

RESEARCH ARTICLE

Trajectory Optimization and Robust Tracking Control for Off-Road Autonomous Vehicle

SEONGIL HONG  **AND GYUHYUN PARK**

AI and Autonomy Technology Center, Advanced Defense Science and Technology Research Institute, Agency for Defense Development, Daejeon 34186, South Korea

Corresponding author: Seongil Hong (science4729@gmail.com)

This work was supported by the Agency for Defense Development.

ABSTRACT This paper presents a control strategy for real-time trajectory optimization and robust path tracking for unmanned off-road vehicles to ensure both stability and performance. The approach takes advantage of a two-degree-of-freedom control framework that combines predictive driving control through perceptual information and feedback control for robust stability. Trajectory generation leverages model predictive control where the particle swarm optimization is used as an optimizer to address problems of the non-smoothness of the traversability information and nonlinear nature of vehicle dynamics. By using the exteroceptive perception, the vehicle could estimate traversability and adapt its motion to achieve fast and smooth driving. For the feedback controller, a system level synthesis is used to faithfully track the planned path despite uncertainty and unknown disturbance. Specifically, we focus on realizing the proposed method as a practical means for the real-time control system. The effectiveness of this method is validated through extensive numerical simulations and experimental tests, demonstrating its practical applicability in uncertain environments for autonomous vehicle navigation.


INDEX TERMS Autonomous vehicle, linear quadratic regulator, model predictive control, vehicle dynamics, convex optimization, robust control, system level synthesis.

I. INTRODUCTION

In recent years, the field robotics has seen remarkable advancements, leading to a shift in the application of robots towards exploration in hostile, uncooperative, and extreme environments. A significant body of research in car-like robots concentrates on operating in complex urban areas, as evidenced by well-known and successful technologies [1], [2], [3], [4], [5], [6], [7], [8]. However, the extraordinary progress in autonomous vehicles also presents a substantial opportunity for deployment in more unstructured terrains, including subterranean [9], agricultural [10], planetary exploration [11], transportation [12], search and rescue operations [13], and military applications [14], [15]. In particular, unmanned vehicles designed for off-road use face the challenge of navigating safely while adapting to the unpredictable nature of unstructured environments. Thus, the

planning and control problems become more difficult, and the failure of these components could have serious consequences, potentially resulting in the loss of human life.

Contemporary control methodologies for autonomous vehicles have largely inherited from the foundational work established during the DARPA Grand and Urban Challenges [16], [17], [18]. These control methods employ a strategy composed of two different and separate subproblems: online trajectory planning and trajectory tracking. First, a local path satisfying driving constraints is generated. Subsequently, this predetermined path is fed into a low-level feedback control law, which calculates the necessary steering and throttle commands to track the path. The planner generates a feasible trajectory based on the kinematic or dynamic models. However, the design of the path tracker, which employs nonlinear or PID control laws, does not take into account robust stability or the model characteristics used by the planner. If the path tracking controller results in a relatively large control error, then the dynamic and

The associate editor coordinating the review of this manuscript and approving it for publication was Salman Ahmed .

kinematic models of the planner are no longer valid. These problems occur more often when driving on an off-road than on a paved road. Thus, the trajectory planner should be tightly integrated with the path tracker by sharing the same characteristics. Advanced path tracking control methods are proposed, most of the algorithms are working in a paved road, and the performance verification is conducted in numerical simulations [19], [20], [21], [22], small-scale vehicle tests [8], and hardware-in-the-loop simulations [23], [24], [25]. Because of the complex interactions between the unpaved road and the vehicle, performance verification through experiment with the full-scale vehicle is crucial.



FIGURE 1. Unmanned ground vehicle.

Recently, model predictive control (MPC) has been widely used for controlling general nonlinear dynamical systems [26], [27] as well as autonomous vehicles [28]. Many of the autonomous systems employ gradient-based MPC approaches, where the cost function is usually smooth [7], [29], [30]. Presently, off-road navigation systems often utilize LiDAR sensors to acquire a three-dimensional point cloud for the semantic and geometric analysis of the environment [15], [31]. Although LiDAR sensors deliver precise spatial data, the generated point cloud is not dense, and consequently the resulting traversability map is also sparse and discrete. Therefore, the gradient-based optimization algorithm becomes challenging to solve such non-smooth optimization problems that arise from perception data.

Sampling-based path planning or stochastic optimal control method [32], [33] can be promising solutions because it is able to consider the discontinuous cost function and do not require any smooth linear dynamics and quadratic approximations of the cost function. However, the performance verification of real-time control systems, such as a full-scale autonomous vehicle, has only recently been demonstrated [34]. As the control periods of trajectory planning or MPC are usually long due to the massive numerical optimization and perception-based control, a path tracker is needed to mitigate an adversarial effect of a fast disturbance [35].

Fig. 1 displays the configuration of the unmanned vehicle to be controlled in this paper, which has three LiDAR sensors installed at the front. We delve into the complexities of developing control systems for the full-scale, off-road autonomous vehicle. To address these challenges, a novel two-degree-of-freedom (TDF) control strategy is introduced, which integrates MPC with a robust feedback controller in a hierarchical and cohesive manner, based on the shared dynamics [35], [36], [37], [38]. Fig. 2 schematically illustrates the structure, which embodies a layered control architecture [39]. This control approach not only accommodates the unpredictable environment of off-road navigation but also ensures stability against uncertainty and disturbance. The MPC is responsible for generating optimal trajectories and feedforward control input that consider the vehicle’s dynamics and the future traversability on a rough terrain, while the fast feedback controller guarantees that the vehicle remains stable and adheres to the planned path.

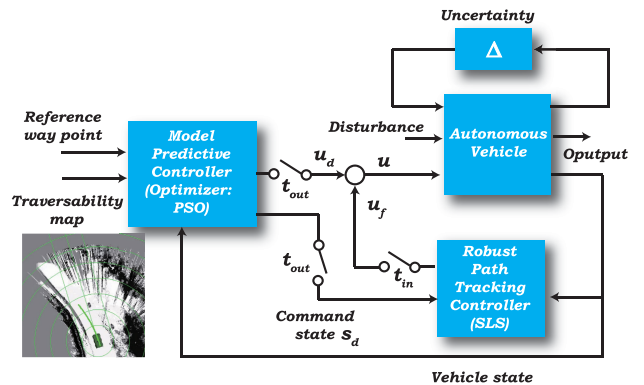


FIGURE 2. The two-degree-of-freedom control structure. The model predictive control is used to generate state trajectory \mathcal{S}^* , the command state $s_d \in \mathcal{S}^*$, and the reference control input u_d . The particle swarm optimization (PSO) is employed as the optimizer of MPC. The path tracking controller is designed based on the system level synthesis (SLS) framework to guarantee that the vehicle remains stable and adheres to the planned path with the control input u_f .

In Fig. 2, the “outer loop” feedback path or MPC generates the optimal state trajectory $\mathcal{S}^* = \{s^*(0), s^*(1), \dots, s^*(T)\}$, and control sequence $\mathcal{A}^* = \{a^*(0), a^*(1), \dots, a^*(T-1)\}$ from the current state over a finite time horizon T . The reference state command, $s_d \in \mathcal{S}^*$, and a control input, $u_d := a^*(0)$, are used for a control period of time t_{out} . Then new s_d and u_d are recomputed as soon as new information about the environment and vehicle state is available. To address problems of the non-smoothness of the traversability map and nonlinear nature of vehicle dynamics, a particle swarm optimization (PSO) is employed as the optimizer of MPC. In order to make the problem more numerically tractable, a local solution is obtained by parameterizing control trajectory with trigonometric curves. We acquire the state trajectory \mathcal{S} by integrating the equations of motion from the initial state $s(0)$ and the control trajectory \mathcal{A} . Therefore, the dynamic or kinematic constraints are unnecessary and state-control trajectories are always strictly feasible even if the

optimization is terminated prematurely due to the limited computation time.

If the controller reduces the tracking error sufficiently, we can linearize the nonlinear system and then design the “inner loop,” tracking controller. The robust path tracking controller, depicted in Fig. 2, stabilizes the errors on the fastest time scales, $t_{in} \leq t_{out}$, with a control input u_f . Here, path tracking refers to a vehicle following a locally generated path by applying appropriate steering motions that guide the vehicle along the optimal trajectory \mathcal{S}^* . We leverage the *system level synthesis* (SLS) framework to transform the controller design process from synthesizing the controller itself to designing the entire closed-loop system [40], [41]. This framework enables us to exploit convexity for numerical optimization and provides transparency for robust control by measuring the degradation in performance of the controller due to uncertainty [42]. In this study, we summarize the core principle of the SLS algorithm and then we emphasize the underlying mathematical implications for a practical application. Due to the transfer functions of the SLS framework, the matrix size of the controller is large to be efficiently implemented on a small ruggedized on-board computer. Thus, we use an equivalent SLS controller with a constant gain matrix.

Our main contributions are summarized as follows. First, for the off-road autonomous vehicle, we take advantage of the TDF controller based on the shared dynamics. Thus, we obtain the predictive driving control when perceptual information is given and simultaneously attains robustness via the fast feedback controller. The robust performance is validated for the unmanned vehicle to successfully execute the navigation missions in highly uncertain environments. Second, this paper is the first to experimentally verify the effectiveness of the state-feedback SLS framework for an autonomous system as far as the authors know. We make use of two different SLS controller, one of which has the transfer matrix, while the other has the constant gain matrix for efficient control input computation. The robust performance of the SLS is verified compared to that of the LQR controller. We also investigate the validity of the dynamic and kinematic models because the performance of the MPC and the path tracking controller heavily depends on these models. Third, particular emphasis is placed on the practical application of the optimization-based MPC as a viable solution for real-time control systems. The state-control solution trajectories are always feasible and smooth by the parameterization of the control sequence with smooth functions.

The remainder of this article is organized as follows: In Section II, we present the vehicle models. Section III and IV design the inner and outer feedback loops of the TDF controller, respectively. The performance and robustness are demonstrated through extensive numerical simulations and experimental tests in Section V. The experimental results are also presented in the supplementary video. We investigate the validity of the kinematic and dynamic models. Finally, Section VI draws conclusions.

II. VEHICLE MODELS FOR STEERING CONTROL

We first present the nonlinear vehicle dynamic and kinematic models and derive the linearized tracking dynamics, which will be used for the outer and inner loop controller design, respectively.

A. BICYCLE MODEL

Consider the unmanned vehicle with a fixed middle and rear axles and a set of front wheels that can be rotated, as illustrated in Fig. 3. We denote by subscripts x and y , the longitudinal and lateral coordinates, respectively, in the body frame O_{xyz} . The subscripts f , m , and r denote “front”, “middle”, and “rear”, respectively. Let l_f , l_m and l_r be the distances from center of mass to each wheels. The longitudinal, lateral, and rotational dynamic equations are

$$m(\dot{v}_x - v_y\dot{\theta}) = F_{xf} \cos \tau - F_{yf} \sin \tau + F_{xm} + F_{xr}, \quad (1a)$$

$$m(\dot{v}_y + v_x\dot{\theta}) = F_{xf} \sin \tau + F_{yf} \cos \tau + F_{ym} + F_{yr}, \quad (1b)$$

$$I_z\ddot{\theta} = l_f(F_{xf} \sin \tau + F_{yf} \cos \tau) - l_m F_{ym} - l_r F_{yr}, \quad (1c)$$

where m is the vehicle mass, and F is the tire force. θ is the vehicle’s yaw angle about Z axis, and I_z is the moment of inertia [1], [43]. v_x and v_y are longitudinal and lateral velocities, respectively. The control variable is the wheel steering angle τ .

