



Model-predictive active steering and obstacle avoidance for autonomous ground vehicles

Yongsoon Yoon^a, Jongho Shin^b, H. Jin Kim^{b,*}, Yongwoon Park^c, Shankar Sastry^d

^a Powertrain, Gasoline System Continental, Seoul, Korea

^b School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Korea

^c Agency for Defense Development, Daejeon, Korea

^d Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA

ARTICLE INFO

Article history:

Received 7 January 2008

Accepted 1 December 2008

Available online 15 January 2009

Keywords:

Model-predictive control

Collision avoidance

Unmanned ground vehicles (UGVs)

Potential function

Trajectory generation

ABSTRACT

This paper presents a model-predictive approach for trajectory generation of unmanned ground vehicles (UGVs) combined with a tire model. An optimal tracking problem while avoiding collision with obstacles is formulated in terms of cost minimization under constraints. Information on obstacles is incorporated online in the nonlinear model-predictive framework as they are sensed within a limited sensing range. The overall problem is solved online with nonlinear programming. For the local path regeneration upon detecting new obstacles, the cost function is augmented using the obstacle information in two methods. The first method uses the distance from the UGV to the nearest detected obstacle, and the second method uses the parallax information from the vehicle about the detected obstacles. Simulation results in cluttered and dynamic environments show that the modified parallax method effectively reflects the threat of the obstacles to the UGV considering the dimension and state variables of the vehicle, showing clear improvements over the distance-based methods.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Various military and civil applications often require UGVs (unmanned ground vehicles) to move autonomously in unknown environments under dynamic and physical constraints, rather than to simply follow a pre-planned path designated by an offline mission-level planning algorithm.

This paper studies an active steering problem of a UGV with obstacle avoidance in complex environments, using a model-predictive method. In order to solve a dynamic trajectory generation problem of the UGV shown in Fig. 1, a bicycle model is used combined with a nonlinear tire-force model to predict the future evolution of the system.

Much research on local obstacle avoidance has been performed, most of which use reactive methods based on sensor data, such as Rimón and Koditschek (1992), Spenko, Jagnemma, and Dubowsky (2004), Fox, Burgard, and Thrun (1997), Simmon (1996) and Minguez and Montano (2000). Some take into account the simple robot dynamics in terms of velocity or

turning radius (Ulrich & Borenstein, 1998). These approaches are computationally efficient, but the vehicle can get stuck in local minima, sometimes the discretization of the world is required, or incorporation of the complicated dynamics is not easy. In Ulrich and Borenstein (2000), A* search was used on a trajectory tree of a fixed depth, and in Arras, Persson, Tomatis, and Siegwart (2002), the dimension of the vehicle was considered with a reduced dynamic window. But as will be shown later, consideration of the location of center of gravity (CG) of the vehicle may not suffice for obstacle avoidance in a cramped area.

Recently, predictive active steering control for autonomous vehicle systems was studied (Borrelli, Falcone, Keviczky, Asgari, & Hrovat, 2005; Falcone, Borrelli, Asgari, Tseng, & Hrovat, 2006; Falcone, Borrelli, Asgari, Tseng, & Hrovat, 2007), with a tire model (Bakker, Nyborg, & Pacejka, 1987), where the autonomous vehicle was directed to follow the given reference which is assumed to be collision-free and achievable by the vehicle. However, in the case of unknown environments, it is difficult to acquire such a safe pre-defined reference. In addition, because the dimension of the vehicle is not explicitly considered, if the environment is complex and cluttered, it can be a serious issue. Also the previous approach might be limited in the implementation point of view, because of the computation time required for optimization.

* Corresponding author. Tel.: +82 2 880 7392; fax: +82 2 887 2662.

E-mail addresses: yongsoon.yoon@continental-corporation.com (Y. Yoon), jh0524@snu.ac.kr (J. Shin), hjinkim@snu.ac.kr (H.J. Kim), woon5901@hanafos.com (Y. Park), sastry@eecs.berkeley.edu (S. Sastry).



Fig. 1. UGV in consideration, its mass, momentum of inertia, length and width are 807 kg, 429.649 kg m², 2.150 m, and 1.290 m, respectively.

This paper considers the online obstacle avoidance as well as navigation toward the destination using a model-predictive approach. In this framework, the controller predicts a future path and solves an optimization problem to plan collision-free trajectories. Model-predictive control (MPC) has been applied to control various systems, including various industrial systems (Qin & Badgwell, 2003). It has also been used to generate safe trajectories for unmanned aerial vehicles (Kim, Shim, & Sastry, 2002; Kim & Shim, 2003) using simplified dynamics in unknown environments. But when compared to the settings considered there, UGVs are often employed in more cluttered environments. The parallax-based method proposed in this paper is a unique and efficient way of incorporating the requirements of considering speed, dimension and actual moving direction in the nonlinear model-predictive framework. With growing utilization of mechatronic components in automobiles (Iserrmann, 2008), the parallax-based formulation can contribute as a part of a collision avoidance mechanism that can improve the limitations of common approaches purely based on distance from objects in concern.

Section 2 describes a nonlinear UGV model combined with a tire model. The basic MPC formulation and algorithm are presented in Section 3. In Section 4, for local path generation with a limited sensing range, distance-based and parallax-based approaches are proposed. Section 5 presents simulation results to validate proposed algorithms, and Section 6 contains concluding remarks.

2. UGV model

The following subscripts are used for the variables throughout this paper:

| | |
|--------|---------------------------------------------|
| (-)f | front wheel |
| (-)r | rear wheel |
| (-)x | x-axis of the robot's body coordinate frame |
| (-)y | y-axis of the robot's body coordinate frame |
| (-)X | x-axis of the inertial reference frame |
| (-)Y | y-axis of the inertial reference frame |
| (-)s | side (lateral) direction |
| (-)l | longitudinal direction |
| (-)ref | reference signals |

2.1. Vehicle dynamics

In this paper, a bicycle model simplified from four-wheel dynamics is considered in combination with a tire model. In this

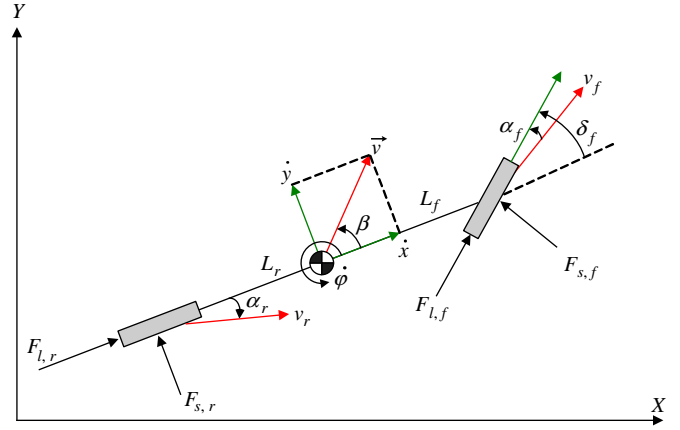


Fig. 2. A schematic diagram of the bicycle robot model.

bicycle model, the motion of the vehicle is a function of the forces F_x , F_y , and momentum M_z exerted on the CG of the UGV (Wollherr, 2002). Eq. (1) describes the dynamics of the vehicle, where m and I_{zz} are the mass and the inertia moment of the vehicle, respectively. As shown in Fig. 2, β is the direction of the velocity with respect to the x-axis of the vehicle's body coordinate frame whose origin coincides with the CG of the vehicle. φ is the yaw angle in the inertial frame and v is the speed of the CG of the vehicle:

$$\begin{aligned} mv(\dot{\beta} + \dot{\varphi}) &= -F_x \sin \beta + F_y \cos \beta, \\ m\dot{v} &= F_x \cos \beta + F_y \sin \beta, \\ I_{zz}\ddot{\varphi} &= M_z. \end{aligned} \quad (1)$$

Setting up equations for the equilibrium of forces and momentum in Fig. 2 yields the geometric relation of Eq. (2), where δ_f is the front steering angle. F_{sf} , F_{sr} , F_{lf} , and F_{lr} are side and longitudinal tire forces exerted on each tire. L_f and L_r are lengths from the CG of the vehicle to the front and rear wheels, respectively:

$$\begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} = \begin{bmatrix} -\sin \delta_f & 0 \\ \cos \delta_f & 1 \\ L_f \cos \delta_f & -L_r \end{bmatrix} \begin{bmatrix} 2F_{sf} \\ 2F_{sr} \end{bmatrix} + \begin{bmatrix} \cos \delta_f & 1 \\ \sin \delta_f & 0 \\ L_f \sin \delta_f & 0 \end{bmatrix} \begin{bmatrix} 2F_{lf} \\ 2F_{lr} \end{bmatrix}. \quad (2)$$

