

Communication in distributed model predictive collision avoidance

Yongsoon Yoon, H. Jin Kim, J. H. Shin
 School of Mechanical & Aerospace Eng.
 Seoul National University
 Seoul, 151-742, KOREA
 Tel: +82-2-880-7392
 Email: {ilsoon6, hjinkim, jh0524}@snu.ac.kr

Tokson Choe and Yongwoon Park
 Agency for Defense Development
 Jochiwongil 461, Yuseong
 Daejeon, KOREA
 Tel: +82-42-822-4271,
 tschoe@add.re.kr, woon5901@hanafos.com

Abstract—This paper presents a model predictive approach for collision avoidance of car-like robots. An optimal problem is formulated in terms of cost minimization under constraints. Information on each robot can be incorporated online in the nonlinear model predictive framework and kinematic constraints are treated by Karush-Kuhn-Tucker(KKT) condition. For distributed collision avoidance of multiple robots with two levels of a communication network, performances are compared. In comparison with different types of communication, how much information the robots share can cause difference in the performance. More successful collision avoidance was possible when the robots share enough amount of information.

I. INTRODUCTION

Many research on local collision avoidance, which use purely reactive methods based on sensor input, have been performed [1], [2], [3], [4]. Some of them take into account the dynamics and kinematics constraints [5]. These approaches are computationally efficient, but the robot can get stuck in local minima, sometimes the discretization of the world is required, or the full dynamics cannot be incorporated. Recently, predictive active steering control for autonomous vehicle systems was studied [7], [8] with a nonlinear tire model [6]. In these work, the autonomous vehicle was directed to follow the given reference which is assumed to be collision-free and achievable. But sometimes we have to consider the dimension of the robot for a tight collision avoidance. And for multiple robots, since a car-like robot model is strongly nonholonomic and the environment changes dynamically, collision avoidance is not easy. It may fail due to a lack of information shared among the robots.

In this paper, we present a model predictive method for active steering control of a mobile robot based on successive on-line optimization. In order to use this approach as a local collision avoidance planner for the robot, we use a bicycle model to predict the future evolution of the system. This paper is focused on the collision avoidance with considerations of dimension of the robot and different levels of communication capability. If a robot runs into dynamic environment, controller predicts a future path and solves an optimization problem to replan collision-free trajectories.

Nonlinear model predictive control has been used to generate safe trajectories for unmanned aerial vehicles [9], but without including the saturation limits as constraints. In the current paper, safety limits on the state variables, magnitude of the inputs, and limited steering rates are explicitly incorporated using the Karush-Kuhn-Tucker (KKT) condition.

II. MODEL

We use a bicycle model which has the yaw degrees of freedom in combination with a nonlinear tire model to describe the dynamics of the robot.

We will use the following notation throughout this paper:

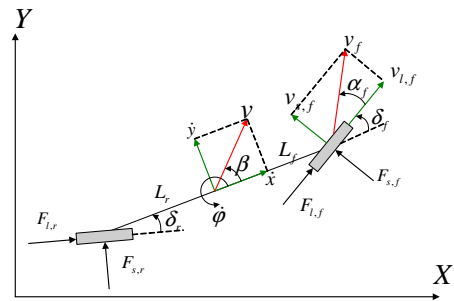


Fig. 1. A schematic diagram of the bicycle robot model.

- (.)f front wheel
- (.)r rear wheel
- (.)x x-axis of the robot's body coordinate
- (.)y y-axis of the robot's body coordinate
- (.)s side direction
- (.)l longitudinal direction
- (.)k time step
- (.)ref reference tracking signals

This nomenclature refers to the robot model depicted in Fig. 1. \dot{x} , \dot{y} and v are defined with respect to the body coordinate frame which its origin coincides the center of the gravity of the robot. α and β denote the slip angle of a tire and the side slip of the robot, respectively, defined as the angle between the velocity vector and their longitudinal axes. δ is the steering angle of tires and ψ denotes the heading angle of

robot with respect to the absolute inertial frame. In this paper, we consider front-wheel steering robots.

A. Nonlinear robot model

Eq. (1) describes dynamics of the robot of mass m and moment of inertia I_{zz} .

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

Setting up the equilibrium of forces and momentum yields Eq. (2), where $F_{sf}, F_{sr}, F_{lf}, F_{lr}$ are lateral and longitudinal tire forces exerted on each tire.