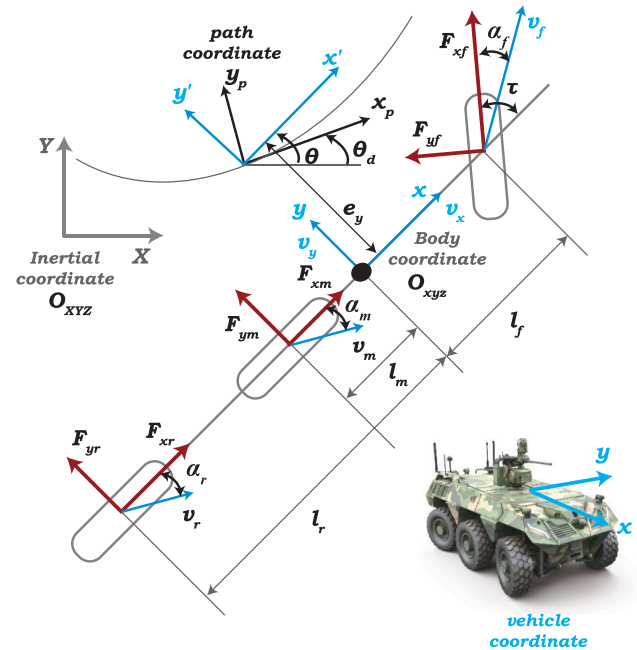


FIGURE 3. Vehicle steering dynamics.

Assume that a constant longitudinal velocity $\dot{v}_x = 0$ and longitudinal forces $F_{xf} = F_{xm} = F_{xr} = 0$ in (1). The lateral force generated by the wheels can be modeled as linearly proportional to the slip angle α formed between the velocity vector v and the centerline of the wheel as follows:

$$F_{yf} = c_f \alpha_f, \quad F_{ym} = c_m \alpha_m, \quad F_{yr} = c_r \alpha_r, \quad (2)$$

where the tire slip angles are

$$\alpha_f = \tau - \tan^{-1} \left(\frac{v_y + l_f \dot{\theta}}{v_x} \right), \quad (3a)$$

$$\alpha_{m,r} = \tan^{-1} \left(\frac{l_{m,r} \dot{\theta} - v_y}{v_x} \right). \quad (3b)$$

Substituting (2) into (1) and solving for \dot{v}_y and $\dot{\theta}$ gives the bicycle model. With a constant longitudinal velocity $\dot{v}_x = 0$, a small steering angle τ and slip angle α , we obtain the linearized dynamic model

$$\dot{v}_y = \frac{-(c_f + c_m + c_r)}{mv_x} v_y + \left[\frac{c_m l_m + c_r l_r - c_f l_f}{mv_x} - v_x \right] \dot{\theta} + \frac{c_f}{m} \tau, \quad (4a)$$

$$\ddot{\theta} = \frac{c_m l_m + c_r l_r - c_f l_f}{I_z v_x} v_y - \frac{c_f l_f^2 + c_m l_m^2 + c_r l_r^2}{I_z v_x} \dot{\theta} + \frac{c_f l_f}{I_z} \tau. \quad (4b)$$

The vehicle's motion in the inertial frame O_{XYZ} is

$$\dot{p}_X = v_x \cos \theta - v_y \sin \theta, \quad (5a)$$

$$\dot{p}_Y = v_x \sin \theta + v_y \cos \theta, \quad (5b)$$

where \dot{p}_X and \dot{p}_Y denote the east and west velocities of the vehicle's center of mass. The positions of p_X and p_Y are used to evaluate the traversability along the predicted state trajectories \mathcal{S} and to check for collisions to avoid obstacles when the perception result is given.

We can use the general equations of motion as follows:

$$\dot{s} = f_c(s, a). \quad (6)$$

If the state and control input are defined as $s = [p_X, p_Y, v_y, \theta, \dot{\theta}]^\top$, and $a = \tau$, then the function f_c is composed of (1b)-(1c) and (5) for the nonlinear lateral and yaw dynamic model. For the linear dynamic model, the function f_c consists of (4) and (5). The nonlinear kinematic model is

$$\dot{p}_X = v \cos(\theta + \beta), \quad (7a)$$

$$\dot{p}_Y = v \sin(\theta + \beta), \quad (7b)$$

$$\dot{\theta} = \frac{v}{l_r} \sin(\beta), \quad (7c)$$

$$\beta = \arctan \left(\frac{l_r}{l_f + l_r} \tan \tau \right), \quad (7d)$$

where β is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the car [44]. The function f_c is composed of (7), where $s = [p_X, p_Y, \theta]^\top$ and $a = \tau$.

B. TRACKING MODEL

Suppose that we are concerned with regulating the lateral position and orientation deviation of the vehicle from a reference state trajectory as depicted in Fig. 3. Let the

orientation error be $e_\theta = \theta - \theta_d$, where θ_d is a constant reference heading angle. The lateral velocity error with respect to the path is $\dot{e}_y = v_y \cos e_\theta + v_x \sin e_\theta$, which is approximated by $\dot{e}_y \approx v_y + v_x e_\theta$ for a small angle error e_θ . The assumption of the small angle approximation is experimentally verified in Section V. The small control error means that the TDF control works well, and then the small steering angle and slip angle in (4) can also be valid assumptions. The lateral acceleration error becomes $\ddot{e}_y = \dot{v}_y + v_x \dot{e}_\theta$, where the longitudinal velocity v_x is constant. The angular velocity and acceleration errors are $\dot{e}_\theta = \dot{\theta}$ and $\ddot{e}_\theta = \ddot{\theta}$, respectively. With (4) and the error states, we obtain the state space model of error dynamics by using

$$\dot{x} = A_c x + B_c u \quad (8)$$

where $x = [e_y, \dot{e}_y, e_\theta, \dot{e}_\theta]^\top$, $u = \tau$

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{c_f + c_m + c_r}{mv_x} & \frac{c_f + c_m + c_r}{m} & \frac{c_m l_m + c_r l_r - c_f l_f}{mv_x} - v_x \\ 0 & 0 & 0 & 1 \\ 0 & \frac{c_m l_m + c_r l_r - c_f l_f}{I_z v_x} & \frac{c_f l_f - c_m l_m - c_r l_r}{I_z} & -\frac{c_f l_f^2 + c_m l_m^2 + c_r l_r^2}{I_z v_x} \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0 & \frac{c_f}{m} & 0 & \frac{c_f l_f}{I_z} \end{bmatrix}^\top.$$

The discrete-time dynamics of continuous one is written as

$$x(k+1) = Ax(k) + Bu(k). \quad (9)$$

III. TRAJECTORY TRACKING CONTROL

The objective of the path tracking controller is to minimize the lateral distance error e_y between the vehicle position and the generated trajectory and the angle difference e_θ in the vehicle's heading and the defined path's heading by using the state space equation (9). At the same time, the controller should minimize steering input for a smooth motion. Therefore, the linear quadratic regulator (LQR) is used

$$\text{minimize}_{x(k), u(k)} \lim_{H \rightarrow \infty} \mathbb{E} \left[\frac{1}{H} \sum_{k=0}^H x(k)^\top Q x(k) + u(k)^\top R u(k) \right] \quad (10a)$$

$$\text{subject to } x(k+1) = Ax(k) + Bu(k) + w(k), \quad (10b)$$

where $x(k) \in \mathbb{R}^n$ is the state at time k , $u(k) \in \mathbb{R}^p$ is the control input, and $w(k) \in \mathbb{R}^n$ is an exogenous Gaussian disturbance with zero-mean and covariance σ . The state-weighting matrix $Q \in \mathbb{R}^{n \times n}$ is positive semi-definite and the control weighting $R \in \mathbb{R}^{p \times p}$ is positive definite. The time horizon H approaches infinity and the mean cost is minimized. For simplicity $u(H)$ is included in the cost function.

We employ lowercase and uppercase Latin letters such as x and A to represent vectors and matrices, respectively. Additionally, lowercase and uppercase boldface letters such as \mathbf{x} and $\Phi(z)$ are used to denote signals and transfer matrices,

respectively. The transfer function is obtained by taking the z -transform, i.e.,

$$\Phi(z) = \sum_{k=1}^{\infty} \phi(k)z^{-k}, \quad (11)$$

where the function $\phi(k)$ is the system response element or k th spectral component of a transfer function Φ . The Hardy spaces \mathcal{H}_2 and \mathcal{H}_∞ represent all proper and real-rational stable transfer matrices $\Phi(z)$ with $\|\Phi\|_{\mathcal{H}_2}$ and $\|\Phi\|_{\mathcal{H}_\infty}$ norms, respectively [45]. The real rational space, $\frac{1}{z}\mathcal{RH}_\infty$, consists of all strictly proper function $\Phi(z)$ and $\Phi(z) \in \mathcal{RH}_\infty$.

In this section, we begin by introducing the system level synthesis framework and then extend the method to deal with uncertainty. Finally, we present an SLS controller with a constant gain matrix for efficient computation.

A. SYSTEM LEVEL SYNTHESIS

From now on, we briefly describe the main ideas of the state-feedback SLS control. Please refer to [40], [41], [42], and [46] for a detailed explanation. Consider now the dynamics in (10b) and state-feedback control policy $u = Kx$ mapping from the state sequence to the control actions. Then, we obtain the following:

$$\begin{bmatrix} x \\ u \end{bmatrix} := \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} w = \begin{bmatrix} (zI - A - BK)^{-1} \\ K(zI - A - BK)^{-1} \end{bmatrix} w, \quad (12)$$

where $\Phi_x, \Phi_u \in \frac{1}{z}\mathcal{RH}_\infty$ are the strictly proper and closed loop transfer functions from the exogenous disturbance w to the state x and control input u , respectively. These maps are called *system response*. Note that the controller K denotes the transfer function while the control gain matrix is represented by K . The LQR problem (10) can be written as

$$\underset{\Phi_x, \Phi_u}{\text{minimize}} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_2} \quad (13a)$$

$$\text{subject to } [zI - A - B] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \quad (13b)$$

$$\Phi_x, \Phi_u \in \frac{1}{z}\mathcal{RH}_\infty, \quad (13c)$$

where I is an identity matrix.

Remark 1: The affine subspace described by (13b) and (13c) parameterizes all system responses from w to (x, u) as defined in (12) by an internally stabilizing state-feedback controller

$$u = Kx. \quad (14)$$

The affine subspace constraint also enforces the system responses to satisfy the system dynamics (10b), and thus, it is referred to as the *achievability constraint*. For any (Φ_x, Φ_u) satisfying the achievability condition, the controller (14) with

$$K = \Phi_u \Phi_x^{-1} \quad (15)$$

is internally stabilizing and ensuring the desired system response (12). Thus, our objective becomes to optimize

directly over these system responses to find the optimal controller (15).