Longitudinal and side tire forces F_{l*} and F_{s*} for each tire are given as nonlinear functions of the slip angle (α), the slip ratio (s), the road friction coefficient (μ), and the total vertical load F_z (detailed in Section 2.2), where the slip ratio s is defined as

$$s = \begin{cases} \frac{rw}{v_l} - 1 & \text{if } v_l > rw, \quad v_l \neq 0 \text{ for breaking,} \\ 1 - \frac{v_l}{rw} & \text{if } v_l < rw, \quad w \neq 0 \text{ for driving.} \end{cases} \quad (3)$$

v_l is the longitudinal velocity of the center of the tire, r is the effective radius of the tire, and w is the angular velocity of the tire. The front and rear tire slip angles α_f, α_r are the functions of $\beta, \dot{\varphi}, v$ as

$$\begin{aligned} \alpha_f &= \delta_f - \tan^{-1} \left(\frac{v \sin \beta + L_f \dot{\varphi}}{v \cos \beta} \right), \\ \alpha_r &= -\tan^{-1} \left(\frac{v \sin \beta - L_r \dot{\varphi}}{v \cos \beta} \right). \end{aligned} \quad (4)$$

The vertical load F_z is distributed between the front and rear wheels depending on the geometry of the vehicle:

$$F_{zf} = \frac{L_r mg}{2(L_f + L_r)}, \quad F_{zr} = \frac{L_f mg}{2(L_f + L_r)}. \quad (5)$$

The motion of the CG of the vehicle in the inertial frame is described by

$$\begin{aligned}\dot{X}_{CG} &= v \cos \beta \cos \varphi - v \sin \beta \sin \varphi, \\ \dot{Y}_{CG} &= v \cos \beta \sin \varphi + v \sin \beta \cos \varphi.\end{aligned}\quad (6)$$

2.2. Tire model

For the tire force terms in Eq. (2), the tire model described by a Pacejka's MAGIC formula (Bakker et al., 1987) are used throughout this paper. This model represents a complex semi-empirical relationship, which takes into consideration the interaction between the longitudinal tire force and the lateral tire force in combined braking and steering conditions. Figs. 3(a) and (b), respectively, show the longitudinal tire force with respect to the slip ratio s and the lateral tire force corresponding to the front-wheel slip angle α_f with different values of F_z . It can be seen that the tire force relationships can be approximated by hyperbolic tangent functions of the slip ratio and tire slip angle, respectively.

2.3. UGV model

In order to reflect the dynamic characteristics of the UGV using a computationally efficient model, the following

simplifications have been made to the nonlinear bicycle model:

$$\begin{aligned}(1) \quad \beta &\ll 1 & \Rightarrow & \sin \beta \approx \beta, \cos \beta \approx 1 \\ (2) \quad \dot{v} &= 0 \text{ and } F_l = 0 & \Rightarrow & F_x = 0 \\ (3) \quad \delta_f &\ll 1 & \Rightarrow & \sin \delta_f \approx \delta_f, \cos \delta_f \approx 1\end{aligned}$$

With the above assumptions and the state and control input vectors $\xi = [\beta, \varphi, \dot{\varphi}, \delta_f, X_{CG}, Y_{CG}]^T$ and $u = \delta_f$, and the output vector η , Eqs. (1), (2), (4) and (6) can be rewritten as the following partially nonlinear affine model:

$$\begin{aligned}\dot{\xi} &= f(\xi, u) = \begin{bmatrix} -\dot{\varphi} + \frac{2F_{sf} + 2F_{sr} + 2\delta_f F_{lf}}{mv} \\ \dot{\varphi} \\ \frac{2L_f F_{sf} - 2L_r F_{sr} + 2L_f \delta_f F_{lf}}{I_{zz}} \\ u \\ v \cos \varphi - v \beta \sin \varphi \\ v \sin \varphi + v \beta \cos \varphi \end{bmatrix} \\ \eta &= h(\xi) = [X_{CG}, Y_{CG}]^T.\end{aligned}\quad (7)$$

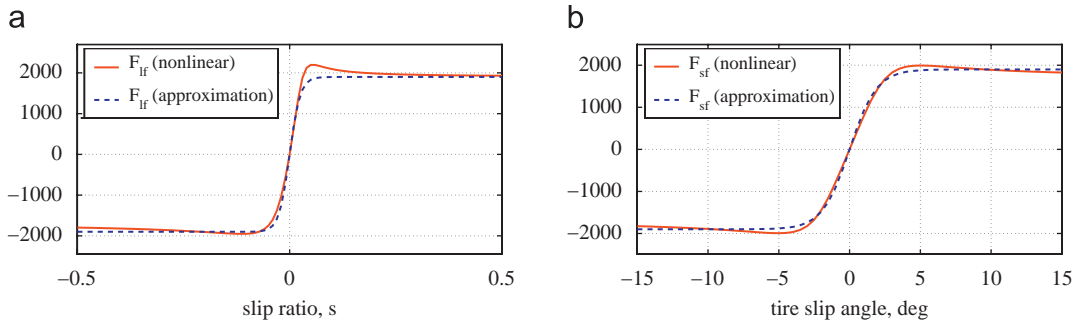


Fig. 3. (a) Longitudinal tire force F_l corresponding to the slip ratio with zero slip angle. (b) Lateral tire force F_s corresponding to the slip angle at zero slip ratio and varying load F_z , respectively. Dashed curves represent the approximations by tanh(·).

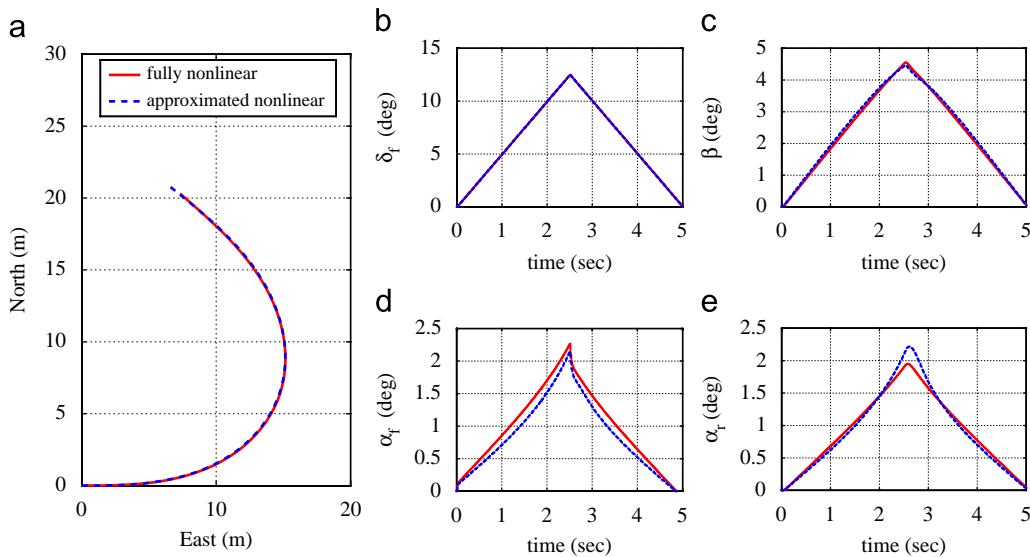


Fig. 4. Comparison of full-nonlinear and partially nonlinear models at 7 m/s. (a) Trajectory of the fully nonlinear model (red solid curve) and partially nonlinear model (blue dashed curve). (b) Front steering angle commands applied for 5 s. (c)–(e) are corresponding β , α_f , α_r , for the both models, respectively (solid: fully nonlinear model, dashed: partially nonlinear model).

2.4. Model comparison

For validation, the simplified model (Eq. (7)) was compared against the fully nonlinear bicycle model (Eqs. (1)–(6)) combined with the nonlinear lateral tire model (Fig. 3). In Fig. 4, the front steering rate of $5^\circ/\text{s}$ is for 2.5 s, and for the next 2.5 s, $-5^\circ/\text{s}$ is applied for pure and partial nonlinear bicycle model with identical initial conditions. In spite of the partial linearization, the dynamic characteristics remain similar.