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

Lateral and longitudinal tire forces for each tire (to be detailed in Sec. II-B) are given by

$$\begin{aligned} F_l &= f_l(\alpha, s, \mu, F_z) \\ F_s &= f_s(\alpha, s, \mu, F_z), \end{aligned} \quad (3)$$

where s is the slip ratio and α is the function of $\beta, \dot{\psi}, v$ and δ . (explained in [6]) The parameter μ in Eq. (3) represents the road friction coefficient and F_z is the total vertical load of the robot. The robot's equations of motion in the inertial frame are

$$\begin{aligned} \dot{X} &= v \cos \beta \cos \psi - v \sin \beta \sin \psi, \\ \dot{Y} &= v \cos \beta \sin \psi - v \sin \beta \cos \psi. \end{aligned} \quad (4)$$

Using the equations Eq. (1) - Eq. (4), the nonlinear robot dynamics can be described by the following differential equation with the slip ratio s and the friction coefficient μ :

$$\begin{aligned} \dot{\xi} &= f_{s,\mu}(\xi, u), \\ \eta &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xi, \end{aligned} \quad (5)$$

where the state and input vectors are $\xi = [\beta \ v \ \psi \ \dot{\psi} \ X \ Y]^T$ and $u = \delta_f$ respectively, and the output vector is $\eta = [X \ Y]^T$.

B. Tire model

To compute the forces exerted on the tires, we need a tire model. The tire model cannot be expressed by a single closed formula, since they are nonlinear and vary strongly with the state of the tire. The tire forces are functions of the axle load F_z , the slip angle of the tire α , the slip ratio s , friction coefficient μ as mentioned in Eq. (3). There are some models available for computational simulations. We used Pacejka's MAGIC formula [6], which is a semi-empirical model that includes the consideration of the interaction between the longitudinal and lateral forces. Fig. 2(a) compares the longitudinal

tire force with respect to the slip ratio and Fig. 2(b) shows the lateral tire force with respect to the slip angle with varying F_z .

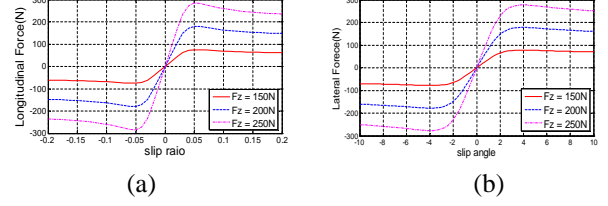


Fig. 2. (a) Longitudinal tire force with different slip ratio. (b) Lateral tire force with different slip angle, respectively.

III. MODEL PREDICTIVE CONTROL PROBLEM

A. Algorithm

Since model predictive algorithm is solved in a discrete time domain, the system dynamics Eq. (5) are discretized with the Euler method as the following equation [10].

$$\begin{aligned} \xi(k+1) &= f_s^{dt}(\xi(k), u(k)), \\ \eta(k+1) &= h(\eta(k)). \end{aligned} \quad (6)$$

Eq. (7) is the cost function, with the additional constraint terms on physical limits and penalty function for collision-free planning.

$$\begin{aligned} J &= \phi(\tilde{\eta}_N) + \sum_{k=0}^{N-1} (L(\tilde{\eta}_k, u_k, \Delta u_k) \\ &+ \sum_i \mu_i S_k^i l_k^i + P_k^{obs} + P_k^{goal}) \\ &i = \Delta u, u, v \end{aligned} \quad (7)$$

$$\phi(\tilde{\eta}_N) = \frac{1}{2} \tilde{\eta}_N^T P_0 \tilde{\eta}_N \quad (8)$$

$$L(\tilde{\eta}_k, u_k, \Delta u_k) = \frac{1}{2} \tilde{\eta}_k^T Q \tilde{\eta}_k + \frac{1}{2} \Delta u_k^T R \Delta u_k + \frac{1}{2} u_k^T T u_k \quad (9)$$

$$\sum_i \mu_i S_k^i l_k^i = \mu_u S_k^u l_k^u + \mu_{\Delta u} S_k^{\Delta u} l_k^{\Delta u} + \mu_v S_k^v l_k^v \quad (10)$$

where N is the horizon-length. $\eta_k, \tilde{\eta}_k$ and Δu_k are formulated as $\eta_k = C\xi_k, \tilde{\eta}_k = \eta_{k,ref} - \eta_k, \Delta u_k = u_k - u_{k-1}$ and μ_i ($i = u, \Delta u, v$) are KKT variables. $P_0, Q, R, T, K_{obs}, K_{goal}$ are weighting parameters. The fourth and fifth terms of Eq. (7) are potential-like functions to guide the robot into the destination avoiding obstacles as followings.