For a simple example, suppose that the system responses from the finite number of disturbance sequence $[w(0)^\top, w(1)^\top, \dots, w(k-1)^\top]^\top$ to the state $x(k)$ with zero initial conditions. Consider the system dynamics of (10b) and a state-feedback control with a constant gain, $u(k) = Kx(k)$ where K is the matrix. The responses are of the following form [47]:

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = \sum_{j=0}^{k-1} \begin{bmatrix} (A + BK)^{k-(j+1)} \\ K(A + BK)^{k-(j+1)} \end{bmatrix} w(j) \quad (16a)$$

$$= \sum_{j=0}^{k-1} \begin{bmatrix} \phi_x(k-j) \\ \phi_u(k-j) \end{bmatrix} w(j), \quad \forall k \geq 1, \quad (16b)$$

where

$$\phi_x(m) = (A + BK)^{m-1}, \quad (17a)$$

$$\phi_u(m) = K(A + BK)^{m-1}. \quad (17b)$$

The system response elements $\phi_x(m)$ and $\phi_u(m)$ satisfy (13c) and the constraint in (13b) as follows:

$$\phi_x(m+1) = A\phi_x(m) + B\phi_u(m), \quad \phi_x(1) = I. \quad (18)$$

The constraint, $\phi_x(1) = I$, means that the disturbance $w(k)$ directly affects the state $x(k+1)$ as presented in (10b). The finite number of state response can be described as:

$$\underbrace{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(k) \end{bmatrix}}_x = \underbrace{\begin{bmatrix} \phi_x(1) & 0 & \cdots & 0 \\ \phi_x(2) & \phi_x(1) & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \phi_x(k) & \phi_x(k-1) & \cdots & \phi_x(1) \end{bmatrix}}_{\Phi_x} \underbrace{\begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(k-1) \end{bmatrix}}_w, \quad (19)$$

where Φ_x can be considered as the matrix representation of the convolution operator.

B. ROBUST SLS CONTROLLER

Now, we consider that the system consists of an approximated nominal model and a set of unknown but bounded modeling errors. The controller is computed to ensure stability and performance for any such admissible realization. Let the estimated matrices (\hat{A}, \hat{B}) for the true dynamic system matrices (A, B) and they satisfy

$$\|\delta_A\|_2 \leq \epsilon_A, \quad \|\delta_B\|_2 \leq \epsilon_B, \quad (20)$$

where $\delta_A := \hat{A} - A$ and $\delta_B := \hat{B} - B$. Let

$$\Delta := [\delta_A \quad \delta_B] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \quad (21)$$

be uncertain transfer functions for the given SLS controller. Here, (Φ_x, Φ_u) is the corresponding system response (12)

with the system matrix $(\widehat{A}, \widehat{B})$. The robust version of the SLS problem of (13) is given as follows [40]:

$$\underset{\Phi_x, \Phi_u}{\text{minimize}} \sup_{\substack{\|\delta_A\|_2 \leq \epsilon_A \\ \|\delta_B\|_2 \leq \epsilon_B}} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x(I + \Delta)^{-1} \\ \Phi_u(I + \Delta)^{-1} \end{bmatrix} \right\|_{\mathcal{H}_2} \quad (22a)$$

$$\text{subject to } [zI - \widehat{A} - \widehat{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \quad (22b)$$

$$\Phi_x, \Phi_u \in \frac{1}{z} \mathcal{RH}_\infty, \quad (22c)$$

$$\|\Delta\|_{\mathcal{H}_\infty} < 1. \quad (22d)$$

The stabilizing controller $K = \Phi_u \Phi_x^{-1}$ of (15) is designed by using the estimated matrices \widehat{A} and \widehat{B} in (22b) instead of the true dynamics A and B . The cost function (22a) is minimized under the worst case for the real parametric uncertainty δ_A and δ_B .

Remark 2: From the uncertain transfer function (21) and the constraint (22b), the following holds

$$[zI - A - B] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I + \Delta. \quad (23)$$

It means that the approximate system responses Φ_x and Φ_u , designed using the estimated system models $(\widehat{A}, \widehat{B})$, cannot exactly satisfy the achievability constraints (13b) for the real system (A, B) when uncertainty Δ exists.

Remark 3: The constraint (23) is equivalent to

$$[zI - A - B] \begin{bmatrix} \Phi_x(I + \Delta)^{-1} \\ \Phi_u(I + \Delta)^{-1} \end{bmatrix} = I. \quad (24)$$

Note that the controller in (15) is rewritten as

$$K = \Phi_u \Phi_x^{-1} = \underbrace{\Phi_u(I + \Delta)^{-1}}_{\text{actual response}} \underbrace{(\Phi_x(I + \Delta)^{-1})^{-1}}_{\text{actual response}}. \quad (25)$$

Thus, the controller K achieves the actual response

$$\begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} \Phi_x(I + \Delta)^{-1} \\ \Phi_u(I + \Delta)^{-1} \end{bmatrix} w. \quad (26)$$

Therefore, the actual uncertain system response in (26) naturally leads to the cost function (22a). The response (26) becomes equivalent to (12) when $\Delta = 0$.

Remark 4: The stability of the perturbed system (26) is guaranteed if and only if $(I + \Delta)^{-1}$ is stable. Furthermore, from the small gain theorem [45], [48], sufficient condition for robust stability is achieved if $\|\Delta\|_{\mathcal{H}_\infty} < 1$. This condition is reflected in (22d). Thus, any controller K ensuring the stability for $(\widehat{A}, \widehat{B})$ with (22d) also stabilizes the real system (A, B) .

To address the computational difficulty in the cost function (22a) arising from Δ , a slightly conservative approach is employed. For a given $\gamma < 1$, the γ -optimal robust control problem is obtained from (22)

$$\underset{\Phi_x, \Phi_u}{\text{minimize}} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_2} \quad (27a)$$

$$\text{subject to } [zI - \widehat{A} - \widehat{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \quad (27b)$$

$$\Phi_x, \Phi_u \in \frac{1}{z} \mathcal{RH}_\infty, \quad (27c)$$

$$\sqrt{2} \left\| \begin{bmatrix} \epsilon_A \Phi_x \\ \epsilon_B \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \gamma, \quad (27d)$$

where the controller is $u = Kx = \Phi_u \Phi_x^{-1}x$. The detailed derivation is described in APPENDIX A.

We note that (Φ_x, Φ_u) are infinite dimensional as can be seen in (11). A finite-dimensional optimization problem and its solution can be acquired with comparable assurances regarding robustness and performance [40], [49]. The finite impulse response (FIR) function with horizon, F , is used

$$\Phi^F(z) = \sum_{k=1}^F \phi(k)z^{-k}. \quad (28)$$

The FIR function $\Phi^F(z)$ is similar to (19).

C. ROBUST SLS GAIN CONTROLLER

Though the SLS controller has the finite impulse response, it is more convenient to use an SLS controller with a constant gain matrix like LQR. This variant is now referred to as the SLS gain controller. In this section, our goal is to find the optimal SLS gain controller $u(k) = Kx(k)$ with the optimal constant gain matrix K instead of the transfer function K [42]. The optimization problem (27) has the following additional constraint:

$$K = \Phi_u \Phi_x^{-1}. \quad (29)$$

Note that the constant gain matrix K is employed instead of the transfer function K . From the achievability constraints of (27c) and the constraint in (29), the system responses of (12) are converted into

$$\Phi_x = (zI - \widehat{A} - \widehat{B}K)^{-1}, \quad (30a)$$

$$\Phi_u = K(zI - \widehat{A} - \widehat{B}K)^{-1}. \quad (30b)$$

Then, the optimization problem (27) becomes

$$\underset{K}{\text{minimize}} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}}K \end{bmatrix} (zI - \widehat{A} - \widehat{B}K)^{-1} \right\|_{\mathcal{H}_2} \quad (31a)$$

$$\text{s.t. } \sqrt{2} \left\| \begin{bmatrix} \epsilon_A I \\ \epsilon_B K \end{bmatrix} (zI - \widehat{A} - \widehat{B}K)^{-1} \right\|_{\mathcal{H}_\infty} \leq \gamma. \quad (31b)$$

Now, the problem is no longer convex. The conservative controller can be obtained via upper bound of this problem and semidefinite program [42], [45], [50], as follows:

$$\underset{P, L, U, V, W}{\text{minimize}} \text{Tr}(QU) + \text{Tr}(RW) \quad (32a)$$

$$\text{subject to } \begin{bmatrix} U & V & P \\ V^T & W & L \\ P & L^T & R \end{bmatrix} \geq 0, \quad (32b)$$

$$\begin{bmatrix} P - I & \widehat{A}P + \widehat{B}L & 0 & 0 \\ P\widehat{A}^\top + L^\top\widehat{B}^\top & P & \epsilon_{AP} & \epsilon_{BL}^\top \\ 0 & \epsilon_{AP} & \frac{1}{2}\gamma^2 I & 0 \\ 0 & \epsilon_{BL} & 0 & \frac{1}{2}\gamma^2 I \end{bmatrix} \geq 0, \quad (32c)$$

where P, U , and W are symmetric matrices. The standard variable substitution $L := KP$ is used. The detailed derivation is presented in APPENDIX B.

IV. MODEL PREDICTIVE CONTROL

The MPC problem for the trajectory optimization is

$$\text{minimize } \mathcal{J} = \sum_{k=0}^T \mathcal{L}(s(k), a(k)) \quad (33a)$$

$$\text{subject to } s(k+1) = \widehat{f}(s(k), a(k)), \quad (33b)$$

$$s_{\min} \leq s(k) \leq s_{\max}, \quad (33c)$$

$$a_{\min} \leq a(k) \leq a_{\max}, \quad (33d)$$

where \widehat{f} is the estimated nonlinear model for real dynamics f_c in (6). A trajectory $\{\mathcal{S}, \mathcal{A}\}$ is a sequence of states $\mathcal{S} = \{s(0), \dots, s(T)\}$ and controls $\mathcal{A} = \{a(0), \dots, a(T)\}$ satisfying the nonlinear equations of motion (33b). At each sampling time step, starting at the current state, an open loop optimal control problem is solved over a finite horizon T . Then, the command state $s_d \in \mathcal{S}^*$ and the optimal control input $u_d := a^*(0)$ are given to the real dynamic system only during the following sampling interval t_{out} as shown in Fig. 2. At the next time step, a new optimal control problem is solved over a shifted horizon.

The total cost \mathcal{J} is the sum of running costs \mathcal{L} until the horizon T is reached. The running cost is

$$\begin{aligned} \mathcal{L}(s(k), a(k)) = & c_1 \text{Traversability}(s) + c_2 \|a\|_2^2 \\ & + c_3 \|a - a_{\text{pre}}\|_2^2 + c_4 \|\theta - \theta_d\|_2^2, \end{aligned} \quad (34)$$

where the ‘‘Traversability’’ cost represents how well the vehicle is expected to traverse in the location of the traversability map. The state $(p_X(k), p_Y(k))$ looks up the values of a two-dimensional traversability map, which represents the area of the non-traversable region as 1 and the traversable area as 0. The example map is presented in Fig. 2. The ‘‘Traversability’’ impose very large penalty to the location $(p_X(k), p_Y(k))$ when it is in the non-traversable region. The second term, $c_2 \|a\|_2^2$, minimizes the magnitude of the control input. The control input of the previous sampling time step is denoted by a_{pre} , and thus minimizing $c_3 \|a - a_{\text{pre}}\|_2^2$ means to generate a smooth control input between current time t and previous time $t - t_{\text{out}}$. The cost of $c_4 \|\theta - \theta_d\|_2^2$ represents the vehicle heading error.

