoothly, the controller needs to optimize the system state and control increments. As a result, in the process of controller design, the objective function is considered as follows:

$$J(k) = \sum_{i=1}^{N_p} ||\eta(k+i|t) - \eta_{ref}(k+i|t)||^2_{Q_{mpc}} + \sum_{i=1}^{N_c-1} ||\Delta U(k+i|t)||^2_{R_{mpc}} + \rho\varepsilon^2 \tag{41}$$

The first term in Equation (41) is the tracking error between the current pose and the reference path, which represents the tracking ability of the reference path. Additionally, our goal is to minimize the term in order to get the optimized result in each iteration, where

$$Q_{mpc} = \mathrm{diag}\Big(Q_{n+1}, Q_{n+2}, \cdots, Q_{n+N_p}\Big),\ R_{mpc} = \mathrm{diag}\Big(R_n, R_{n+1}, \cdots, R_{n+N_p-1}\Big) \tag{42}$$

$$(k) = \begin{bmatrix} x \\ y \\ \varphi \\ v(k) \\ cte(k) \\ e\varphi(k) \end{bmatrix},\ \eta_{ref}(k) = \begin{bmatrix} x_{ref} \\ y_{ref} \\ \varphi_{ref} \\ v_{max} \\ 0 \\ 0 \end{bmatrix},\ \Delta U(k) = \begin{bmatrix} \Delta a \\ \Delta\delta \end{bmatrix} \tag{43}$$

The first term in Equation (41) represents the tracking ability of the reference path. The second term represents the constraints to the change of the control vector. Q_{mpc} and R_{mpc} are the weighting matrices where $Q_{mpc} \succcurlyeq 0$ and $R_{mpc} \succcurlyeq 0$. The mutation of the control vector will be inevitable, and it affects the continuity of the control vector since the objective function is not able to constrain the increments of the control vector in every period. Hence, soft constraints are applied to the implementation objective function in this study. The ρ is the weighting coefficient, and ε is the relaxation factor. It can constrain the increments of the control vector directly, and it can also prevent the algorithm from obtaining an insoluble solution. In Equation (43), the state model $\eta(k)$, reference state model $\eta_{ref}(k)$, and the actual increments of the control vector $\Delta U(k)$ are addressed. Moreover, $J(k)$ will be transformed into a QP form as in Equation (44).

$$Y\frac{1}{2}x^T Hx + f^T x,\ x = [\Delta U\ \varepsilon]^T \tag{44}$$

The purpose of the MPC design is to minimize the tracking error, and the determination of weighting matrices Q_n and R_n is crucial to the MPC performance. However, parameter tuning is difficult and scenario oriented. It usually relies on empirical knowledge and trial-and-error methods. Hence, such a parameter tuning procedure is time consuming and inefficient. To solve this problem, this paper proposes a reinforcement learning-based MPC (RLMPC) controller to generate certain MPC parameters.

The concept of applying RL is simple, and a typical RL system is formed with the interaction of an agent and an environment. The RL training framework is shown in Figure 3, and $\pi(a_t|S_t)$ is the policy that can determine which action a_t could be applied according to the observed state S_t. The reward function R_t evaluates the rewards for the action applied to the RL. The agent is desired to react with the environment to obtain a higher reward by updating the policy.

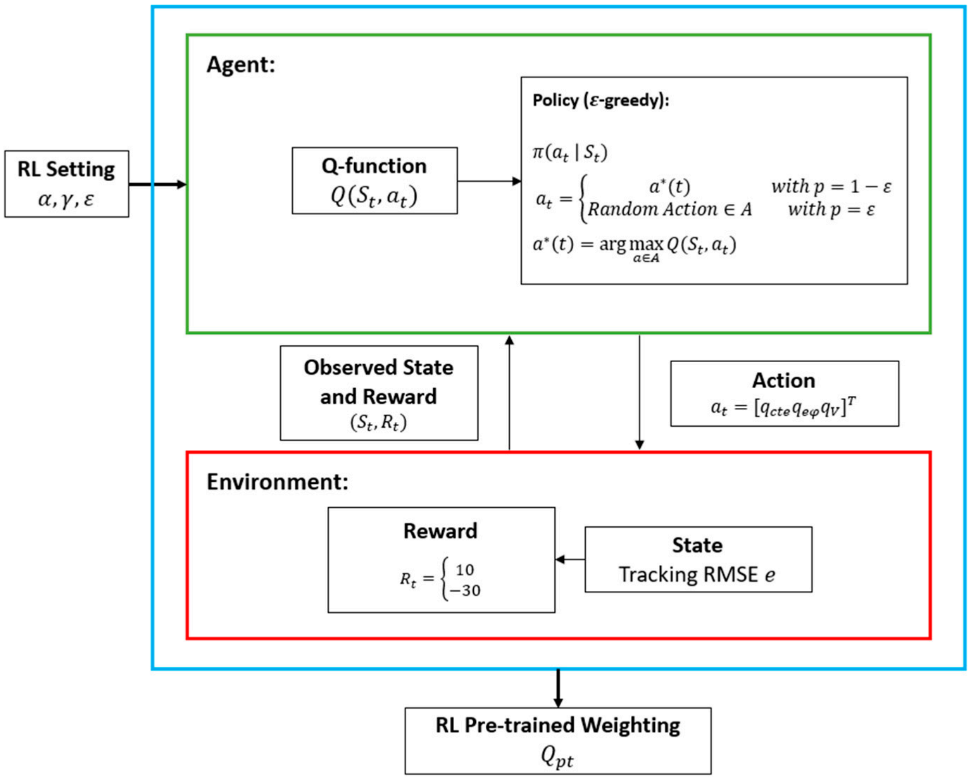
RL Setting
$\alpha, \gamma, \varepsilon$
Agent:
Q-function
$Q(S_t, a_t)$
Policy (ε-greedy):
$\pi(a_t \mid S_t)$
$a_t = \begin{cases} a^*(t) & with\ p = 1-\varepsilon \\ Random\ Action \in A & with\ p = \varepsilon \end{cases}$
$a^*(t) = \arg\max_{a \in A} Q(S_t, a_t)$
Observed State and Reward
(S_t, R_t)
Action
$a_t = [q_{cte} q_{e\varphi} q_v]^T$
Environment:
Reward
$R_t = \begin{cases} 10 \\ -30 \end{cases}$
State
Tracking RMSE e
RL Pre-trained Weighting
Q_{pt}