3. Model-predictive algorithm

3.1. Basic model-predictive algorithm

For discrete-time domain implementation (Kim et al., 2002), the UGV dynamic equations of Eq. (7) are discretized as

$$\begin{aligned}\xi_{t+1} &= f^{\text{dt}}(\xi_t, u_t), \\ \eta_{t+1} &= h(\eta_t).\end{aligned}\quad (8)$$

Given the current state $\xi_{0,t} = \xi_t$, MPC is to compute the optimal control sequence $[u_{k,t}^*]_{k=0}^{N-1}$ with the receding horizon principle at each time step t , which solves the following:

$$[u_{k,t}^*]_{k=0}^{N-1} = \text{argmin}(J(\xi_{0,t}, [u_{k,t}]_{k=0}^{N-1})) \quad (9)$$

$$\text{subject to } h(\xi_{k,t}, u_{k,t}) = \xi_{k+1,t} - f^{\text{dt}}(\xi_{k,t}, u_{k,t}) = 0, \quad (10)$$

$$g_u(u_{k,t}) = |u_{k,t}| - u_{\text{sat}} \leq 0, \quad (11)$$

$$g_{\delta_f}(\delta_{f,k,t}) = |\delta_{f,k,t}| - \delta_{f,\text{sat}} \leq 0, \quad (12)$$

$$g_{\alpha_f}(\alpha_{f,k,t}) = |\alpha_{f,k,t}| - \alpha_{f,\text{sat}} \leq 0, \quad (13)$$

$$g_{\alpha_r}(\alpha_{r,k,t}) = |\alpha_{r,k,t}| - \alpha_{r,\text{sat}} \leq 0, \quad (14)$$

where $k = 0, \dots, N-1$.

Here t is the current time, N is the length of the look-ahead horizon and k is the time step within the look-ahead horizon at each t . Eq. (10) is the discretized UGV dynamics. The inequality constraints in Eqs. (11) and (12) reflect the limit on u and δ_f . Eqs. (13) and (14) are for the preference on the smaller tire slip angles.

The following cost function has been considered for tracking a given reference trajectory (Mayne, Rawlings, Rao, & Scokaert, 2000):

$$J(\xi_t, u_t) = \phi(\tilde{\eta}_{N,t}) + \sum_{k=0}^{N-1} L(\tilde{\eta}_{k,t}, u_{k,t}), \quad (15)$$

$$\phi(\tilde{\eta}_{N,t}) = \frac{1}{2} \tilde{\eta}_{N,t}^T P \tilde{\eta}_{N,t}, \quad (16)$$

$$L(\tilde{\eta}_{k,t}, u_{k,t}) = \frac{1}{2} \tilde{\eta}_{k,t}^T Q \tilde{\eta}_{k,t} + \frac{1}{2} u_{k,t}^T R u_{k,t}. \quad (17)$$

$\tilde{\eta}_{k,t}$ is defined as $\tilde{\eta}_{k,t} = \eta_{\text{ref},k,t} - \eta_{k,t}$, where $\eta_{\text{ref},k,t}$ is the reference trajectory. P , Q , and R are constant weighting matrices. Eq. (16) penalizes the deviation at the final stage. The first term of Eq. (17) is the penalty on the deviation from the desired reference, and the second term penalizes the large control signal in the view of the minimal energy consumption.

The equality and inequality constraints are incorporated into the augmented cost function as follows:

$$\begin{aligned}J_{\text{aug}}(\xi_t, u_t) &= \phi(\tilde{\eta}_{N,t}) + \sum_{k=0}^{N-1} L(\tilde{\eta}_{k,t}, u_{k,t}) + \lambda_k h(\xi_{k,t}, u_{k,t}) \\ &\quad + \mu_u G_u(u_{k,t}) + \mu_{\delta_f} G_{\delta_f}(\delta_{f,k,t}) + \mu_{\alpha_f} G_{\alpha_f}(\alpha_{f,k,t}) \\ &\quad + \mu_{\alpha_r} G_{\alpha_r}(\alpha_{r,k,t}),\end{aligned}\quad (18)$$

where $\mu_u, \mu_{\delta_f}, \mu_{\alpha_f}, \mu_{\alpha_r}$ are weighting parameters for the saturation constraints in KKT (Karush–Kuhn–Tucker) conditions. From Eqs. (11) to (14), $G_u, G_{\delta_f}, G_{\alpha_f}, G_{\alpha_r}$ are determined as

$$G_u(u_{k,t}) = \begin{cases} \frac{1}{2} g_u(u_{k,t})^2 & \text{if } g_u(u_{k,t}) > 0, \\ 0 & \text{if } g_u(u_{k,t}) \leq 0, \end{cases} \quad (19)$$

$$G_{\delta_f}(\delta_{f,k,t}) = \begin{cases} \frac{1}{2} g_{\delta_f}(\delta_{f,k,t})^2 & \text{if } g_{\delta_f}(\delta_{f,k,t}) > 0, \\ 0 & \text{if } g_{\delta_f}(\delta_{f,k,t}) \leq 0, \end{cases} \quad (20)$$

$$G_{\alpha_f}(\alpha_{f,k,t}) = \begin{cases} \frac{1}{2} g_{\alpha_f}(\alpha_{f,k,t})^2 & \text{if } g_{\alpha_f}(\alpha_{f,k,t}) > 0, \\ 0 & \text{if } g_{\alpha_f}(\alpha_{f,k,t}) \leq 0, \end{cases} \quad (21)$$

$$G_{\alpha_r}(\alpha_{r,k,t}) = \begin{cases} \frac{1}{2} g_{\alpha_r}(\alpha_{r,k,t})^2 & \text{if } g_{\alpha_r}(\alpha_{r,k,t}) > 0, \\ 0 & \text{if } g_{\alpha_r}(\alpha_{r,k,t}) \leq 0. \end{cases} \quad (22)$$

With the augmented cost function (Eqs. (18)–(22)), the online optimization is performed using the gradient search (Kim et al., 2002; Sutton & Bitmead, 2000).

3.2. Modification for trajectory generation

In the setting considered in this paper, UGVs have limited information containing only the start and goal location but not the location of obstacles. They are demanded to navigate from the start to their goal without any collision using sensor data only measured by sensor mounted on the UGV. It is assumed that UGVs can estimate their own state variables, control input commands and output variables precisely.

Fig. 5 shows the trajectory generation procedure which consists of global and local path planning. Initially, the UGV starts with a known goal and start location, and generates the global trajectory offline. On the way to the goal point, if sensors detect obstacles, the UGV regenerates a local trajectory avoiding obstacles online. Additional cost-function terms used for this local path computation will be treated in Section 4.

For simplicity, the shortest straight line connecting the start location $[X_{CG,0,0}, Y_{CG,0,0}]$ and the destination $[X_{CG,f}, Y_{CG,f}]$ was used as the reference trajectory $\eta_{\text{ref},k,t} = [X_{CG,k,t,\text{ref}}, Y_{CG,k,t,\text{ref}}]$. Specifying $\eta_{\text{ref},k,t}$ as fixed functions of time is not preferable because deviation from those values can be accumulated due to newly detected obstacles. At each time t , starting from the current position $\eta_{0,t}$, $\eta_{k,t} = [X_{CG,k,t}, Y_{CG,k,t}]^T$ for $k = 1, \dots, N-1$ is predicted using the control input sequence obtained from the previous

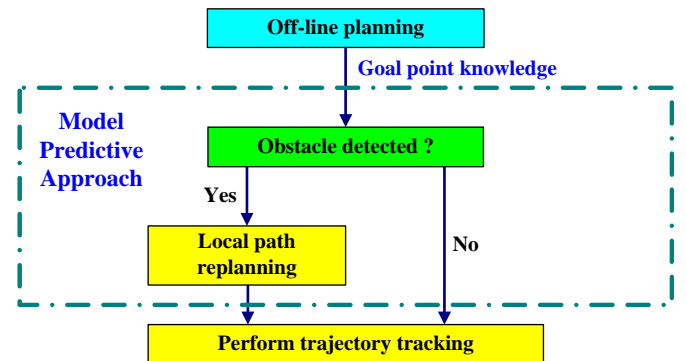


Fig. 5. Overview of trajectory generation architecture.