Due to the non-smoothness of traversability information and nonlinear nature and vehicle dynamics, we leverage the particle swarm optimization. This population-based search algorithm is based on the social-psychological tendency of individuals to emulate success of neighbors and their own successes [51], [52]. To make the problem more numerically tractable and to reduce computational burden, we parameterize the control trajectory \mathcal{A} . The state trajectory

Algorithm 1 Model Predictive Control Based on PSO

```

1: Given: Dynamic model,  $s(k+1) = \widehat{f}(s(k), a(k))$ ;
   Number of particles,  $i = 1, \dots, N$ ;
   Time steps,  $k = 0, \dots, T$ ;
   Number of iterations,  $j = 1, \dots, J$ ;
2: while vehicle does not reached the final way point do
3:    $j \leftarrow 1$  : Initialize the PSO iteration count
4:   for each particle  $i = 1, \dots, N$  do
5:     Sample  $p_1^i \sim \text{Uniform}[p_{\min}, p_{\max}]$ 
6:   end for
7:   repeat
8:     for each particle  $i = 1, \dots, N$  do
9:        $s_j^i(0) \leftarrow$  Get vehicle's current state
10:      for each time step  $k = 0, \dots, T$  do
11:        Control Input:  $a_j^i(k) \leftarrow$  Eq. (35)
12:        Running Cost:  $\mathcal{L}(s_j^i(k), a_j^i(k))$  with Eq. (34)
13:        Total Cost:
14:           $J(a_j^i(k)) \leftarrow J(a_j^i(k-1)) + \mathcal{L}(s_j^i(k), a_j^i(k))$ 
15:          Next Time State:  $s_j^i(k+1) = \widehat{f}(s_j^i(k), a_j^i(k))$ 
16:        end for
17:        // Set the control sequence
18:         $\mathcal{A}(p_j^i) \leftarrow \{a_j^i(0), a_j^i(1), \dots, a_j^i(T)\}$ 
19:        // Set the personal best position
20:        if  $J(\mathcal{A}(p_j^i)) < J(\mathcal{A}(q^i))$  then
21:           $q^i \leftarrow p_j^i$ 
22:        end if
23:        // Set the global best position
24:        if  $J(\mathcal{A}(p_j^i)) < J(\mathcal{A}(q^*))$  then
25:           $q^* \leftarrow p_j^i$ 
26:        end if
27:      end for
28:       $J_j^* \leftarrow J(\mathcal{A}(q^*))$ .
29:      for each particle  $i = 1, \dots, N$  do
30:        Update the velocity  $v^i$  using Eq. (38)
31:        Update the position  $p^i$  using Eq. (37)
32:      end for
33:       $j \leftarrow j + 1$ 
34:    until convergence criterion satisfied:
35:       $\|J_j^* - J_{j-1}^*\| \leq \epsilon$  or  $j = J$ 
36:    for each time step  $k = 0, \dots, T$  do
37:      Optimal Control:  $a^*(k) \leftarrow a(k, q^*)$  in Eq. (35)
38:      Optimal Control Sequence:  $\mathcal{A}^* \leftarrow \text{Append}(a^*(k))$ 
39:      Optimal Control Sequence:  $\mathcal{S}^* \leftarrow \text{Append}(s^*(k))$ 
40:      Optimal State:  $s^*(k+1) = \widehat{f}(s^*(k), a^*(k))$ 
41:    end for
42:    Set Reference Input:  $u_d := a^*(0)$  of  $\mathcal{A}^*$ 
43:    Set Command State:  $s_d \in \mathcal{S}^*$ 
44:  end while

```

\mathcal{S} is obtained by integration of (33b) from the initial state $s(0)$. Thus, the dynamic constraints are unnecessary and state-control trajectories are always strictly feasible even if the optimization is terminated prematurely due to the limited computation time. The state constraint (33c) serves to crop

the trajectory if it exceeds the size of the traversability map. If the controls are unconstrained, the optimization problems become fully unconstrained optimization [53], [54]. In addition, the stochastic randomness of the sampling procedure may give rise to considerable chattering in the resulting control. The rapid changes in control action commands can burden the actuators and potentially leads to system instability [33]. However, the chattering can be completely removed by the parameterization of the control sequence with smooth functions as follows.

The control trajectory \mathcal{A} is parameterized by the sum of smooth trigonometric curves and varying parameters in the following manner:

$$a(k, p_j^i) = \sum_{m=1}^M A_{jm}^i \cos(\omega_{jm}^i k) + A_{jm}^i \sin(\omega_{jm}^i k), \quad (35)$$

where $k = 0, \dots, T$ is time sequences and $i = 1, \dots, N$ is number of particles in swarm. $m = 1, \dots, M$ denotes the parameters for amplitudes and frequencies. Let p_j^i denote the position of particle i in the search space at the iteration step $1 \leq j \leq J$

$$p_j^i = [A_{j1}^i, A_{j2}^i, \dots, A_{jM}^i, \omega_{j1}^i, \omega_{j2}^i, \dots, \omega_{jM}^i]^\top, \quad (36)$$

where A_{jm}^i and ω_{jm}^i are amplitude and frequency parameters to be found to minimize the cost.

The position of the particle i is changed by adding a velocity, r_j^i , to the current position, i.e.

$$p_{j+1}^i = p_j^i + r_{j+1}^i, \quad j = 1, \dots, L, \quad (37)$$

where the initial position p_1^i is uniform random sample generated from $[p_{\min}, p_{\max}]$. The velocity of particle i is calculated as

$$r_{j+1}^i = q^i + \underbrace{c_1 \text{rand}_1(q^i - p_j^i)}_{\text{cognitive}} + \underbrace{c_2 \text{rand}_2(q^* - p_j^i)}_{\text{social}}, \quad (38)$$

where c_1 and c_2 are positive acceleration constants used to scale the contribution of the ‘‘cognitive’’ and ‘‘social’’ components, respectively. The scalars ‘‘rand₁’’ and ‘‘rand₂’’ are random values in the range $[0, 1]$, sampled from a uniform distribution introducing a stochastic element. The *personal best position* q^i is the best position that the i -th particle has visited since the first iteration. The *global best position*, q^* is discovered by any of the particles so far.

The iteration is repeated until the convergence criteria is satisfied $\|\mathcal{J}_j^* - \mathcal{J}_{j-1}^*\| \leq \epsilon$ or the maximum number of iteration is reached $j = J$. The proposed method is summarized in Algorithm 1, where the control input $a(k, p_j^i)$ is denoted by $a_j^i(k)$. The state-control trajectories are always strictly feasible even if the optimization is terminated prematurely due to the limited computation time. The dynamic or kinematic constraints are automatically satisfied because the state trajectory \mathcal{S} is obtained by integrating the equations of motion from the initial state $s(0)$ and the parameterized control trajectory \mathcal{A} .

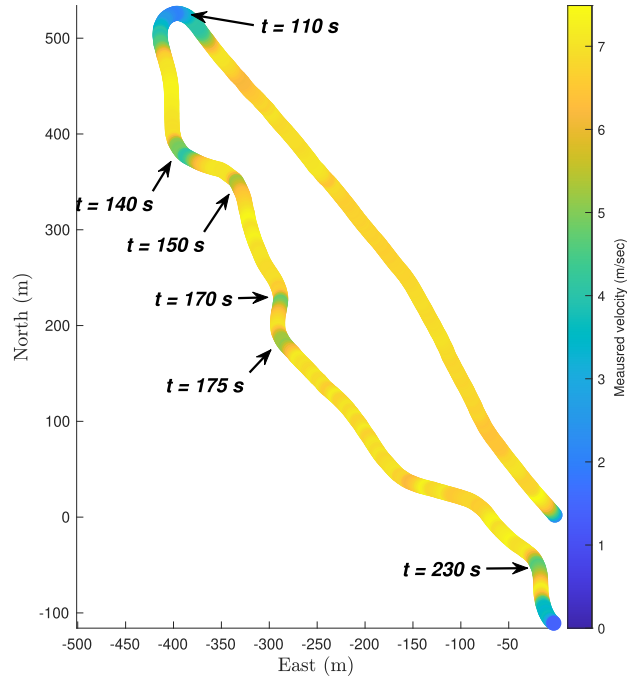


FIGURE 4. Vehicle trace and measured velocity of the experiment test with the robust SLS gain controller and MPC.

V. NUMERICAL AND EXPERIMENTAL RESULTS

We begin by outlining the test setup for both numerical simulations and experimental tests. Next, we highlight the robust performance of the path tracker and TDF controller, respectively. Finally, we end this section by validating the kinematic and dynamic models.

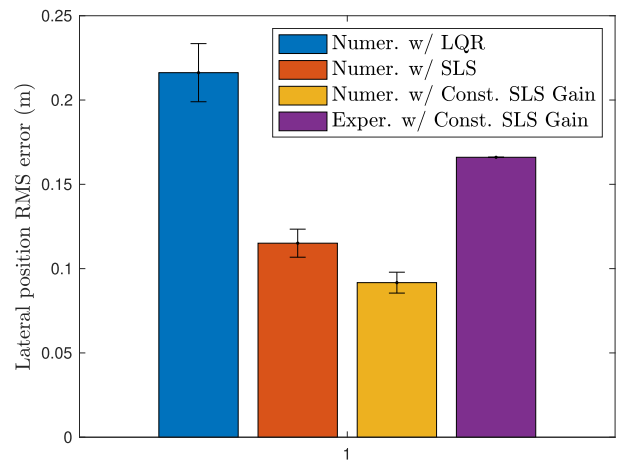


FIGURE 5. Lateral position control errors e_y for numerical simulation and experimental results. Experiment is performed with the constant SLS gain and MPC.

A. TEST SETUP

The unmanned surveillance vehicle and the corresponding control structure are depicted in Fig. 1 and Fig. 2, respectively. The MPC loop of (33) determines the feedforward steering input and the state trajectory that the path tracking controller should follow. The nonlinear and linear equations