Figure 3. Proposed RL framework.

$Q(S_t, a_t)$ is a component of the agent, and it updates in each iteration. By applying the Markov decision process (MDP) Equation (45), a Q-function can estimate the future state and reward of the system with the current state and action. As a consequence, the updated $Q(S_t, a_t)$ is indicated in Equation (46). The iteration process is further applied for the generation of optimal weighting of the RLMPC. The operation of the proposed RLMPC is to generate a datum value of $cte(k)$, $e\varphi(k)$, and $v(k)$, and the rest of the parameters remain manually tuned to reduce the RL complexity. The definitions of state, action, and reward for the proposed RLMPC are shown in Equation (47). The relative RL setting is shown in Table 1, and α, ε, and γ are denoted as the learning rate, greedy index, and discount rate, respectively.

$$P(S_{t+1}|S_t) = P(S_{t+1}|S_1, S_2, \ldots, S_t) \tag{45}$$

$$Q^*(S_t, a_t) = Q(S_t, a_t) + \alpha\left(R_t + (\gamma \max_{a \in A} Q(S_{t+1}, a_{t+1})) - Q(S_t, a_t)\right) \tag{46}$$

$$\begin{gathered}\text{State} = \begin{cases} 0.895\ \text{m} \le e \le 0.91\ \text{m} & R_t = 10 \\ else & R_t = -30 \end{cases}, \\ \text{Action} = \begin{cases} q_V = 700 \sim 1000 & \pm 100 \\ q_{cte} = 600 \sim 1100 & \pm 100 \quad \Rightarrow 144\ \text{Actions} \\ q_{e\varphi} = 600 \sim 1100 & \pm 100 \end{cases}\end{gathered} \tag{47}$$

Table 1. RL Parameter Setting.

α-Parameter	ε-Parameter	γ-Parameter	Training Episodes
0.5	0.2	0.95	2000

Furthermore, the constraints of the control vector and control increments are designed by the Kronecker product in the form indicated in Equations (48) and (49).

$$u_{min}(t+k) \preccurlyeq u(t+k) \preccurlyeq u_{max}(t+k),\ k = 0, 1, \ldots, N_c - 1 \tag{48}$$

$$\Delta u_{min}(t+k)\ \preccurlyeq\ u(t+k)\ \preccurlyeq\ \Delta u_{max}(t+k),\ k = 0, 1, \ldots, N_c - 1 \tag{49}$$

According to the performance limitation of the proposed EV, the constraints applied in this paper are:

$$0\ \text{m/s} \le v_r \le 1.25\ \text{m/s},\ -1.0\ \text{m/s}^2 \le a \le 1.0\ \text{m/s}^2,\ -17^\circ \le \delta_f \le 17^\circ \tag{50}$$

By combining Equations (41), (48) and (49), the optimized objective expression is defined in terms of applying the barrier interior point method (BIPM) with RL pretrained weighting matrices. After solving the QP problem in every time step, a series of control increments in the control horizon can be obtained, as shown in Equation (51).

$$\Delta U_t^* = [\Delta u_t^*,\ \Delta u_{+1t}^*, \cdots,\ \Delta u_{t+N_c-1}^*]^T \tag{51}$$

The first element of the control series in Equation (51) is the actual input increment of the system, as indicated in Equation (52).

$$u(t) = u(t-1) + \Delta u_t^* \tag{52}$$

In the next time step, the system predicts a new output according to the state and undergoes the optimization process for new control increments. This procedure iterates until the whole path tracking mission is finished. As a consequence, the abovementioned RLMPC iteration process is summarized in Figure 4.