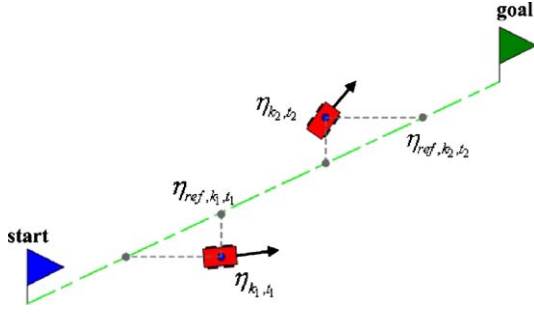


Fig. 6. Reference location defined on the shortest straight line corresponding to the current CG of the vehicle.

optimization step. Then $\eta_{ref,k,t}$ is obtained as the point on the straight line with the same X or Y coordinate as $\eta_{k,t}$, depending on which is closer to the goal location, as shown in Fig. 6.

In order to promote the UGV to advance toward the destination rather than retreating due to the hazard, the following term is added to Eq. (18):

$$PF_{goal,k,t} = \frac{1}{2} K_{goal} d_{goal,k,t}^2, \quad (23)$$

where K_{goal} is the weighting parameter, and $d_{goal,k,t} = \|\eta_{CG,k,t} - \eta_{goal}\|$ is the distance between $\eta_{CG,k,t}$ and $\eta_{goal} = [X_{CG,f}, Y_{CG,f}]^T$. The difference of this approach and the traditional potential-field approaches lies in the optimization in the receding horizon framework. This fact will result in the large differences in the overall performances, as will be seen in Section 5.

4. Local trajectory regeneration

This section presents two approaches for adjusting Eq. (18) in order to accomplish obstacle avoidance using the sensor information of a limited range. In the first approach, the minimum distance between the vehicle and obstacles is used, weighted with the vehicle speed. In the second approach, consideration of the dimension of the vehicle is made by a clever use of parallax information.

4.1. Weighted-distance-based approach

In this distance-based approach, the potential-like cost functions using the information of the distance to the detected obstacles from the vehicle are constructed and added to Eq. (18) to generate the repulsive field from the obstacles.

Let $d_{obs,k,t,i}$ denote the distance between the UGV surface and the i th detected obstacle at time (k,t) . As an extension of the traditional approach, consideration of the speed of UGV is also made as the following form:

$$PF_{obs,k,t} = K_{obs} \frac{v}{\min_i(d_{obs,k,t,i}) + \varepsilon}, \quad (24)$$

where K_{obs} is a weighting parameter, and ε is a small positive constant for non-singularity. Note that, the dimension of the vehicle is indirectly considered, but the actual moving direction of the vehicle is not considered.

4.2. Modified-parallax-based approach

Humans perceive the distance from an object with parallax, i.e. the angle between the straight lines from two different

observation points to the object. This principle is applied here, with modification to reflect the actual moving direction of the UGV.

Suppose that at time (k,t) , obstacles i and j are detected within the sensing range and their coordinates are $[p_{X,k,t,i}, p_{Y,k,t,i}]$ and $[q_{X,k,t,j}, q_{Y,k,t,j}]$. Then, $(p_{X,k,t,i}, p_{Y,k,t,i})$, the coordinates of the obstacle i in the vehicle coordinate frame are determined as follows:

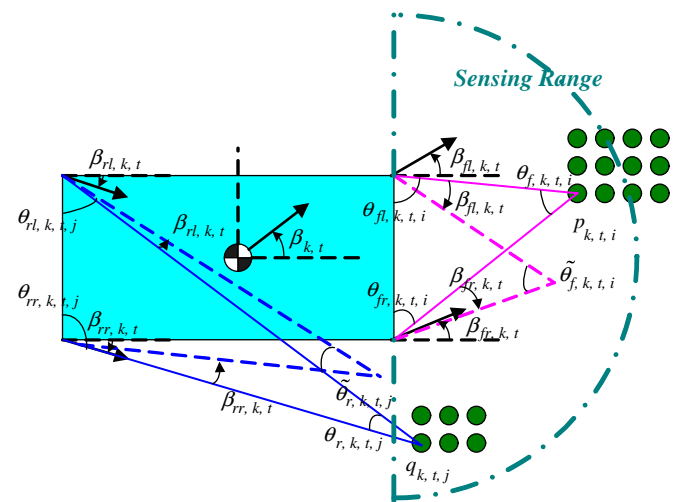
$$\begin{aligned} p_{X,k,t,i} &= (p_{X,k,t,i} - X_{CG,k,t}) \cos \varphi_{k,t} + (p_{Y,k,t,i} - Y_{CG,k,t}) \sin \varphi_{k,t}, \\ p_{Y,k,t,i} &= (p_{Y,k,t,i} - Y_{CG,k,t}) \cos \varphi_{k,t} + (-p_{X,k,t,i} + X_{CG,k,t}) \sin \varphi_{k,t}. \end{aligned} \quad (25)$$

$(q_{X,k,t,j}, q_{Y,k,t,j})$ are computed similarly from $[q_{X,k,t,j}, q_{Y,k,t,j}]$.

Now, as shown in Fig. 7, suppose that $\theta_{fl,k,t,i}$ and $\theta_{fr,k,t,i}$ ($\theta_{rl,k,t,j}$ and $\theta_{rr,k,t,j}$) denote the angles between the front (rear) face and the lines connecting the obstacle i (j) with each vertex of the vehicle. And let $\beta_{fl,k,t}$, $\beta_{fr,k,t}$, $\beta_{rl,k,t}$, and $\beta_{rr,k,t}$ denote the moving direction of four vertices of the UGV. They can be calculated as follows:

$$\begin{aligned} \theta_{fl,k,t,i} &= \tan^{-1} \frac{p_{X,k,t,i} - \frac{L}{2}}{\frac{W}{2} - p_{Y,k,t,i}}, & \theta_{fr,k,t,i} &= \tan^{-1} \frac{p_{X,k,t,i} - \frac{L}{2}}{\frac{W}{2} + p_{Y,k,t,i}}, \\ \theta_{rl,k,t,j} &= \tan^{-1} \frac{q_{X,k,t,j} + \frac{L}{2}}{\frac{W}{2} - q_{Y,k,t,j}}, & \theta_{rr,k,t,j} &= \tan^{-1} \frac{q_{X,k,t,j} + \frac{L}{2}}{\frac{W}{2} + q_{Y,k,t,j}}, \\ \beta_{fl,k,t} &= \tan^{-1} \frac{v \sin \beta_{k,t} + \frac{L}{2} \dot{\varphi}_{k,t}}{v \cos \beta_{k,t} - \frac{W}{2} \dot{\varphi}_{k,t}}, & \beta_{fr,k,t} &= \tan^{-1} \frac{v \sin \beta_{k,t} + \frac{L}{2} \dot{\varphi}_{k,t}}{v \cos \beta_{k,t} + \frac{W}{2} \dot{\varphi}_{k,t}}, \\ \beta_{rl,k,t} &= \tan^{-1} \frac{v \sin \beta_{k,t} - \frac{L}{2} \dot{\varphi}_{k,t}}{v \cos \beta_{k,t} - \frac{W}{2} \dot{\varphi}_{k,t}}, & \beta_{rr,k,t} &= \tan^{-1} \frac{v \sin \beta_{k,t} - \frac{L}{2} \dot{\varphi}_{k,t}}{v \cos \beta_{k,t} + \frac{W}{2} \dot{\varphi}_{k,t}}. \end{aligned} \quad (26)$$

Then, front and rear modified parallax (MP) angles, defined as Eq. (27), incorporate the moving direction of four vertices



$p_{k,t,i}$: the i th obstacle making $\theta_{f,k,t,i}$ at time t .

$q_{k,t,j}$: the j th obstacle making $\theta_{r,k,t,j}$ at time t .