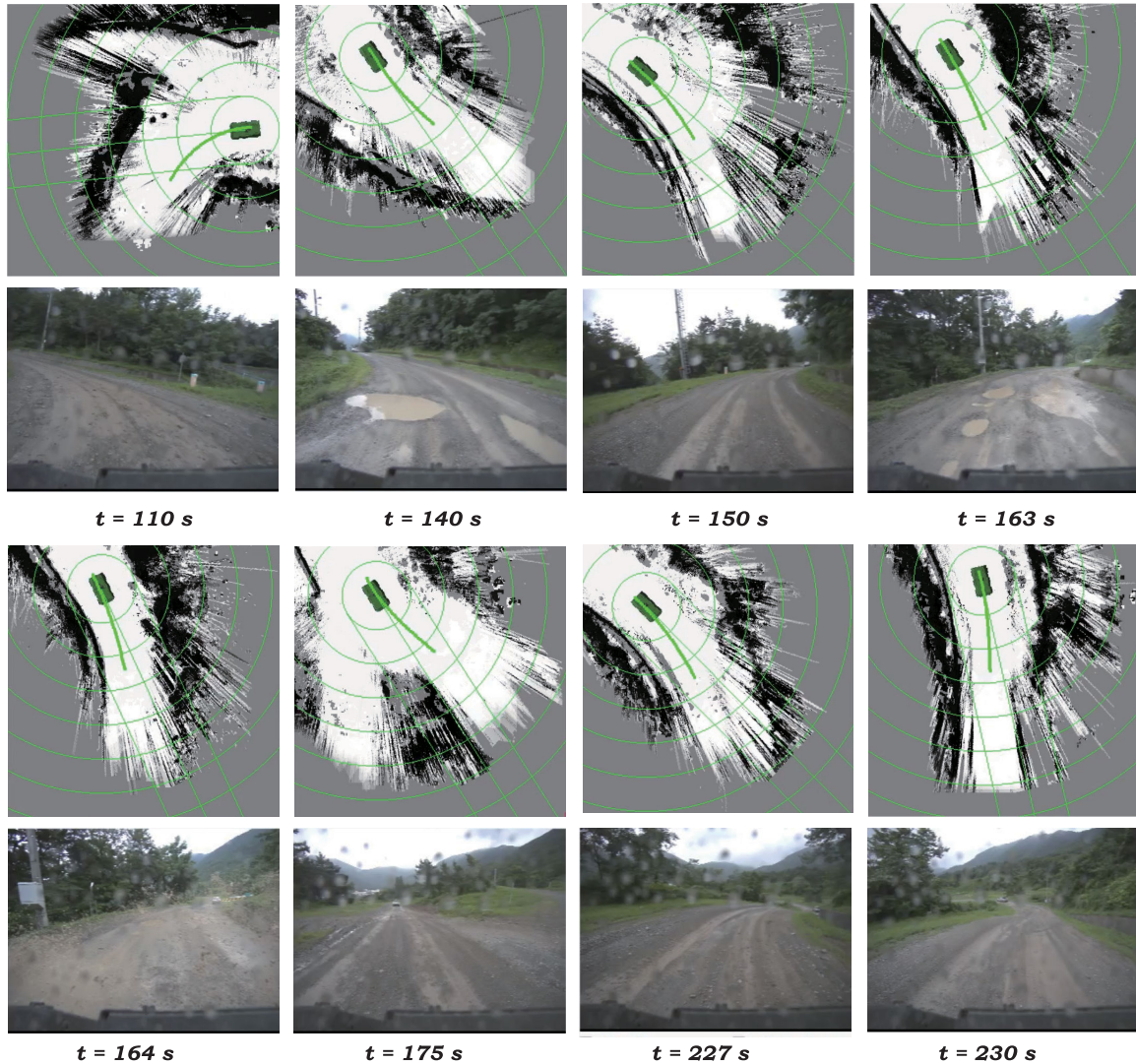


FIGURE 6. Test results of the TDF control system with the robust SLS gain controller. Robust stability is guaranteed even on the rain-soaked and unpaved road. (Top) Optimal trajectories (green lines) on the traversability map. Online trajectory generator, MPC uses a traversability map constructed by using three LiDAR sensors and continuously replans the trajectory during execution. (Bottom) Environment image captured by the on-board visible camera.

of motions in (6) and (9) are used for MPC and the design of the path tracking controller, respectively. The SLS controllers are synthesized in MATLAB and Python, where CVX and CVXPY packages are used [55], [56]. The time horizon for MPC is set to $T = 40$ in (33a). The number of particles is $N = 10$ and the number of amplitude and frequency parameters is $M = 5$ as presented in (35). The maximum number of iteration of (36) is $J = 100$.

We consider the effect of discretization time on the prediction error because the inner and outer control loops operate at two different control frequencies. The inner and outer loop control periods are $t_{in} = 0.02$ s and $t_{out} = 0.1$ s, respectively. Larger discretization time allows the model to predict over longer horizon, whereas shorter step size tends to yield more accurate predictions when we use the same number of predicted points, $T = 40$. As the computation for

PSO is demanding, the nonlinear kinematic model with fewer states in (7) is used for the estimated dynamics \hat{f} instead of the linear dynamics in (4) to enhance computational efficiency. In addition, it is known that the accuracy of the kinematic model is less sensitive to the larger step size [57]. The path tracking controller is designed based on the more accurate dynamic model using the discretization time of $t_{in} = 0.02$ s. All program is written in C/C++ and parallel computation is used in PSO. We do not use any graphical processing unit (GPU).

We are given estimated system matrices $\hat{A} \in \mathbb{R}^{4 \times 4}$, $\hat{B} \in \mathbb{R}^{4 \times 1}$ for the true system (9). The time horizon for the FIR function in (28) is $F = 40$. Then, the state-feedback SLS controller is realized by a standard state space form [49], $A_K \in \mathbb{R}^{156 \times 156}$, $B_K \in \mathbb{R}^{156 \times 4}$, $C_K \in \mathbb{R}^{1 \times 156}$, and $D_K \in \mathbb{R}^{1 \times 4}$. The matrix size of the SLS controller is too large

TABLE 1. The root mean square (RMS) errors for lateral position control errors e_y and stability. Units are meters.

Controller	Numerical			Experimental	
	LQR	SLS	SLS Gain	LQR	SLS Gain
Lat. Pos. Err.	0.22	0.12	0.09	-	0.17
Stability	○	○	○	×	○

for efficient implementation on a small, ruggedized on-board computer. Therefore, the experiment is conducted with the LQR and SLS gain controllers, where each path tracker has a constant gain matrix $K \in \mathbb{R}^{1 \times 4}$.

Field tests are conducted at Changwon Proving Ground in Changwon City, South Korea. The track features a varied road course with several turns, straight-line paths, and uphill and downhill sections, as shown in Fig. 4. We verified the effectiveness of the TDF controller over four distinct seasons throughout the year. Representative results are presented in the following sections.

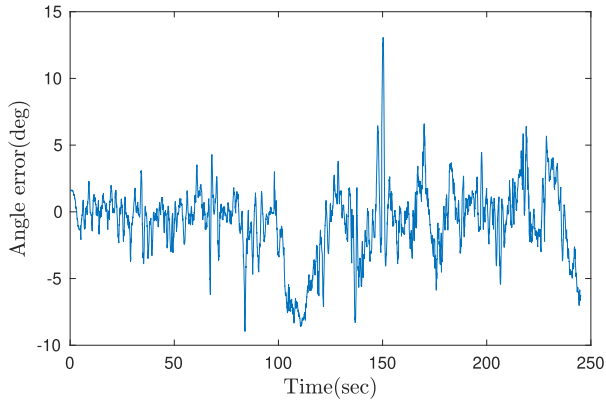


FIGURE 7. Control angle error e_θ for TDF control system with the SLS gain controller. Maximum and RMS angle errors are 13.1 (deg) and 2.8 (deg) respectively.

B. ROBUST PERFORMANCE FOR TRACKING CONTROL

The path tracker incorporates three types of controllers, one of which is the transfer matrix K , while the other two have constant gain matrices K . The transfer matrix is obtained from the SLS framework in (27) and it is realized in the form of the FIR function. The latter two controllers come from the LQR of (10) and the robust SLS gain controller in (32). The same state weighting Q and control input weighting matrix R are used. We confirm that the same system responses Φ_x and Φ_u are obtained from LQR and SLS when $\Delta = 0$. The performance of three path tracking controllers is verified by simulating the closed-loop system using the nonlinear vehicle model in (1). Here, to evaluate only the tracker’s performance, MPC is not used.

We measure the control performance over 15,000 time steps to simulate the test scenario imitating the environment shown in Fig. 4. For numerical simulations, the parameter-dependent and nonlinear dynamic model, $f_c(\pi)$ in (6) is used with the parameter vector $\pi = (\pi_1, \dots, \pi_m)$

in (1). Given interval bounds $\pi_i \in [\underline{\pi}_i, \overline{\pi}_i]$ on each π_i , where $\pm 10\%$ variations from the nominal value are applied. We repeat this simulation for 500 independent runs, each with a different Gaussian random disturbance w and perturbed parameter dependent dynamics $f_c(\pi)$. The parameter vector π_i is uniformly sampled from $[\underline{\pi}_i, \overline{\pi}_i]$. The averages of lateral position control errors from the simulation tests are illustrated on the left side of Table 1 and Fig. 5. Although stability is ensured by all controllers in numerical simulations, the performance of the LQR controller is found to be inferior to the other two controllers.

C. ROBUST PERFORMANCE FOR TDF CONTROL

The vehicle trace and corresponding measured velocity are presented in Fig. 4. During the course of the experimentation, the MPC loop with the robust SLS gain controller is utilized. Fig. 6 illustrates the optimal trajectories on the traversability map along with the environment image captured by the on-board visible camera. Three LiDAR sensors are used to acquire a three-dimensional point cloud for the analysis of the environment. Within the traversability map, areas marked in white and black denote traversable and non-traversable regions, respectively. The gray space represents the probability of traversability between 0 and 1. The structure of the map is similar to the well-known occupancy grid map, but each cell reflects the traversability rather than occupancy of the given space. Each cell is

$$p(i, j) = \begin{cases} < 0.5 & \text{Non-traversable} \\ 0.5 & \text{Unknown} \\ > 0.5 & \text{Traversable} \end{cases} \quad (39)$$

where i, j are the cell indices in the map.

As shown in Fig. 6, by using the exteroceptive perception, the vehicle could anticipate the terrain and adapt its motion to achieve fast and smooth driving via MPC. With these results, we see that the generated trajectory satisfies vehicle dynamics (7) while simultaneously avoiding obstacles and minimizing the cost in (34). The trajectories are generated as far from the obstacles as possible, which are represented by black dots. The controller is solving the optimization problem (33) within $t_{out} = 0.1$ s, and updates are made to the cost function, to reflect new regions of the state space. In the inner loop, the robust SLS gain controller stabilizes the system for every sampling interval $t_{in} = 0.02$ s.

We conclude that the TDF controller with the robust SLS gain controller can guarantee the robust performance even in wet, dirty, and off-road environments as well as in dry conditions. Due to uncertainty and unknown disturbance, the experimental result of the SLS gain controller exhibits a slightly larger control error when compared to the numerical simulation, as evidenced by Table 1 and Fig. 5. Fig. 9 shows that the control angle error e_θ , where the maximum and RMS values are 13.1 deg and 2.8 deg respectively. It verifies that the TDF control works well so that the small angle approximation in (8) is a valid assumption. The driving area after 120 s is

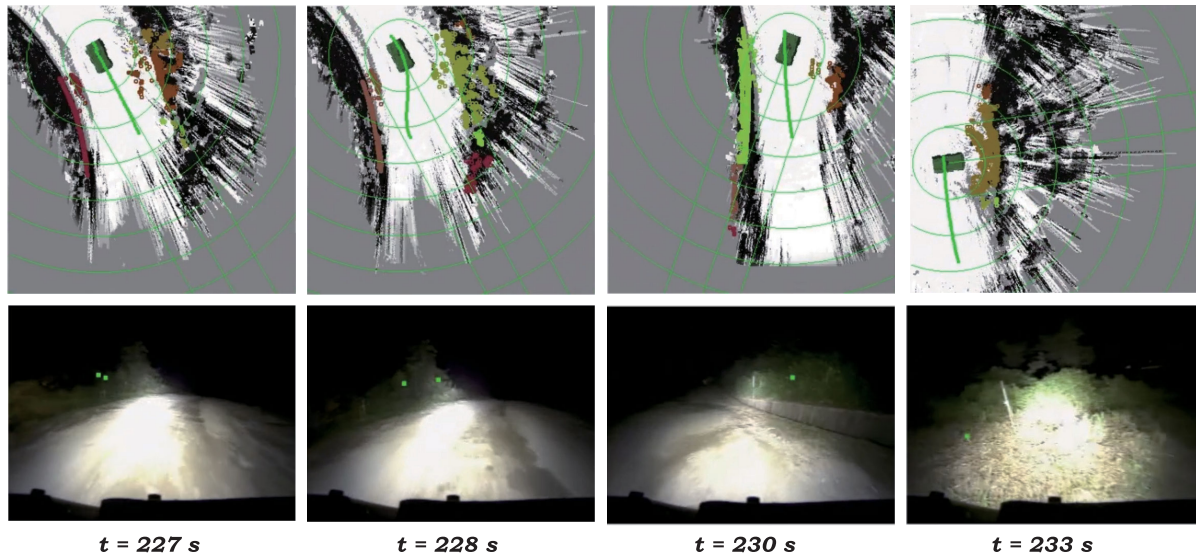


FIGURE 8. Test results of the TDF control system with the LQR controller. The controller cannot always ensure stability and thus robust stability is not guaranteed. (Top) Optimal trajectories (green lines) on the traversability map. (Bottom) Environment image captured by the on-board visible camera.

downhill with many curves, as presented in Fig. 4. The control angle error is increased from 120 s because the vehicle slips more often in this area.