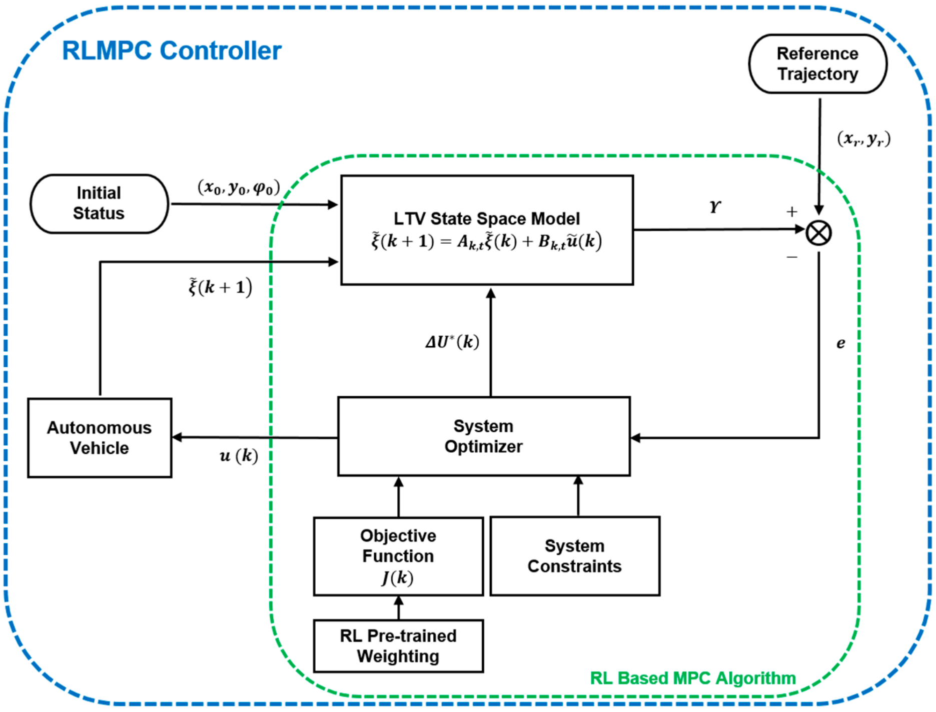

Figure 4. Proposed RL-based MPC (RLMPC) training framework.

4. Simulations and Experiments

To evaluate the performance of the proposed vehicle positioning and path tracking methods, simulations and experiments on a full-scale, laboratory-made EV were arranged, and they are organized in three subsections.

4.1. Simulation of RLMPC-Based Path Tracking

The simulations were arranged to evaluate the path tracking performance with respect to the manually tuned MPC and the RL-trained MPC methods. The trajectories of the abovementioned simulation results are shown in Figures 5 and 6. It is noted that the green line represents the calculated path in every MPC iteration. The weighting matrices of the manually tuned MPC and the RL-trained MPC are indicated in Equations (53) and (54), respectively. The vehicle started from (0,0) with a heading of 0 rad, and it attempted to track the trajectory following a line path equation of y = 2. The manual parameter tuning took time, and the trajectory exhibited an overshoot at the desired line path. On the other hand, with the RL, a proper MPC parameter, indicated in Equation (54), was obtained without a time-consuming trial-and-error process. It was apparent that the RLMPC tracked the line path well without overshoots.

$$Q_n = \mathrm{diag}(10\ 10\ 5\ 1000\ 5000\ 1500),\ R_n = \begin{bmatrix} 40 & 100 \end{bmatrix}^T \tag{53}$$

$$Q_{pt} = \mathrm{diag}(10\ 10\ 5\ 800\ 1100\ 700),\ R_{pt} = \begin{bmatrix} 40 & 100 \end{bmatrix}^T \tag{54}$$

Finally, the RL performance is also indicated in Figure 7. As the learning iteration accumulated, the successful counts increased. Because there were only two states in this training procedure, this study raised the learning rate for fast convergence. The training result with the parameters shown in Table 1 is indicated in Equation (54).

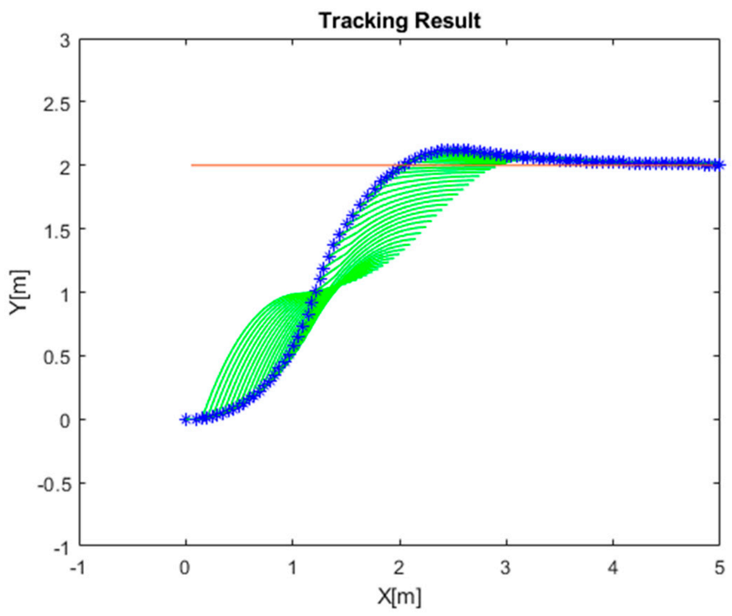

Figure 5. Line path tracking with manually tuned MPC parameters.

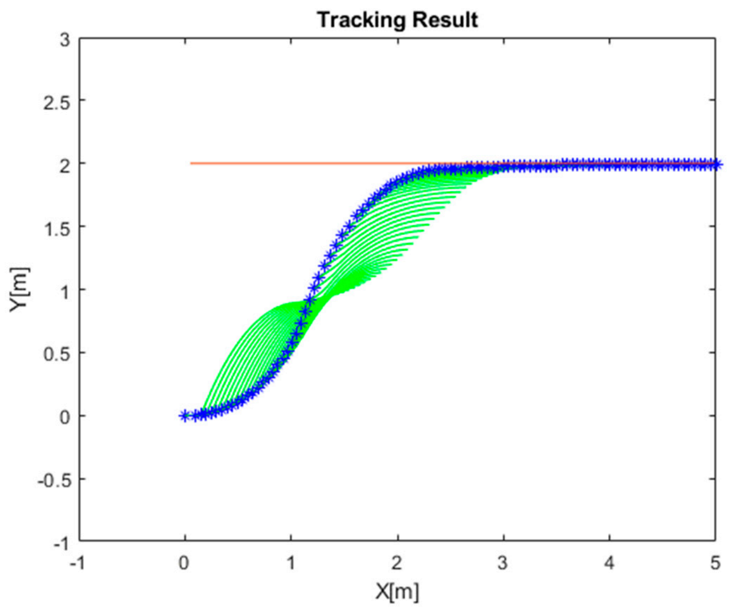

Figure 6. Line path tracking with RL-trained MPC parameters.