Fig. 7. Modified parallax (MP) angles $\tilde{\theta}_{f,k,t,i}$ and $\tilde{\theta}_{r,k,t,j}$, defined about the detected obstacles located at $p_{k,t,i}$ and $q_{k,t,j}$ with respect to the front and rear of the vehicle.

of the UGV:

if obstacles are detected within sensing range

$$\left\{ \begin{array}{ll} \text{if} & 0 < (\theta_{fl,k,t,i} - \beta_{fl,k,t}) + (\theta_{fr,k,t,i} + \beta_{fr,k,t}) < \pi, \\ & \tilde{\theta}_{f,k,t,i} = \pi - [(\theta_{fl,k,t,i} - \beta_{fl,k,t}) + (\theta_{fr,k,t,i} + \beta_{fr,k,t})] \\ \text{else} & \\ & \tilde{\theta}_{f,k,t,i} = 0, \\ \text{if} & 0 < (\theta_{rl,k,t,j} - \beta_{rl,k,t}) + (\theta_{rr,k,t,j} + \beta_{rr,k,t}) < \pi, \\ & \tilde{\theta}_{r,k,t,j} = \pi - [(\theta_{rl,k,t,j} - \beta_{rl,k,t}) + (\theta_{rr,k,t,j} + \beta_{rr,k,t})] \\ \text{else} & \\ & \tilde{\theta}_{r,k,t,j} = 0 \end{array} \right.$$

else

$$\tilde{\theta}_{f,k,t,i} = \tilde{\theta}_{r,k,t,i} = 0. \quad (27)$$

Then with $\tilde{\theta}_{f,k,t,i}$ and $\tilde{\theta}_{r,k,t,j}$ computed by Eq. (27) for each detected obstacle, the obstacles with the largest MP values are considered using the following function:

$$PF_{obs,k,t,i,j} = K_{obs} \exp \left(\max_i \frac{\tilde{\theta}_{f,k,t,i}}{\theta_{f,c}(v)} + \max_j \frac{\tilde{\theta}_{r,k,t,j}}{\theta_{r,c}(v)} \right), \quad (28)$$

where $\theta_{f,c}$ and $\theta_{r,c}$ are the critical MP values defined as

$$\theta_{f,c}(v) = \frac{K_{MPf}}{v}, \quad \theta_{r,c}(v) = \frac{K_{MP,r}}{v}, \quad (29)$$

where K_{MPf} and $K_{MP,r}$ are weighting values. Eq. (29) reflects the difficulty to turn at a higher speed in formulating the degree of threat. Eq. (28) penalize the higher MP values.

Even when two obstacles are located in the identical parallax, the modification using Eq. (28) allows to reflect different levels of threat with respect to the actual moving direction of four vertices of the UGV. Figs. 11(a) and (b) show hazard distribution when MP-based and distance-based penalty functions are implemented, respectively. In the distance-based method, a finite number of spots are selected around the vehicle, and the least distance between these spots and obstacles is considered. On the other hand, the MP-based method used two largest MP information defined at front and rear face of the vehicle. When the vehicle turns right in Fig. 11, the real hazard level of two different obstacles at the identical distance from the vehicle can be different depending on the relative direction. The purely distance-based method in Fig. 11(b) cannot reflect this, but the MP in Fig. 11(a) can.

Furthermore, the MP contains information about the vehicle dimension. For instance, when an obstacle is located in front of a vehicle, it would trigger a larger parallax value to a wider vehicle than to a narrower one. Also, imagine two obstacles with the same parallax value $\pi - (\theta_{rl} + \theta_{rr})$, one of which is close to a long vehicle and the other to a short vehicle. The larger L value in Eq. (26) makes the correction term $|\beta_{rl} - \beta_{rr}|$ larger in absolute value, especially when $\dot{\varphi}$ is large. In Eq. (27), this leads to the larger modification to the original parallax value associated with the longer vehicle. And when its gradient term is used in MPC optimization, Eq. (28) contains rich information about state variables (β , φ , $\dot{\varphi}$ and the CG location) (Fig. 8).

5. Simulation results

To test the performance of the proposed approaches, numerical simulations are performed using the setting given in Table 1. Section 5.1 presents navigation in a rather simple environment. In Section 5.2, a more cluttered environment is considered as a trajectory space. In all plots of trajectories, the green dashed line

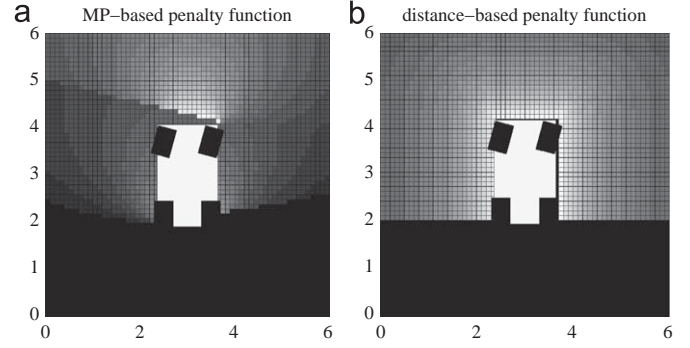


Fig. 8. Comparison of (a) MP-based and (b) distance-based penalty functions, corresponding to Eqs. (28) and (24), respectively. Lighter color denotes a more dangerous point.

Table 1
Numerical setting..

| | |
|------------------------------|----------------------------------------------------------|
| Horizon length (N) | 40 |
| Sampling time (Δt) | 0.05 s |
| Control command limit | $-60^\circ/\text{s} \leq u \leq 60^\circ/\text{s}$ |
| Steering angle limit | $-30^\circ \leq \delta_f \leq 30^\circ$ |
| Slip angle of a tire limit | $-6^\circ \leq \alpha_f, \alpha_r \leq 6^\circ$ |
| UGV specification | $m = 807 \text{ kg}$, $I_{zz} = 429.649 \text{ kg m}^2$ |
| UGV dimension | $1.290 \text{ m(W)} \times 2.150 \text{ m(L)}$ |
| Axle length | $L_r = 0.835 \text{ m}$, $L_f = 0.715 \text{ m}$ |
| Sensing range | 10 m with 180° field-of-view |

connecting the start and the goal locations represents the global reference trajectories planned offline.

5.1. Steering in a simple environment

In this scenario shown in Fig. 9, the space between the blocks is 10 m, which is comfortable for the UGV to navigate through. The distance- and MP-based methods in Eqs. (24) and (28) are implemented using the MPC approach and compared with the standard potential-field approach.

In the potential-field approach shown in cyan in Fig. 9, the controller finds the optimal steering rate that solves the following problem, using the similar cost-function terms as the MPC:

$$u_t^* = \underset{-u_{sat} \leq u \leq u_{sat}}{\operatorname{argmin}} (K_{ref} \tilde{\eta}_{t+\Delta t}^T \tilde{\eta}_{t+\Delta t}^T + PF_{goal,t+\Delta t} + PF_{obs,t+\Delta t}) \quad (30)$$

subject to

$$\begin{aligned} |\delta_{t+\Delta t}| &\leq \delta_{sat}, \\ |\alpha_{f,t+\Delta t}| &\leq \alpha_{sat}, \\ |\alpha_{r,t+\Delta t}| &\leq \alpha_{sat}, \end{aligned}$$

where K_{ref} is a weighting constant, and PF_{goal} and PF_{obs} are from Eqs. (23) and (24), respectively. Since three potential function terms are considered in Eq. (30) with three constraints which limit the possible control space, the gain tuning requires heavy efforts.

Red dashed curves in Fig. 9 are the results of the distance-based MPC method, where Eq. (24) is incorporated into Eq. (18). Blue solid curves in all plots represent the results of MP-based MPC method, where Eq. (28) is added to Eq. (18).

In all cases, successful collision-free trajectories are acquired, and u , δ_f , α_f , and α_r are kept within the given saturation values. In comparison between distance-based and MP-based MPC methods, clear difference in the generated trajectories does not exist. But