Yet there are a few instances when the LQR controller fails to maintain stability. In other words, the TDF controller with LQR cannot always ensure stability. Fig. 8 shows one case that robust stability is not guaranteed on the dry road. When the vehicle slips at $t = 227$ s, the path tracker does not compensate this small deviation. As the tracking error grows, the reference trajectory of MPC is abruptly changed to follow given way points. The tracking controller attempt to adheres to the trajectory by drastically changing the steering, but subsequently stability is lost. The experimental results of Fig. 6 and 8 are also demonstrated in the supplementary video.

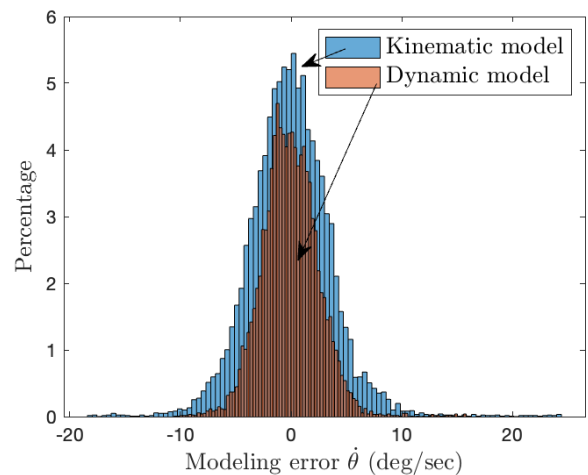


FIGURE 10. Distribution of yaw rate prediction errors between kinematic and dynamic models with discretization interval $t_{in} = 0.02$ s.

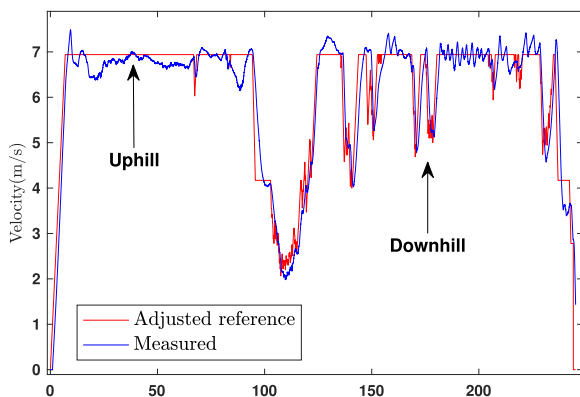


FIGURE 9. Experimental results for the adjusted reference velocity and measured velocity when the command velocity is set at 6.94 m/s.

Fig. 9 shows the adjusted reference velocity and measured velocity for TDF control with the SLS gain controller. Here, the command velocity is set at 6.94 m/s; however, the

reference velocity for the speed control system is adaptively reduced to guarantee its stability. This velocity is adjusted by the curvature of the optimal trajectory S^* generated by MPC and is used as the command to the speed controller. The driving course from the starting point to 100 s is uphill and is a straight line, so there is little change in the reference velocity. In the road section between 90 s and 120 s, where the curvature is very large as can be seen in Fig. 6, the reference velocity is greatly reduced. The driving area after 120 s is downhill with many curves, as presented in Fig. 4. To ensure stability, there are many changes to the reference velocity from 120 s and the response of the speed controller exhibits many overshoots as shown in Fig. 9. Because the brake and motor torque are frequently used to tack the speed command when going downhill.

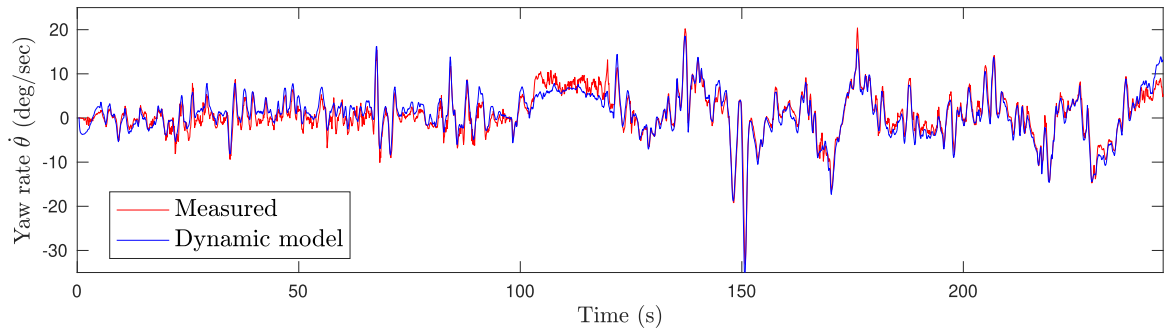


FIGURE 11. Measured yaw rate and estimated yaw rate of the dynamic model.

TABLE 2. The root mean square (RMS) prediction errors, with the integration time $t_{in} = 0.02$ s.

Model	Yaw rate RMS error (deg/s)
Kinematic	3.17
Dynamic	1.96

D. MODEL VALIDATION FOR CONTROLLED SYSTEM

We investigate the validity of the linear dynamics in (4) and nonlinear kinematic model in (7) as the performance of the TDF controller heavily depends on these models. In numerical simulations, it has been observed that the kinematic model tends to cause oversteering, resulting in a smaller turning radius, compared to the dynamic one. This difference arises because the kinematic model cannot properly account for slip.

We evaluate how well the two models are able to predict vehicle’s yaw rate in comparison to the state measured in the experimental test. This error is referred to as the *prediction error*. The states are initialized with zeros and measured control input or steering sequences of the experimental tests of Fig. 4 are inputted into the models. The distributions and RMS errors of the prediction errors are presented in Fig. 10 and Table 2, respectively. The linear dynamics proves to be more accurate than the kinematic model under off-road conditions.

Fig. 11 shows the yaw rates of the dynamic model and the real autonomous vehicle. The estimated yaw rate is observed to be lower than that of the experimental result. Specifically, while the dynamic model is more accurate than the kinematic one, it still does not capture the tire’s slip in curved off-road conditions. However, the robust TDF controller consistently ensures stability and performance in the face of uncertainty.

VI. CONCLUSION

This paper introduces a real-time optimal trajectory generator and a robust controller designed for an unmanned off-road vehicle. We take advantage of the two-degree-of-freedom controller in order to achieve predictive driving control when perceptual information is given while the fast state-feedback controller concurrently attains robustness. To generate trajectory, the model predictive control method is used where

the optimization is solved via PSO. It addresses problems of the non-smoothness of the traversability information and nonlinear nature of vehicle dynamics. For the feedback controller, we leverage the system level synthesis framework for robustness and convexity. Especially, we concentrate on realizing the proposed method as a practical means of the real-time control system. The robust performance of TDF control system is highlighted through extensive numerical and experimental tests for the unmanned vehicle to successfully execute the navigation missions in highly uncertain environments.

APPENDIX A
MATHEMATICAL DERIVATIONS FOR ROBUST SLS CONTROLLER

The cost function in (22a) satisfies the following inequality if $\|\Delta\|_{\mathcal{H}_\infty} \leq \gamma < 1$

$$C_\Delta \leq C \|(I + \Delta)^{-1}\|_{\mathcal{H}_\infty} \leq \frac{C}{1 - \|\Delta\|_{\mathcal{H}_\infty}} \leq \frac{C}{1 - \gamma} \quad (40)$$

where

$$C_\Delta = \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x(I + \Delta)^{-1} \\ \Phi_u(I + \Delta)^{-1} \end{bmatrix} \right\|_{\mathcal{H}_2}, \quad (41a)$$

$$C = \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_2}. \quad (41b)$$

In (40), the following inequalities are used

$$\begin{aligned} &\|(I + \Delta)^{-1}\|_{\mathcal{H}_\infty} \\ &\leq \|(I - \Delta)^{-1}\|_{\mathcal{H}_\infty} \\ &\leq I + \|\Delta\|_{\mathcal{H}_\infty} + \|\Delta\|_{\mathcal{H}_\infty}^2 + \dots \leq \frac{1}{1 - \|\Delta\|_{\mathcal{H}_\infty}}. \end{aligned} \quad (42)$$

To address the intractability of $\|\Delta\|_{\mathcal{H}_\infty}$ in (22d), the following upper bound is used

$$\begin{aligned} \|\Delta\|_{\mathcal{H}_\infty} &= \left\| \begin{bmatrix} \delta_A & \delta_B \\ \epsilon_A & \epsilon_B \end{bmatrix} \begin{bmatrix} \epsilon_A \Phi_x \\ \epsilon_B \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \\ &\leq \left\| \begin{bmatrix} \delta_A & \delta_B \\ \epsilon_A & \epsilon_B \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} \epsilon_A \Phi_x \\ \epsilon_B \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \sqrt{2} \left\| \begin{bmatrix} \epsilon_A \Phi_x \\ \epsilon_B \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_\infty}. \end{aligned} \quad (43)$$

With (40) and (43), the robust SLS problem in (22) becomes

$$\underset{\gamma \in [0,1]}{\text{minimize}} \frac{1}{1-\gamma} \underset{\Phi_x, \Phi_u}{\text{minimize}} \mathcal{C} \quad (44a)$$

$$\text{subject to (22b), (22c),} \quad (44b)$$

$$\sqrt{2} \left\| \begin{bmatrix} \epsilon_A \Phi_x \\ \epsilon_B \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \gamma. \quad (44c)$$

In this paper, we assume that γ is given and focus on the minimization of \mathcal{H}_2 performance index rather than the robustness maximization as shown in (27).