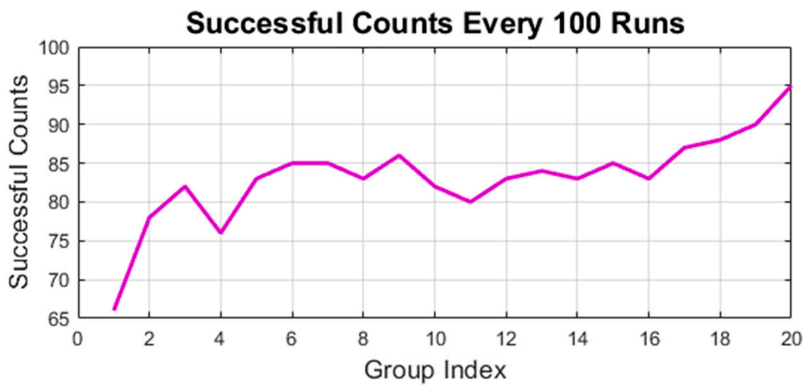

Figure 7. RL learning performance.

4.2. Validation of Estimated Distance with Position Estimation

The second experiment was arranged to evaluate the UKF-based vehicle positioning system, and a rectangular path on the NTUST campus was arranged for validation, as shown in Figure 8. It is noted that, due to the campus driving speed and route limitation of the small campus of NTUST (90,000 m^2), the following experiments were arranged simply. However, such a simple test environment was sufficient to examine the feasibility of integrating the aforementioned two technical aspects.

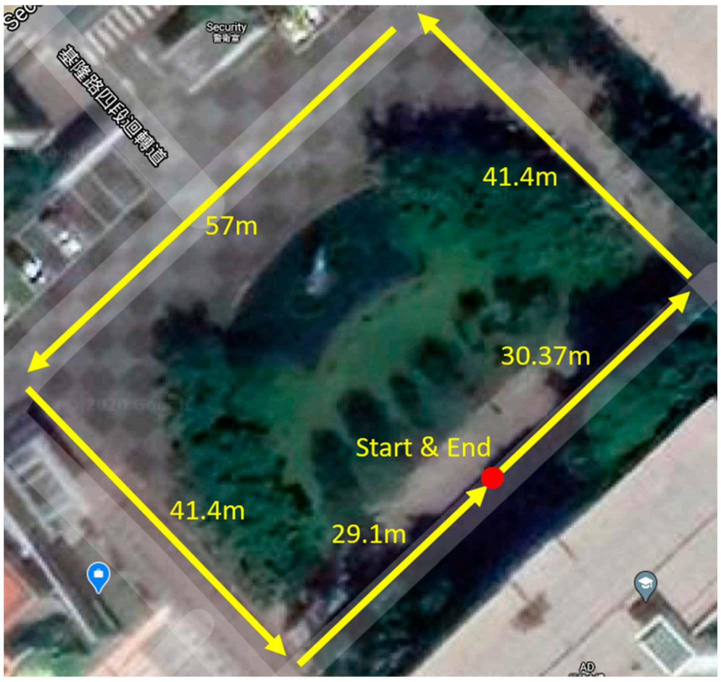

Figure 8. A rectangular path on the NTUST campus for vehicle positioning validation.

The RTK-GPS position obtained from the GGA instruction needed to be projected from the WGS84 system to the Cartesian coordinate system. Because the validation region in this paper was small, the equirectangular projection method was applied. The total distance of the rectangular path was approximately 199.37 m. The comparison of the traveling distance and the trajectory of odometry, RTK-GPS, and UKF are shown in Table 2 and Figure 9. Moreover, the comparison of odometry and the RTK and UKF methods are shown in Table 3.

Table 2. Comparison of Travel Distances Using Odometry, RTKGPS, and UKF.

Methods	Ground Truth (m)	Estimated Distance (m)	Error (%)	Mean Error (m)	Standard Deviation (m)
Odometry	199.27	192.882	3.205	6.387	4.856
RTKGPS	199.27	197.462	1.684	3.356	2.243
UKF	199.27	198.201	0.826	1.646	1.198

Table 3. Performance Comparison with literature [10,14].

	Proposed Method	Wei [10]	Mikov [14]
Ground Truth (m)	199.27	674.5	10,352 ± 3749
Estimated Distance (m)	198.201	683.18	10,404 ± 3777
Error (%)	0.826	1.29	0.502 ± 0.746
Mean Error (m)	1.646	1.23	-
Standard Deviation (m)	1.198	1.05	28