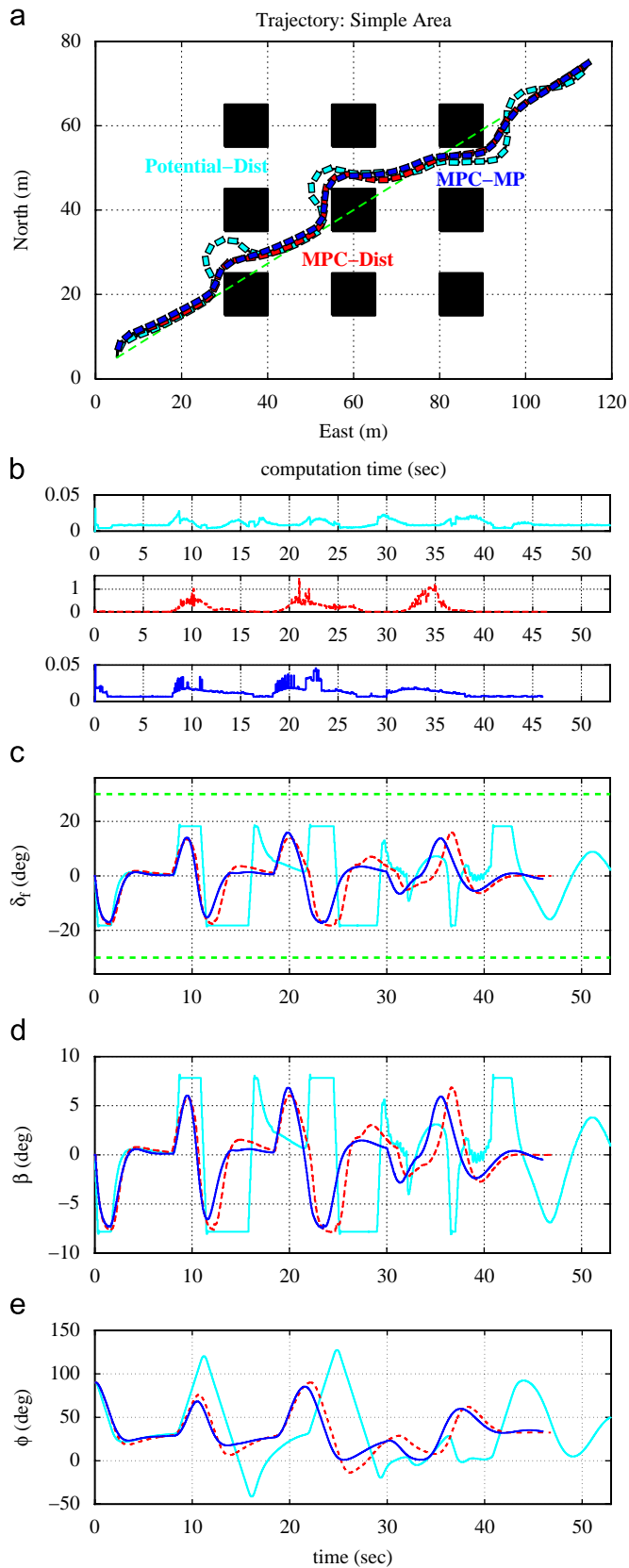


Fig. 9. Navigation using the distance-based potential function, distance-based MPC, and MP-based MPC methods: (a) resulting trajectories (cyan: potential, red: distance-MPC, blue: MP-MPC), (b) computation time (from top, potential, distance-MPC, and MP-MPC), (c) front steering angle, (d) slip angle of the CG, and (e) yaw angle. In all plots, cyan: potential field, red dashed: distance-based MPC, and blue solid: MP-MPC. Horizontal dashed lines are the saturation values.

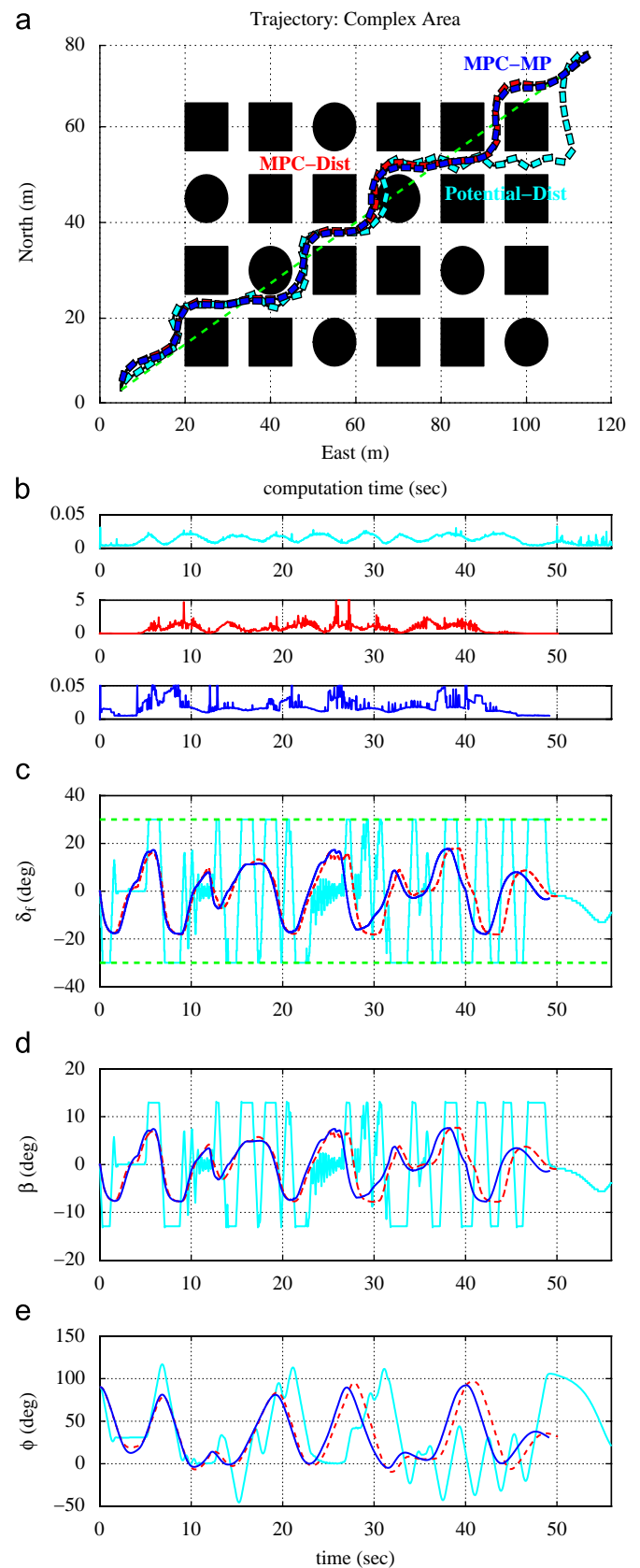


Fig. 10. Navigation in a more cluttered environment: (a) resulting trajectories, (b) computation time (top: distance-based method with potential field, middle: distance-based method with MPC, bottom: MP-based method with MPC), (c) front steering angle, (d) slip angle of the CG, (e) yaw angle. In all plots, cyan: potential field, red dashed curve: distance-based MPC, and blue solid curve: MP-based MPC. In (c), horizontal dashed lines denote the saturation values.

the MP-based method shows much increased computational efficiency (Fig. 9(b)).

In Fig. 9(b), the MP-based method in the MPC approach requires slightly longer computation time than the potential function approach. However, the obtained trajectory is better in the MPC approach in terms of the length and control effort (Fig. 9(a)). Furthermore, Figs. 9(c)–(e) show that the potential function approach uses the much larger control input resulting in larger changes in the state variables due to the lack of look-ahead philosophy.

In the MP-based MPC approach, the results are satisfactory in the view of computational efficiency and smoothness of the generated trajectory. With Eq. (28), the penalty cost in local path regeneration depends on relative location between the vehicle and obstacles, instant heading and its rate and the actual moving direction of vertices of the vehicle. Although distance information from obstacles is not included directly, MP values of Eq. (27) defined from the front and rear sides of the vehicle effectively express the degree of hazard and renders a safe path regeneration possible.

5.2. Steering in a complex environment

In this section, an area containing polygonal and circular blocks is assumed. The space between the blocks is 5 m which is rather tight for the UGV. Also note the short sensing range compared with the vehicle speed.

Fig. 10 shows the results of the three methods. The cyan curves are the results of potential function method, red dashed curves are those of distance-based MPC, and solid blue curves are the results of the MP-based MPC approach. For the potential-field approach even after extensive tuning, collision occurs with the circular obstacle centered at [70 m, 45 m].

In simpler obstacle avoidance problems, conventional approaches may suffice, which consider the UGV as a point and apply a distance-based potential function. However, to avoid the local minima issues and consider the global objective in the local trajectory generation, some form of looking ahead is desirable. Additional tangential fields or random perturbation to the potential function can take the vehicle out of local minima. However, it is very difficult to find a proper combination that can solve the collision and the constrained dynamics simultaneously. MPC provides a more convenient setting for local trajectory generation of the vehicles with complex dynamics.

The advantage of the MP-based method over the distance-based method becomes more clear in terms of the computation load reduced by the factor of more than 50 (see Fig. 10(b)). Whereas the computation time of the distance-MPC increased noticeably compared with the simpler environment of Fig. 9(b), the computation time of the MP-MPC increased only slightly.

Furthermore, in complex or tight environments, the dimension of the vehicle should be taken into account. Distance-based penalty functions consider two obstacles at the same distance from the vehicle with the same magnitude of threat, regardless of the size or moving direction of the vehicle. But the MP-based MPC method effectively reflects these, as discussed in Section 4.2. Also it is simple to implement, because the parallax angle about a detected obstacle can be estimated easily.