APPENDIX B MATHEMATICAL DERIVATIONS FOR SLS GAIN CONTROLLER

A. \mathcal{H}_∞ CONSTRAINT

Let $A_{cl} := \widehat{A} + \widehat{B}K$ and identity and zero matrices with appropriate dimensions be denoted by $I_n \in \mathbb{R}^{n \times n}$ and $0_{n \times p} \in \mathbb{R}^{n \times p}$, respectively. The Kalman-Yakubovich-Popov lemma or bounded real lemma enables us to reformulate the \mathcal{H}_∞ constraint (31b) represented in the time domain as linear matrix inequalities with a positive definite matrix $X = X^\top > 0$ [58]:

$$\begin{bmatrix} A_{cl} & I_n \\ \begin{bmatrix} \sqrt{2}\epsilon_A I_n \\ \sqrt{2}\epsilon_B K \end{bmatrix} & 0_{(n+p) \times n} \end{bmatrix}^\top \begin{bmatrix} X & 0_{n \times (n+p)} \\ 0_{(n+p) \times n} & I_{n+p} \end{bmatrix} \\ \times \begin{bmatrix} A_{cl} & I_n \\ \begin{bmatrix} \sqrt{2}\epsilon_A I_n \\ \sqrt{2}\epsilon_B K \end{bmatrix} & 0_{(n+p) \times n} \end{bmatrix} - \begin{bmatrix} X & 0_{n \times n} \\ 0_{n \times n} & \gamma^2 I_n \end{bmatrix} \quad (45)$$

$$= \begin{bmatrix} A_{cl}^\top X A_{cl} + \begin{bmatrix} \sqrt{2}\epsilon_A I_n \\ \sqrt{2}\epsilon_B K \end{bmatrix}^\top \begin{bmatrix} \sqrt{2}\epsilon_A I_n \\ \sqrt{2}\epsilon_B K \end{bmatrix} A_{cl}^\top X \\ X A_{cl} & X \end{bmatrix} \\ - \begin{bmatrix} X & 0_{n \times n} \\ 0_{n \times n} & \gamma^2 I_n \end{bmatrix} \leq 0. \quad (46)$$

Dividing both sides of inequality (46) with γ^2 and renaming X/γ^2 as X , (46) can be written as

$$\begin{bmatrix} A_{cl}^\top X A_{cl} - X & A_{cl}^\top X \\ X A_{cl} & X - I_n \end{bmatrix} - \begin{bmatrix} \sqrt{2}\epsilon_A I_n & \sqrt{2}\epsilon_B K^\top \\ 0_{n \times n} & 0_{n \times p} \end{bmatrix} \\ \times \begin{bmatrix} -\gamma^{-2} I_n & 0_{n \times p} \\ 0_{p \times n} & -\gamma^{-2} I_p \end{bmatrix} \begin{bmatrix} \sqrt{2}\epsilon_A I_n & 0_{n \times n} \\ \sqrt{2}\epsilon_B K & 0_{p \times n} \end{bmatrix} \leq 0. \quad (47)$$

From Schur complement lemma [50], (47) can be rewritten as

$$\begin{bmatrix} A_{cl}^\top X A_{cl} - X & A_{cl}^\top X & \sqrt{2}\epsilon_A I_n & \sqrt{2}\epsilon_B K^\top \\ X A_{cl} & X - I_n & 0_{n \times n} & 0_{n \times p} \\ \sqrt{2}\epsilon_A I_n & 0_{n \times n} & -\gamma^2 I_n & 0_{n \times p} \\ \sqrt{2}\epsilon_B K & 0_{p \times n} & 0_{p \times n} & -\gamma^2 I_p \end{bmatrix} \leq 0. \quad (48)$$

The following equation is equivalent to (48)

$$\begin{bmatrix} -X^{-1} & A_{cl} & I_n & 0_{n \times n} & 0_{n \times p} \\ A_{cl}^\top & -X & 0_{n \times n} & \sqrt{2}\epsilon_A I_n & \sqrt{2}\epsilon_B K^\top \\ I_n & 0_{n \times n} & -I_n & 0_{n \times n} & 0_{n \times p} \\ 0_{n \times n} & \sqrt{2}\epsilon_A I_n & 0_{n \times n} & -\gamma^2 I_n & 0_{n \times p} \\ 0_{p \times n} & \sqrt{2}\epsilon_B K & 0_{p \times n} & 0_{p \times n} & -\gamma^2 I_p \end{bmatrix} \leq 0. \quad (49)$$

We can obtain (48) by applying Schur's lemma to (49). Let us multiply both sides of (49) by the matrix D such as $F = DED^\top$. Where E is the matrix in (49) and $D := \text{diag}[I_n, X^{-1}, I_n, \frac{1}{\sqrt{2}}I_n, \frac{1}{\sqrt{2}}I_p]$. Then, applying $Y := X^{-1}$ to F leads to the following inequality:

$$\begin{bmatrix} -Y & A_{cl}Y & I_n & 0_{n \times n} & 0_{n \times p} \\ YA_{cl}^\top & -Y & 0_{n \times n} & \epsilon_A Y & \epsilon_B YK^\top \\ I_n & 0_{n \times n} & -I_n & 0_{n \times n} & 0_{n \times p} \\ 0_{n \times n} & \epsilon_A Y & 0_{n \times n} & -\frac{1}{2}\gamma^2 I_n & 0_{n \times p} \\ 0_{p \times n} & \epsilon_B KY & 0_{p \times n} & 0_{p \times n} & -\frac{1}{2}\gamma^2 I_p \end{bmatrix} \leq 0. \quad (50)$$

We use the similarity transformation, $H = J^{-1}GJ$, to change the basis. Where G is the matrix of (50), and the transformation matrix is

$$J^{-1} := \begin{bmatrix} I_n & 0_{n \times n} & 0_{n \times n} & 0_{n \times n} & 0_{n \times p} \\ 0_{n \times n} & I_n & 0_{n \times n} & 0_{n \times n} & 0_{n \times p} \\ 0_{n \times n} & 0_{n \times n} & 0_{n \times n} & I_n & 0_{n \times p} \\ 0_{p \times n} & 0_{p \times n} & 0_{p \times n} & 0_{p \times n} & I_p \\ 0_{n \times n} & 0_{n \times n} & I_n & 0_{n \times n} & 0_{n \times p} \end{bmatrix}. \quad (51)$$

Thus, the equation (50) is rewritten as:

$$\begin{bmatrix} -Y & A_{cl}Y & 0_{n \times n} & 0_{n \times p} & I_n \\ YA_{cl}^\top & -Y & \epsilon_A Y & \epsilon_B YK^\top & 0_{n \times n} \\ 0_{n \times n} & \epsilon_A Y & -\frac{1}{2}\gamma^2 I_n & 0_{n \times p} & 0_{n \times n} \\ 0_{p \times n} & \epsilon_B KY & 0_{p \times n} & -\frac{1}{2}\gamma^2 I_p & 0_{p \times n} \\ I_n & 0_{n \times n} & 0_{n \times n} & 0_{n \times p} & -I_n \end{bmatrix} \leq 0. \quad (52)$$

With Schur's lemma, the inequality (52) is rewritten as:

$$\begin{bmatrix} I_n - Y & A_{cl}Y & 0_{n \times n} & 0_{n \times p} \\ YA_{cl}^\top & -Y & \epsilon_A Y & \epsilon_B YK^\top \\ 0_{n \times n} & \epsilon_A Y & -\frac{1}{2}\gamma^2 I_n & 0_{n \times p} \\ 0_{p \times n} & \epsilon_B KY & 0_{p \times n} & -\frac{1}{2}\gamma^2 I_p \end{bmatrix} \leq 0. \quad (53)$$

Thus, the \mathcal{H}_∞ in (31b) is finally expressed in (53).

B. LINEAR MATRIX INEQUALITIES

The \mathcal{H}_2 norm of the cost function (31a) can be written with the trace of a matrix

$$\text{Tr}(QZ + RKZK^\top) \quad (54)$$

if there is a matrix $Z = Z^\top > 0$ satisfying Lyapunov equation [45]

$$(\widehat{A} + \widehat{B}K)Z(\widehat{A} + \widehat{B}K)^\top - Z + I_n \leq 0. \quad (55)$$

With (54) and (55), the optimization problem (31) can be written as the following nonconvex semidefinite program if there are matrices $Z = Z^\top > 0$ and $Y = Y^\top > 0$ such that

$$\min_{Y, Z, K} \text{Tr} \left(\begin{bmatrix} Q^{\frac{1}{2}} & 0_{n \times p} \\ 0_{p \times n} & R^{\frac{1}{2}} \end{bmatrix}^\top \begin{bmatrix} Z & 0_{n \times p} \\ 0_{p \times n} & KZK^\top \end{bmatrix} \begin{bmatrix} Q^{\frac{1}{2}} & 0_{n \times p} \\ 0_{p \times n} & R^{\frac{1}{2}} \end{bmatrix} \right) \quad (56a)$$

$$\text{subject to} \begin{bmatrix} I_n - Z & (\widehat{A} + \widehat{B}K)Z \\ Z(\widehat{A} + \widehat{B}K)^\top & -Z \end{bmatrix} \leq 0 \quad (56b)$$

$$\begin{bmatrix} I_n - Y & (\widehat{A} + \widehat{BK})Y & 0_{n \times n} & 0_{n \times p} \\ Y(\widehat{A} + \widehat{BK})^\top & -Y & \epsilon_A Y & \epsilon_B Y K^\top \\ 0_{n \times n} & \epsilon_A Y & -\frac{1}{2}\gamma^2 I_n & 0_{n \times p} \\ 0_{p \times n} & \epsilon_B K Y & 0_{p \times n} & -\frac{1}{2}\gamma^2 I_p \end{bmatrix} \leq 0. \quad (56c)$$

The constraint (55) is converted to (56b) by Schur’s lemma and the \mathcal{H}_∞ constraint (31b) is rewritten with (53) and (56c). The optimization problem involves two constraints (56b) and (56c) with Lyapunov matrices Z and Y respectively. Due to these Lyapunov matrices and control variable K , the optimization problem becomes non-convex.

To recover convexity, we impose the constraint $P := Y = Z \succ 0$ and thus we have a single closed-loop Lyapunov function [59]. The constraint (56b) becomes redundant and thus unnecessary. We can perform the standard variable substitution $L := KP$, and then we obtain the following optimization problem

$$\min_{P,L} \text{Tr} \left(\begin{bmatrix} Q & 0_{n \times p} \\ 0_{p \times n} & R \end{bmatrix} \begin{bmatrix} P & 0_{n \times p} \\ 0_{p \times n} & LP^{-1}L^\top \end{bmatrix} \right) \quad (57a)$$

subject to

$$\begin{bmatrix} I_n - P & \widehat{AP} + \widehat{BL} & 0_{n \times n} & 0_{n \times p} \\ P\widehat{A}^\top + L^\top \widehat{B}^\top & -P & \epsilon_A P & \epsilon_B L^\top \\ 0_{n \times n} & \epsilon_A P & -\frac{1}{2}\gamma^2 I_n & 0_{n \times p} \\ 0_{p \times n} & \epsilon_B L & 0_{p \times n} & -\frac{1}{2}\gamma^2 I_p \end{bmatrix} \leq 0. \quad (57b)$$

If we use the following inequality

$$\begin{bmatrix} U & V \\ V^\top & W \end{bmatrix} - \begin{bmatrix} P & 0_{n \times p} \\ 0_{p \times n} & LP^{-1}L^\top \end{bmatrix} \quad (58a)$$

$$= \begin{bmatrix} U & V \\ V^\top & W \end{bmatrix} - \begin{bmatrix} P \\ L \end{bmatrix} P^{-1} \begin{bmatrix} P & L^\top \end{bmatrix} \quad (58b)$$

$$= \begin{bmatrix} U & V & P \\ V^\top & W & L \\ P & L^\top & P \end{bmatrix} \geq 0, \quad (58c)$$

then the nonlinear component, $LP^{-1}L^\top$ in the cost function (57a) becomes linear. Finally, the convex optimization problem is obtained as

$$\text{minimize } \text{Tr}(QU) + \text{Tr}(RW) \quad (59a)$$

P, L, U, V, W

$$\text{subject to } \begin{bmatrix} U & V & P \\ V^\top & W & L \\ P & L^\top & P \end{bmatrix} \geq 0, \quad (59b)$$

$$\begin{bmatrix} P - I_n & \widehat{AP} + \widehat{BL} & 0_{n \times n} & 0_{n \times p} \\ P\widehat{A}^\top + L^\top \widehat{B}^\top & P & \epsilon_A P & \epsilon_B L^\top \\ 0_{n \times n} & \epsilon_A P & \frac{1}{2}\gamma^2 I_n & 0_{n \times p} \\ 0_{p \times n} & \epsilon_B L & 0_{p \times n} & \frac{1}{2}\gamma^2 I_p \end{bmatrix} \geq 0. \quad (59c)$$

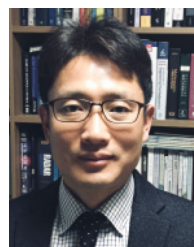
ACKNOWLEDGMENT

An earlier version of this paper was presented in part at the Workshop of the IEEE Intelligent Vehicles Symposium, Jeju, South Korea [60].