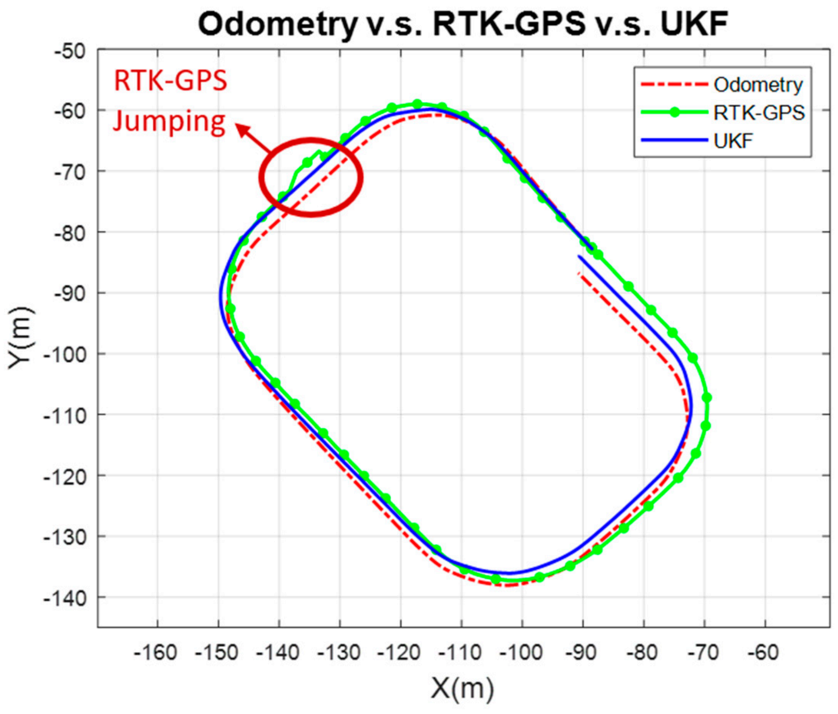

Figure 9. Trajectory comparison of odometry, RTK, and UKF.

The experimental results showed that the error of the UKF position estimator was 0.826%, as evaluated by the ground truth measured by the authors. Table 2 also indicates that the UKF represented the minimum error, mean error, and standard deviation when compared to odometry and RTK. Moreover, Table 3 shows a performance comparison with the state-of-the-art methods [10,14]. Our purpose was to know the performance of our UKF localization. We evaluated if the error percentage produced by the UKF was superior to other approaches. Although the experimental range and path were not the same, the proposed UKF demonstrated acceptable mean error and standard deviation.

Finally, Figure 9 shows the trajectory comparison of odometry and the RTK-GPS and UKF methods. It is also worth noting that on the top left of Figure 9 the RTK-GPS exhibits an outward jump. However, the UKF position estimator remained quite stable. The UKF position estimator also reduced the accumulative error that occurred when using odometry only.

4.3. Integrated Experiment with EV by Applying RL-Based MPC

The path can be any combined planner equations. For example, the test scenario 2 was composed of four segments. The reference path was formed by recording the trajectory of manual driving. The recorded trajectory was manually processed in terms of driving as four segments, and each segment was further represented as an equation in terms of the curve fitting approach. Such combined equations are available to be tracked in terms of RL MPC.

In this experiment, an EV was used for trajectory tracking based on human-tuned and RLMPC control schemes. Two experimental scenarios were arranged on the NTUST campus: (1) a straight-line path and (2) a combinational path. It is noted that the straight-line path of scenario 1 is indicated as in Equation (55) and the combinational path of scenario 2 is indicated in Equation (56). For the combinational path in Equation (56), a smoothing spline was utilized to obtain a piecewise linear function with four intervals (i = 1 to 4). The weight was set as $w_i = 1$, and the corresponding smoothing parameter for each interval is indicated in Equation (57).

$$y(x) = -1.095x - 260.7 \tag{55}$$

$$p\sum_i w_i(y_i - s(x_i))^2 + (1-p)\int(\frac{d^2s}{dx^2})^2dx \tag{56}$$

$$p = \begin{cases} 0.999 & \text{, for interval 1} \\ 0.991 & \text{, for interval 2} \\ 0.769 & \text{, for interval 3} \\ 0.763 & \text{, for interval 4} \end{cases} \tag{57}$$

$$Q_n = \text{diag}(10\ 10\ 5\ 900\ 2100\ 1000),\ R_n = \begin{bmatrix} 40 & 100 \end{bmatrix}^T \tag{58}$$

Mechanism tolerance, hardware limitations, and other factors might influence practical implementation. This work applied a pre-trained weighting matrix, shown in Equation (54), as the datum value of a full-scale EV experiment. Based on empirical knowledge and the pre-trained datum value of the weighting matrix, it can significantly reduce the parameter tuning time. The practical weighting matrices Q_n and R_n were further revised for RLMPC, as indicated in Equation (58). The weighting matrices applied in non-RLMPC that were tuned by the operator were the same as the simulation case indicated in Equation (53).

For scenario 1 experiments, the path tracking results of MPC and RLMPC are shown in Figure 10, and the tracking errors of MPC and RLMPC are indicated in Figure 11. The line path tracking results were quite similar to the aforementioned simulation results shown in Figures 5 and 6. The human-tuned MPC represented some oscillation when the EV reached the line path. Nevertheless, the RLMPC exhibited a smaller error after the 70th sample.

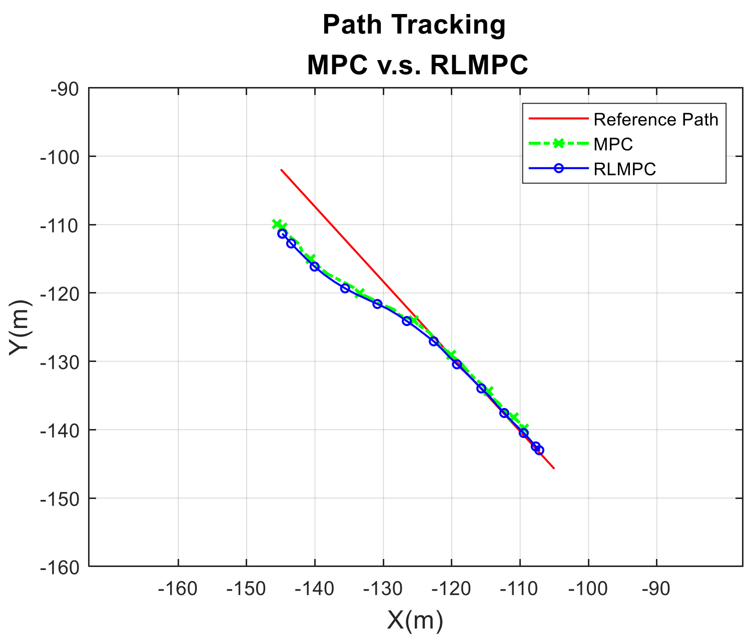

Figure 10. Trajectory comparison of MPC and RLMPC in scenario 1.