Fig. 11 illustrates an advantage of using the MP-based formulation (shown in (b)) over the distance-based approach (shown in (a)). In (a), the current value of the distance-based function term Eq. (24) would render the hazard levels of all the four obstacles the same. With the addition of the attraction to the goal position and tracking of the shortest path, the forward and right directions would be preferred almost equally. The optimization algorithm tends to spend much time in this type of situations. In (b), the MP-based formulation would consider the two

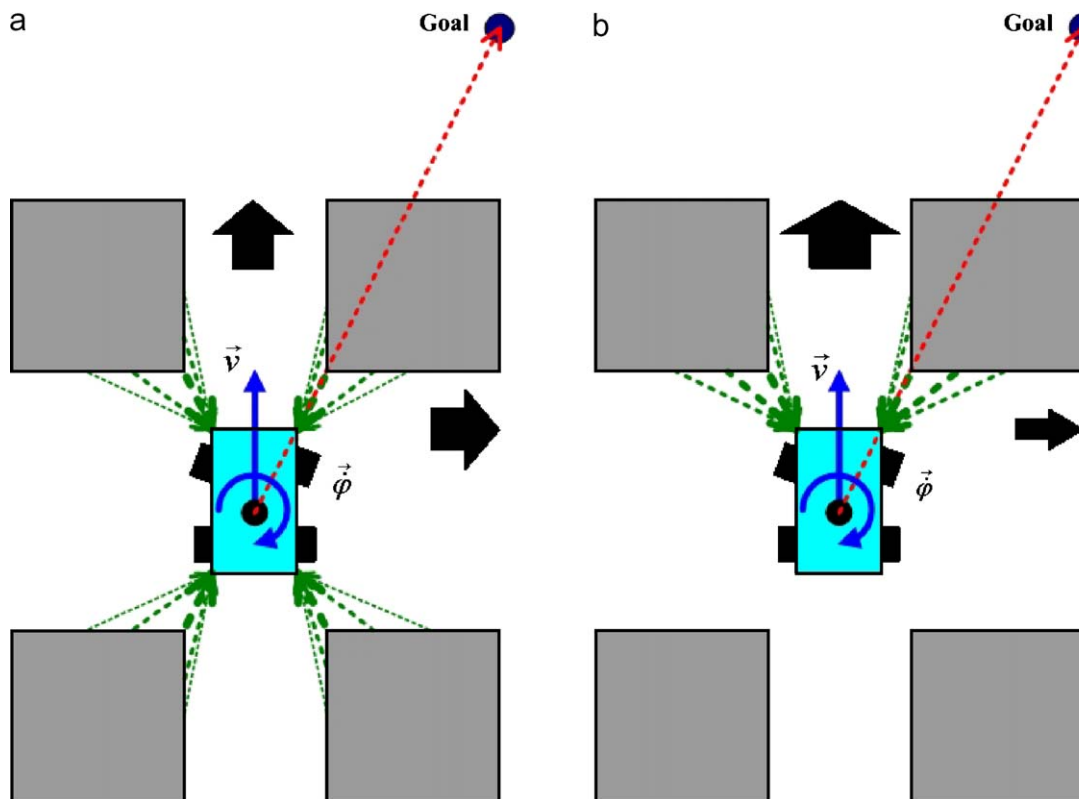


Fig. 11. (a) Distance-based method vs. (b) MP-based method. The MP formulation would select the front direction more effectively than the distance-based method.

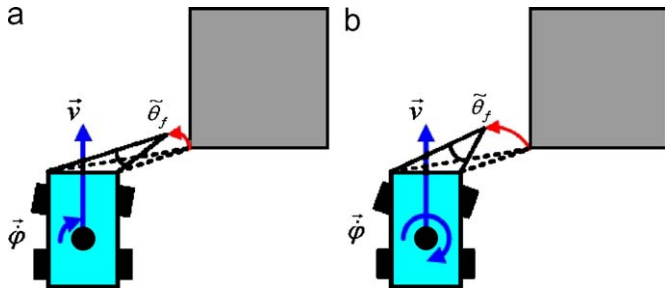


Fig. 12. MP for (a) a low rotational speed and (b) a high rotational speed. The front MP value will be larger when the heading rate is higher as in (b).

obstacles in front, and the consideration of the goal position and the shortest path would prefer the forward and the right directions. However, if the vehicle turns right, the MP value of the lower side of the upper-right obstacle would significantly increase. The optimization algorithm can select the forward motion within shorter search time.

Fig. 12 shows how the MP formulation takes the rotational speed into consideration. In (a) and (b), suppose that the relative location of the vehicle with respect to an obstacle is identical but the yaw rate is higher in (b). Then, although the original parallax values of the obstacle are identical, their modified values will be different. When $\dot{\phi}$ is larger, the computation of Eqs. (26) and (27) generates the modification term $\beta_{fr,k,t} - \beta_{fl,k,t}$ of larger magnitude.

5.3. Extensions of MP MPC

Fig. 13 shows the effect of vehicle speed in applying the MP-MPC methods. Fig. 13(a) shows the resultant trajectory with various velocity when the identical conditions and weighting parameters are used. For all the cases, safe trajectory generation was possible within the saturation limit. Although at higher speed the tire slip angle tends to increase and a larger rotational radius is required, difference was not noticeable in computation time. The same look-ahead horizon length corresponds to a wider prediction area if the vehicle moves faster, and it seems that the convergence of optimization does not depend strongly on the vehicle speed within the tested range.

In order to use MPC against moving obstacles, prediction about them is also required. In the simulation shown in Fig. 14, it is assumed that the heading angle and the location of the moving obstacles can be measured. Then, estimation of any other states such as speed (v), heading rate ($\dot{\phi}$) is needed. These variables are estimated from propagating information obtained from two consecutive time instants throughout the look-ahead horizon.

Fig. 14(b) shows a typical optimization process in terms of the cost-function reduction and the associated trajectories during the gradient descent. In the simulation shown in Fig. 14(c), two patrol vehicles are assumed as moving obstacles. The trajectories of these two patrols are generated with the MP-MPC method against stationary obstacles only. And the first patrol and the second one start to move toward west at 0 and 28 s, respectively. As can be seen in Fig. 14, the controller generates the safe trajectory without any collision. The vehicle turns left when it meets the first patrol coming from East, and turns right when it encounters the second one coming from the East. Fig. 14(a) shows the trajectory without the moving obstacles in the same environment for comparison. Computation time is up to about two times that of static environments, because predictive information about moving obstacles makes it more difficult to find a collision-free configuration. Because the most amount of computation time is taken to change the control sequence candidate when obstacles suddenly

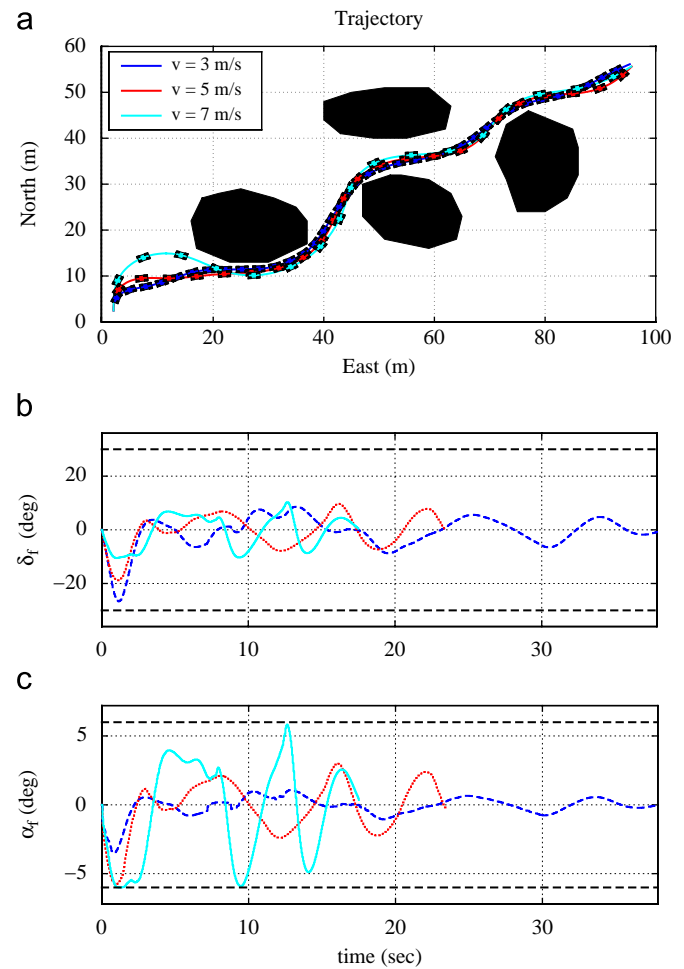


Fig. 13. MPC-MP navigation varying vehicle speed: (a) resulting trajectories, (b) front steering angle, and (c) front-tire slip angle. Blue dashed: 3 m/s, red dotted: 5 m/s, solid cyan: 7 m/s.

appear at the end of the horizon, the performance is still much better than the potential function method even if the optimization is forced to stop at the sampling time of 50 ms. Further improvements can be made by selecting better initial guesses of the control input sequence during the optimization or employing adaptive cost-function weighting matrices that penalize initial time steps more than the later steps. In addition, the optimization routine can be improved by multiple shooting with partially reduced sequential quadratic programming (SQP) (Schafer, Kuhl, Diehl, Schloder, & Bock, 2007).