REFERENCES

- [1] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems,” *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.
- [2] D. González, J. Pérez, V. Milanés, and F. Nashashibi, “A review of motion planning techniques for automated vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
- [3] B. Paden, M. Cáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [4] C. E. Beal and J. C. Gerdes, “Model predictive control for vehicle stabilization at the limits of handling,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1258–1269, Jul. 2013.
- [5] S. Aradi, “Survey of deep reinforcement learning for motion planning of autonomous vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 2022.
- [6] H. Gao, Z. Kan, and K. Li, “Robust lateral trajectory following control of unmanned vehicle based on model predictive control,” *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 3, pp. 1278–1287, Jun. 2022.
- [7] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, “Autonomous vehicles on the edge: A survey on autonomous vehicle racing,” *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 458–488, 2022.
- [8] A. Arab, K. Yu, J. Yu, and J. Yi, “Motion planning and control of autonomous aggressive vehicle maneuvers,” *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1488–1500, Apr. 2024.
- [9] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, and K. Alexis, “CERBERUS in the DARPA subterranean challenge,” *Sci. Robot.*, vol. 7, no. 66, May 2022, Art. no. eabp9742.
- [10] D. Ball, B. Upcroft, G. Wyeth, P. Corke, A. English, P. Ross, T. Patten, R. Fitch, S. Sukkarieh, and A. Bate, “Vision-based obstacle detection and navigation for an agricultural robot,” *J. Field Robot.*, vol. 33, no. 8, pp. 1107–1130, 2016.
- [11] K. Skonieczny, D. K. Shukla, M. Faragalli, M. Cole, and K. D. Iagnemma, “Data-driven mobility risk prediction for planetary rovers,” *J. Field Robot.*, vol. 36, no. 2, pp. 475–491, Mar. 2019.
- [12] K. A. Q. Caldas, F. M. Barbosa, J. A. R. Silva, T. C. Santos, I. P. Gomes, L. A. Rosero, D. F. Wolf, and V. Grassi, “Autonomous driving of trucks in off-road environment,” *J. Control, Autom. Electr. Syst.*, vol. 34, no. 6, pp. 1179–1193, Dec. 2023.
- [13] S. Hong, G. Park, and Y. Lee, “Robust design and task-priority control for rescue robot HURCULES,” *J. Field Robot.*, vol. 40, no. 5, pp. 1151–1171, Aug. 2023.
- [14] J. Shin, D. Kwak, and T. Lee, “Robust path control for an autonomous ground vehicle in rough terrain,” *Control Eng. Pract.*, vol. 98, May 2020, Art. no. 104384.
- [15] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng, G. Okopal, D. Fox, B. Boots, and A. Shaban, “TerrainNet: Visual modeling of complex terrain for high-speed, off-road navigation,” 2023, *arXiv:2303.15771*.
- [16] S. Thrun, “Stanley: The robot that won the DARPA grand challenge,” *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.
- [17] L. B. Cremean, T. B. Foote, J. H. Gillula, G. H. Hines, D. Kogan, K. L. Kriebbaum, J. C. Lamb, J. Leibs, L. Lindzey, and C. E. Rasmussen, “Alice: An information-rich autonomous vehicle for high-speed desert navigation,” *J. Field Robot.*, vol. 23, no. 9, pp. 777–810, 2006.
- [18] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, “Real-time motion planning with applications to autonomous urban driving,” *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Sep. 2009.
- [19] H. Wang, B. Liu, X. Ping, and Q. An, “Path tracking control for autonomous vehicles based on an improved MPC,” *IEEE Access*, vol. 7, pp. 161064–161073, 2019.
- [20] Y. Jeong and S. Yim, “Model predictive control-based integrated path tracking and velocity control for autonomous vehicle with four-wheel independent steering and driving,” *Electronics*, vol. 10, no. 22, p. 2812, Nov. 2021.
- [21] J. Cao, C. Song, S. Peng, S. Song, X. Zhang, and F. Xiao, “Trajectory tracking control algorithm for autonomous vehicle considering cornering characteristics,” *IEEE Access*, vol. 8, pp. 59470–59484, 2020.

- [22] H. Wu, Z. Si, and Z. Li, "Trajectory tracking control for four-wheel independent drive intelligent vehicle based on model predictive control," *IEEE Access*, vol. 8, pp. 73071–73081, 2020.
- [23] Z. Li, P. Wang, H. Liu, Y. Hu, and H. Chen, "Coordinated longitudinal and lateral vehicle stability control based on the combined-slip tire model in the MPC framework," *Mech. Syst. Signal Process.*, vol. 161, Dec. 2021, Art. no. 107947.
- [24] Y. Zhang, Y. Lin, Y. Qin, M. Dong, L. Gao, and E. Hashemi, "A new adaptive cruise control considering crash avoidance for intelligent vehicle," *IEEE Trans. Ind. Electron.*, vol. 71, no. 1, pp. 688–696, Jan. 2024.
- [25] Y. Zhang, M. Xu, Y. Qin, M. Dong, L. Gao, and E. Hashemi, "MILE: Multi-objective integrated model predictive adaptive cruise control for intelligent vehicle," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 8539–8548, Jul. 2023.
- [26] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Syst. Mag.*, vol. 36, no. 6, pp. 30–44, Dec. 2016.
- [27] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 3, no. 1, pp. 269–296, May 2020.
- [28] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3313–3323, Sep. 2019.
- [29] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optim. Control Appl. Methods*, vol. 36, no. 5, pp. 628–647, Sep. 2015.
- [30] S. Hong, K. Jang, S. Kim, and J. Park, "Regularized hierarchical quadratic program for real-time whole-body motion generation," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 4, pp. 2115–2126, Aug. 2021.
- [31] J. Seo, T. Kim, K. Kwak, J. Min, and I. Shim, "ScATE: A scalable framework for Self-supervised traversability estimation in unstructured environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 888–895, Feb. 2023.
- [32] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1603–1622, Dec. 2018.
- [33] T. Kim, G. Park, K. Kwak, J. Bae, and W. Lee, "Smooth model predictive path integral control without smoothing," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10406–10413, Oct. 2022.
- [34] T. Han, A. Liu, A. Li, A. Spitzer, G. Shi, and B. Boots, "Model predictive control for aggressive driving over uneven terrain," 2023, *arXiv:2311.12284*.
- [35] T. Kim, H. Lee, S. Hong, and W. Lee, "TOAST: Trajectory optimization and simultaneous tracking using shared neural network dynamics," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9747–9754, Oct. 2022.
- [36] R. M. Murray, J. Hauser, A. Jadbabaie, M. B. Milam, N. Petit, W. B. Dunbar, and R. Franz, "Online control customization via optimization-based control," in *Software-Enabled Control: Information Technology for Dynamical Systems*. Hoboken, NJ, USA: Wiley, 2003, pp. 149–174.
- [37] M. B. Milam, "Real-time optimal trajectory generation for constrained dynamical systems," Ph.D. dissertation, Dept. Eng. Appl. Sci., California Inst. Technol., CA, USA, 2003.
- [38] S. Hong and K. D. Cho, "Kinematic algorithms and robust controller design for inertially stabilized system," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 76–87, Feb. 2014.
- [39] N. Matni, A. D. Ames, and J. C. Doyle, "Towards a theory of control architecture: A quantitative framework for layered multi-rate control," 2024, *arXiv:2401.15185*.
- [40] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, "System level synthesis," *Annu. Rev. Control*, vol. 47, pp. 364–393, 2019.
- [41] Y.-S. Wang, N. Matni, and J. C. Doyle, "A system-level approach to controller synthesis," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4079–4093, Oct. 2019.
- [42] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Found. Comput. Math.*, vol. 20, no. 4, pp. 633–679, Aug. 2020.
- [43] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," Robot. Inst., Pittsburgh, PA, USA, Tech. Rep. CMU-RITR-09-08, 2009.
- [44] K. J. Åström and R. M. Murray, *Feedback Systems: Introduction for Scientists Engineers*. Princeton, NJ, USA: Princeton Univ. Press, 2021.
- [45] K.-Z. Liu and Y. Yao, *Robust Control: Theory and Applications*. Hoboken, NJ, USA: Wiley, 2016.
- [46] J. C. Doyle, N. Matni, Y.-S. Wang, J. Anderson, and S. Low, "System level synthesis: A tutorial," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 2856–2867.
- [47] P. J. Antsaklis and A. N. Michel, *Linear Systems*. Basel, Switzerland: Birkhäuser, 2005.
- [48] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. Chelmsford, MA, USA: Courier Corporation, 2013.
- [49] J. Anderson and N. Matni, "Structured state space realizations for SLS distributed controllers," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 982–987.
- [50] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15. Philadelphia, PA, USA: SIAM, 1994.
- [51] A. P. Engelbrecht, *Computational Intelligence: An Introduction*. Hoboken, NJ, USA: Wiley, 2007.
- [52] F. vandenBergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.
- [53] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1168–1175.
- [54] S. Hong, G. Park, Y. Lee, W. Lee, B. Choi, O. Sim, and J.-H. Oh, "Development of a tele-operated rescue robot for a disaster response," *Int. J. Humanoid Robot.*, vol. 15, no. 4, Aug. 2018, Art. no. 1850008.
- [55] M. Grant and S. Boyd. (2014). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1*. [Online]. Available: <https://cvxr.com/cvx/>
- [56] A. Agrawal, R. Verschuere, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *J. Control Decis.*, vol. 5, no. 1, pp. 42–60, Jan. 2018.
- [57] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 1094–1099.
- [58] F. Najson, "The Kalman-Yakubovich-Popov lemma for discrete-time positive linear systems," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 5188–5193.
- [59] C. Scherer, P. Gahinet, and M. Chilali, "Multiobjective output-feedback control via LMI optimization," *IEEE Trans. Autom. Control*, vol. 42, no. 7, pp. 896–911, Jul. 1997.
- [60] S. Hong and G. Park, "Robust path tracking control for off-road autonomous vehicle via system level synthesis," in *Proc. IEEE Intell. Vehicles Symp.*, Jeju Island, South Korea, Jun. 2024, pp. 213–218.



SEONGIL HONG received the B.S. degree in mechanical engineering from Pusan National University, in 2002, the M.S. degree in mechanical engineering from POSTECH, South Korea, in 2004, and the Ph.D. degree in mechanical and aerospace engineering from Seoul National University, South Korea, in 2021. From 2004 to 2006, he was with Hyundai-Kia Motor Company, Hwasung, South Korea, where he was a Research Engineer engaged in noise, vibration, and harshness problems. In 2006, he joined the Agency for Defense Development, South Korea, where he is currently a Principal Researcher. His research interests include control, robotics, autonomous systems, and machine learning.



GYUHYUN PARK received the B.Sc. and M.Sc. degrees in mechanical and aerospace engineering from Seoul National University, Seoul, South Korea, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree. Since 2014, he has been a Senior Researcher with the Agency for Defense Development, Daejeon, South Korea. His research interests include control, robotics, and machine learning.