For the scenario 2 experiments, the path tracking results of MPC and RLMPC are shown in Figure 12, and the tracking errors of MPC and RLMPC are indicated in Figure 13. It was apparent that the RLMPC outperformed the tracking error compared to the human-tuned MPC. To provide a confident and quantitative error evaluation, all the experiments were performed three times for the performance comparison, as indicated in Table 4. Table 4 shows the relative statistical data of averaging the values of the three trials. Both of the average RMSEs were less than 0.3 m, and the maximum errors were less than 0.7 m.

The overall results showed that the RLMPC and human-tuned MPC followed the same trajectory well. However, with well-converged parameters, RLMPC had better performance than MPC tuned by humans in terms of maximum error, average error, standard deviation, and RMSE.

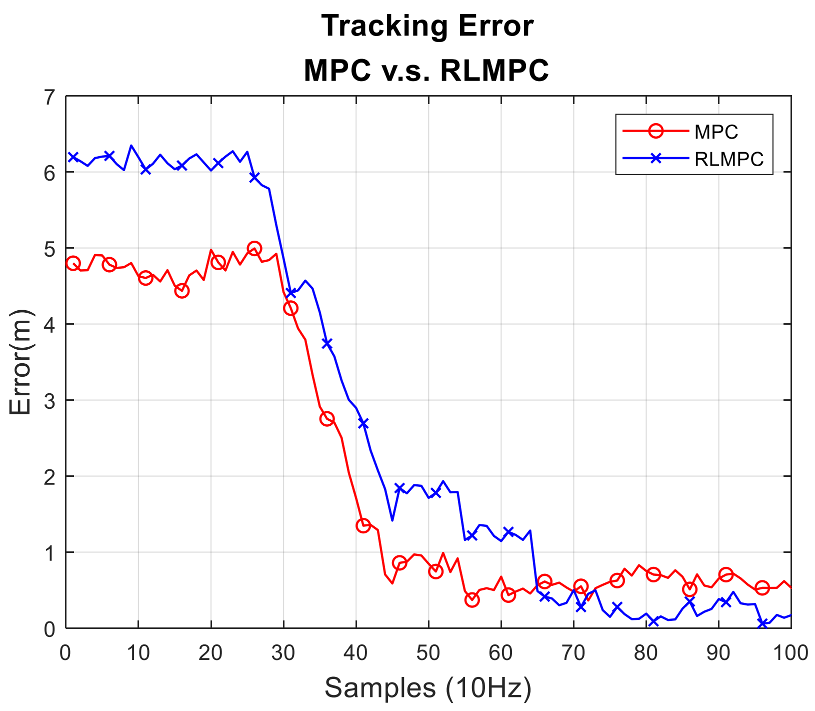

Figure 11. Tracking error comparison of MPC and RLMPC in Scenario 1.

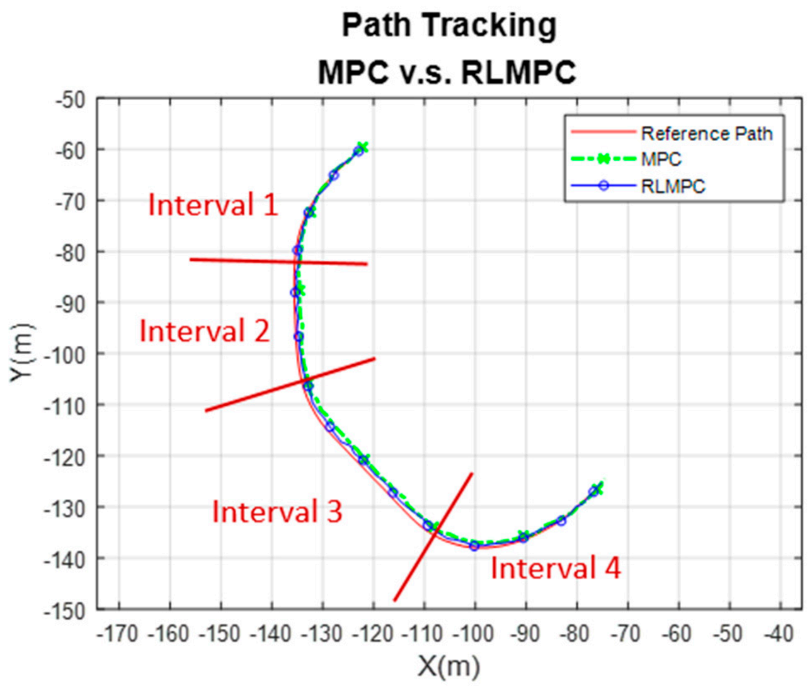

Figure 12. Trajectory comparison of MPC and RLMPC in scenario 2.