6. Conclusion

A model-predictive approach for the collision-free trajectory generation of a UGV was presented. An optimal problem while avoiding obstacles was formulated in terms of cost minimization under constraints in the model-predictive control (MPC) framework. It is solved with nonlinear programming and the constraints were incorporated in the augmented Lagrangian. The local path planning step uses the information about the obstacles, detected by a limited-range sensor mounted on the vehicle. For implementation of the local path regenerator, the distance-based method and parallax-based method were proposed, with velocity weighting.

In order to address the obstacle avoidance in complex environments, the dimension of the UGV is taken into account. With an approach such as potential functions, rapid computation

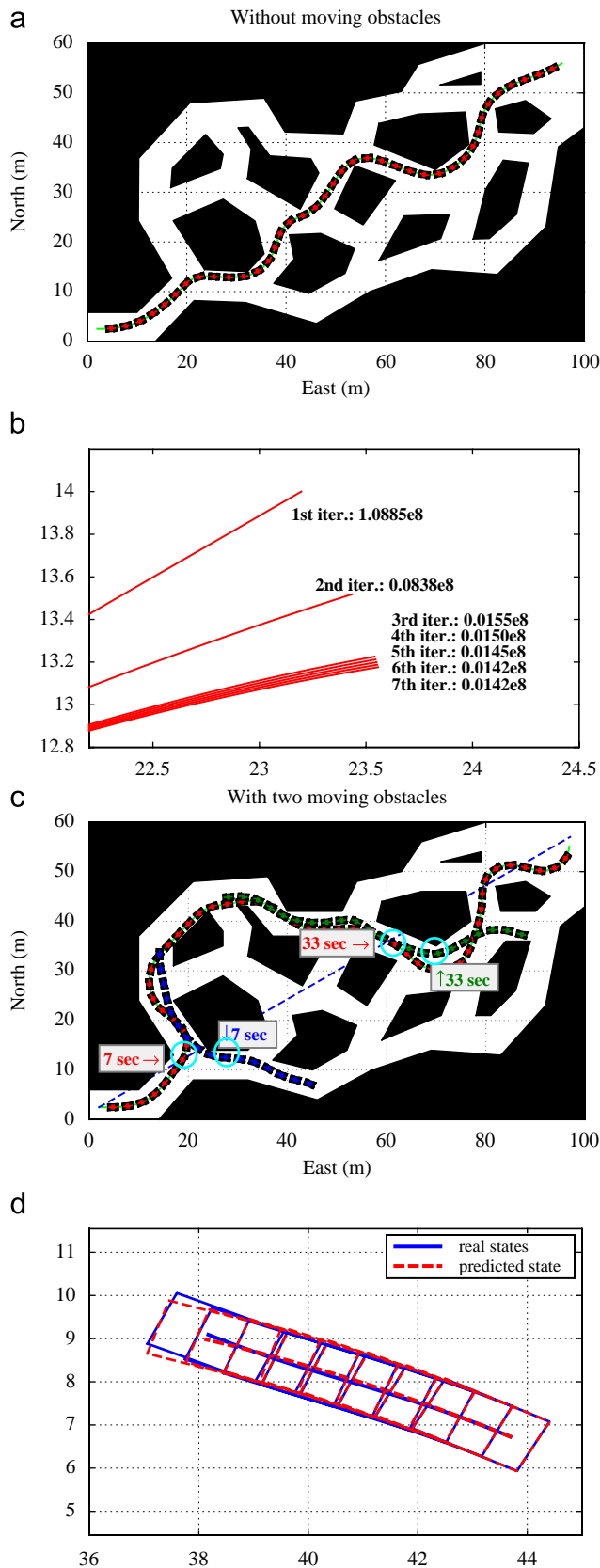


Fig. 14. MPC-MP navigation involving moving obstacles: (a) resulting trajectory without considering moving obstacles, (b) reduction of the cost function and convergence of the associated trajectory during the gradient descent optimization, (c) resulting trajectory (red) against two moving obstacles, and (d) prediction of the position of a moving obstacle.

for trajectory regeneration is possible, but collision with obstacles can happen due to local minima or difficulty to incorporate constrained dynamics over a longer time duration. The results with the MPC approach show that the application of distance-based method may require too much computation load especially in complex environments. On the other hand, the modified-parallax (MP)-based local path generator effectively utilizes rich information about UGV (including dimension, velocity and actual moving direction of the UGV), and it shows a high computational efficiency and better characteristics in path length and control efforts. Thus, the proposed MP-MPC approach can be a good candidate for real-time obstacle avoidance of autonomous ground vehicles in complex environments.

Acknowledgments

This work was partially supported by the Institute of Advanced Aerospace Technology at Seoul National University under the award D6-DU-LC-01 from the Agency for Defense Development and the Korea Science and Engineering Foundation (KOSEF) Grant ROA-2007-000-10017-0.

References

- Arras, K. O., Persson, J., Tomatis, N., & Siegwart, R. (2002). Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. *Proceeding of the IEEE international conference on robotics and automation*.
- Bakker, E., Nyborg, L., & Pacejka, H. B. (1987). Tyre modeling for use in vehicle dynamics studies. *SAE paper # 870421*.
- Borrelli, F., Falcone, P., Keviczky, T., Asgari, J., & Hrovat, D. (2005). Mpc-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, 3(2/3/4), 265–291.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., & Hrovat, D. (2006). A real-time model predictive control approach for autonomous active steering. In *IFAC workshop on NMPC for fast systems* (pp. 59–64), Grenoble, France.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., & Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3), 566–580.
- Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 23–33.
- Isermann, R. (2008). Mechatronic systems—innovative products with embedded control. *Control Engineering Practice*, 16(1), 14–29.
- Kim, H. J., & Shim, D. H. (2003). A flight control system for aerial robots: Algorithms and experiments. *Control Engineering Practice*, 11(12), 1389–1400.
- Kim, H. J., Shim, D. H., & Sastry, S. (2002). Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In *Proceedings of the American control conference* (pp. 3576–3581), Anchorage, AK.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789–814.
- Minguez, J., & Montano, L. (2000). Nearness diagram navigation(nd): A new real-time collision avoidance approach. In *Proceeding of the international conference on intelligent robots and system (IROS)*.
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 732–764.
- Rimon, R., & Koditschek, D. (1992). Exact robot navigations using artificial potential functions. *IEEE Transaction on Robotics and Automation*, 8(5), 501–518.
- Schafer, A., Kuhl, P., Diehl, M., Schloder, J., & Bock, H. G. (2007). Fast reduced multiple shooting method for nonlinear model predictive control. *Chemical Engineering and Processing*, 46, 1200–1214.
- Simmon, R. G. (1996). The curvature velocity method for local obstacle avoidance. In *Proceeding of the IEEE international conference on robotics and automation* (pp. 2275–2282).
- Spensko, M., Iagnemma, K., & Dubowsky, S. (2004). High speed hazard avoidance for mobile robots in rough terrain. In *Proceeding of the 2004 SPIE conference on unmanned ground vehicle technology*.
- Sutton, G. J., & Bitmead, R. R. (2000). Computational implementation of nmmpc to nonlinear submarine. In F. Allgöwer, & A. Zheng (Eds.), *Nonlinear model predictive control*, Vol. 26 (pp. 461–471). Basel: Birkhäuser.
- Ulrich, I., & Borenstein, J. (1998). Vfh+: Reliable obstacle avoidance for fast mobile robots. In *Proceeding of the IEEE international conference on robotics and automation* (pp. 1572–1577).
- Ulrich, I., & Borenstein, J. (2000). Vfh*: Local obstacle avoidance with look-ahead verification. In *Proceeding of the IEEE international conference on robotics and automation* (pp. 2505–2511).
- Wollherr, D. (2002). *Robust steering control for swerving manoeuvres of a motor vehicle*. Ph.D. Thesis. Automatic Control Lab, ETH, Zurich.