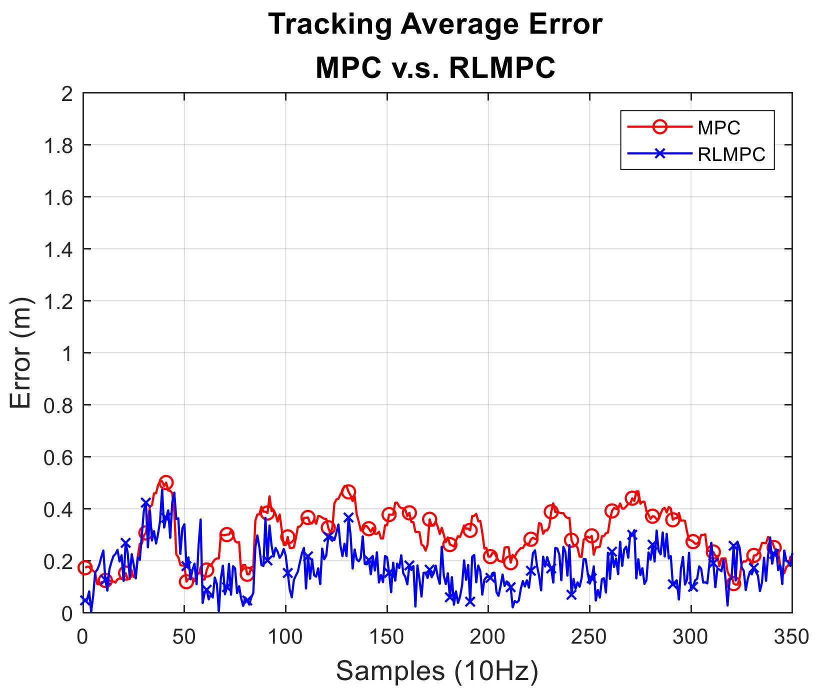

Figure 13. Tracking error comparison of MPC and RLMPC in Scenario 2.

Table 4. Comparison of Path Tracking Performance of Scenario 2.

Method	Maximum Error (m)	Average Error (m)	Standard Deviation (m)	RMSE (m)
MPC	0.671	0.291	0.138	0.257
RLMPC	0.615	0.196	0.112	0.227

5. Conclusions and Future Works

In this paper, a reinforcement learning-based MPC framework is presented. The proposed RLMPC significantly reduced the efforts of tuning MPC parameters. The RLMPC executed with the UKF-based vehicle positioning system that considered the RTK, odometry, and IMU sensor data. The proposed UKF vehicle positioning and RLMPC path tracking methods were validated with a full-scale, laboratory-made EV on the NTUST campus. On a 199.27 m loop path, the UKF estimated travel distance error was 0.82%. The MPC parameters generated by RL achieved an RMSE of 0.227 m in the path tracking experiments, and it also exhibited better tracking performance than the human-tuned MPC parameters.

Moreover, the aim of this work was to integrate two important practices of realizing an autonomous vehicle in a campus environment, including vehicle positioning and path tracking. Such a project is helpful to students in university to easily reach, learn, and practice key technologies of autonomous vehicles. As a consequence, this work was not aiming at providing significant improvement on the localization accuracy or RL MPC performance. Hence, the future works on the localization accuracy and RL MPC performance in terms of two independent projects will be studied based on the laboratory-made electric vehicle and the preliminary localization and path tracking results. For example, the use of robust MPC based on safe reinforcement learning [31] would be an important direction to improve the path tracking performance. Moreover, due to the campus driving speed and route limitation of the small campus of NTUST (90,000 m^2), this paper just showed the preliminary results on the successful integration of vehicle localization and path tracking. Moving the vehicle to a more complicated test environment would be another important future work.

On the other hand, obstacle avoidance was not a purpose of this work. The 360° scan formed with two lidar devices temporarily stopped the driving of EVs during path tracking when any obstacles were detected to ensure safety. In the future, an effective reactive obstacle avoidance approach will be further considered to improve autonomous driving efficiency in a dynamic obstacle environment. Moreover, the proposed UKF can further combine 3D-LiDar and visual SLAM to improve the vehicle positioning robustness. On the other hand, only three components in Q_n were considered for RL in this paper. In the future, all parameters in this matrix will be considered and evaluated.

Author Contributions: J.-A.Y. was responsible for hypothoses, system deployment, experiments, and paper draft writing; C.-H.K. is the advisor who oriented the research direction for the paper and gave comments and advice to do this research. All authors have read and agreed to the published version of the manuscript.

Funding: Ministry of Science and Technology, Taiwan under Grant MOST 109-2221-E-011-112-MY3 and the "Center for Cyber-physical System Innovation" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

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

References

1. Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), Lake Louise, AB, Canada, 4 October 2000; pp. 153–158.
2. Rose, C.; Britt, J.; Allen, J.; Bevly, D.M. An Integrated Vehicle Navigation System Utilizing Lane-Detection and Lateral Position Estimation Systems in Difficult Environments for GPS. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2615–2629. [CrossRef]
3. Um, I.; Park, S.; Oh, S.; Kim, H. Analyzing Location Accuracy of Unmanned Vehicle According to RTCM Message Frequency of RTK-GPS. In Proceedings of the 2019 25th Asia-Pacific Conference on Communications (APCC), Ho Chi Minh City, Vietnam, 6–8 November 2019; pp. 326–330.
4. Kim, J.W.; Lee, D.K.; Bin Cao, S.; Lee, S.S. Availability evaluation and development of network-RTK for vehicle in downtown. In Proceedings of the 2014 International Conference on Information and Communication Technology Convergence (ICTC), Busan, Korea, 22–24 October 2014; pp. 557–558.
5. Liu, T.; Li, B. Single-frequency BDS/GPS RTK with low-cost U-blox receivers. In Proceedings of the 2017 Forum on Cooperative Positioning and Service (CPGPS), Harbin, China, 19–21 May 2017; pp. 232–238.
6. Ng, K.M.; Johari, J.; Abdullah, S.A.C.; Ahmad, A.; Laja, B.N. Performance Evaluation of the RTK-GNSS Navigating under Different Landscape. In Proceedings of the 2018 18th International Conference on Control, Automation and Systems (ICCAS), PyeongChang, Korea, 17–20 October 2018; pp. 1424–1428.
7. Um, I.; Park, S.; Kim, H.T.; Kim, H. Configuring RTK-GPS Architecture for System Redundancy in Multi-Drone Operations. *IEEE Access* **2020**, *8*, 76228–76242. [CrossRef]
8. Maier, D.; Kleiner, A. Improved GPS sensor model for mobile robots in urban terrain. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 4385–4390.
9. Yi, S.; Worrall, S.; Nebot, E. Geographical Map Registration and Fusion of Lidar-Aerial Orthoimagery in GIS. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 128–134.
10. Wei, L.; Cappelle, C.; Ruichek, Y.; Zann, F. Intelligent Vehicle Localization in Urban Environments Using EKF-based Visual Odometry and GPS Fusion. *IFAC Proc. Vol.* **2011**, *44*, 13776–13781. [CrossRef]
11. Aftatah, M.; Lahrech, A.; Abounada, A. Fusion of GPS/INS/Odometer measurements for land vehicle navigation with GPS outage. In Proceedings of the 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), Marrakech, Morocco, 24–26 May 2016; pp. 48–55. [CrossRef]
12. Henkel, P.; Sperl, A.; Mittmann, U.; Bensch, R.; Farber, P. Precise Positioning of Robots with Fusion of GNSS, INS, Odometry, LPS and Vision. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–6.
13. Mohamed, S.A.S.; Haghbayan, M.-H.; Westerlund, T.; Heikkonen, J.; Tenhunen, H.; Plosila, J. A Survey on Odometry for Autonomous Navigation Systems. *IEEE Access* **2019**, *7*, 97466–97486. [CrossRef]
14. Mikov, A.; Panyov, A.; Kosyanchuk, V.; Prikhodko, I. Sensor Fusion for Land Vehicle Localization Using Inertial MEMS and Odometry. In Proceedings of the 2019 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL), Naples, FL, USA, 1–5 April 2019; pp. 1–2. [CrossRef]
15. Zhao, S.; Chen, Y.; Farrell, J.A. High-Precision Vehicle Navigation in Urban Environments Using an MEM's IMU and Single-Frequency GPS Receiver. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2854–2867. [CrossRef]
16. Chen, M.; Ren, Y. MPC based path tracking control for autonomous vehicle with multi-constraints. In Proceedings of the 2017 International Conference on Advanced Mechatronic Systems (ICAMechS), Xiamen, China, 6–9 December 2017; pp. 477–482.

17. Paden, B.; Cap, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [CrossRef]
18. Jhang, J.-H.; Lian, F.-L. An Autonomous Parking System of Optimally Integrating Bidirectional Rapidly-Exploring Random Trees* and Parking-Oriented Model Predictive Control. *IEEE Access* **2020**, *8*, 163502–163523. [CrossRef]
19. Shen, C.; Guo, H.; Liu, F.; Chen, H. MPC-based path tracking controller design for autonomous ground vehicles. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 9584–9589.
20. Tang, W.; Yang, M.; Wang, C.; Wang, B.; Zhang, L.; Le, F. MPC-Based Path Planning for Lane Changing in Highway Environment. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 1003–1008.
21. Chen, Y.; Wang, J. Trajectory Tracking Control for Autonomous Vehicles in Different Cut-in Scenarios. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 4878–4883.
22. Dixit, S.; Montanaro, U.; Dianati, M.; Oxtoby, D.; Mizutani, T.; Mouzakitis, A.; Fallah, S. Trajectory Planning for Autonomous High-Speed Overtaking in Structured Environments Using Robust MPC. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 2310–2323. [CrossRef]
23. Liniger, A.; Domahidi, A.; Morari, M. Optimization-based autonomous racing of 1:43 scale RC cars. *Optim. Control Appl. Methods* **2015**, *36*, 628–647. [CrossRef]
24. Rosolia, U.; Carvalho, A.; Borrelli, F. Autonomous racing using learning Model Predictive Control. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5115–5120.
25. Lin, F.; Chen, Y.; Zhao, Y.; Wang, S. Path tracking of autonomous vehicle based on adaptive model predictive control. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1–12. [CrossRef]
26. Görges, D. Relations between Model Predictive Control and Reinforcement Learning. *IFAC-PapersOnLine* **2017**, *50*, 4920–4928. [CrossRef]
27. Xie, H.; Xu, X.; Li, Y.; Hong, W.; Shi, J. Model Predictive Control Guided Reinforcement Learning Control Scheme. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
28. Guan, Y.; Ren, Y.; Li, S.E.; Sun, Q.; Luo, L.; Li, K. Centralized Cooperation for Connected and Automated Vehicles at Intersections by Proximal Policy Optimization. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12597–12608. [CrossRef]
29. Tange, Y.; Kiryu, S.; Matsui, T. Model Predictive Control Based on Deep Reinforcement Learning Method with Discrete-Valued Input. In Proceedings of the 2019 IEEE Conference on Control Technology and Applications (CCTA), Hong Kong, China, 19–21 August 2019.
30. Modalavalasa, H.K.; Makkena, M.L. Intelligent Parameter Tuning Using Segmented Adaptive Reinforcement Learning Algorithm. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 252–256.
31. Zanon, M.; Gros, S. Safe Reinforcement Learning Using Robust MPC. *IEEE Trans. Autom. Control* **2021**, *66*, 3638–3652. [CrossRef]