$$P_k^{obs} = \frac{K_{obs} \cdot l_k^{obs}}{\min \|q_{k,j} - q_{obs}\| + \epsilon} \quad (11)$$

$$P_k^{goal} = K_{goal} \cdot \|q_k - q_{goal}\|^2 \quad (12)$$

$q_{k,j}$ and q_{obs} are location of the j th critical point around the robot and location of obstacles detected by sonar-like on-board sensors, respectively at time k . This is for consideration of

dimension of the robot. l_k^{obs} is to consider the range of the sensor and it is defined as the following:

$$S_k^{obs} = d_{max} - \min \|q_{k,j} - q_{obs}\|, \quad (13)$$

$$l_k^{obs} = \begin{cases} 0 & , \text{ if } S_k^{obs} < 0, \\ 1 & , \text{ else,} \end{cases}$$

ϵ is a constant for keeping the denominator of Eq. (11) non-zero. And q_k, q_{goal} are XY coordinate of the center of gravity of the robot and location of the destination in the inertial frame. Eq. (10) contains penalty on the larger values than the fixed saturation values, defined in [10]. These terms are formulated as constraints in the nonlinear model predictive framework. KKT variables mean the weights of the penalty about constraint violation of the state variables and steering input. The online optimization was solved with the augmented lagrangian approach using gradient descent method explained in [9], [10].

B. Dimension of the Robot

In order to apply Eq. (11), two points corresponding to the shortest distance between the robot and obstacle should be known. We assume that this information can be obtained by sensors mounted around the robot. In Fig. 3, q_k and

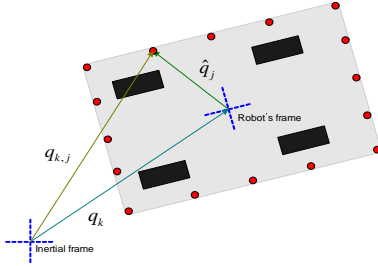


Fig. 3. Red-circles: critical points around the robot.

$q_{k,j}$ respectively are the center of gravity of the robot and position of the j_{th} point around the robot in the inertial frame, respectively. And \hat{q}_j is coordinate of the j_{th} point in the robot's frame.

IV. SIMULATION RESULTS

In order to validate the effectiveness of our approach on the nonlinear car-like robot models, we performed numerical simulations. The following parameters were used for two simulations presented here:

- horizon length : 15
- sample time : $\Delta t = 0.05$ sec (i.e. 20 Hz)
- steering angles : $-30^\circ \leq \delta_f \leq 30^\circ$
- changes of steering angles : $-3^\circ \leq \Delta\delta_f \leq 3^\circ$
- minimum velocity : $v_{min} \leq v, v_{min} = 2$ m/s
- maximum sensor range : $d_{max} = 15$ m

A. Two robots : one-step broadcast

In this simulation, two robots are employed for collision avoidance. At time zero, two robots were facing each other with initial states of $[0 \ 4 \ 0 \ 0 \ 10 \ 10]^T$ and $[0 \ 4 \ \pi \ 0 \ 40 \ 10]^T$, respectively. The robots are commanded to move along the reference trajectories which is dashed green line shown in Fig. 5(a). The robots are of an identical rectangular shape and each robot thinks the other robot as a moving rectangular obstacle.

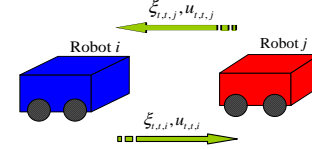


Fig. 4. Flow of communication network sharing 1-step information between the robots.

We assumed that each robot knows the first input command and previous position of the other robot by the communication network between the robots(See Fig. 4). The robot i thinks the robot j will navigate with the same control input from the known previous location for the next 15 steps as the following:

$$\xi_{t+k+1,t,j} = f(\xi_{t+k,t,j}, u_{t,t,j}), \quad k = 0 \dots N - 1 \quad (14)$$

$\xi_{t+k,t,j}$ denotes the state vector of the j_{th} robot predicted at time $t+k$ obtained by starting from the state $\xi_{t+k} = \xi(t)$.

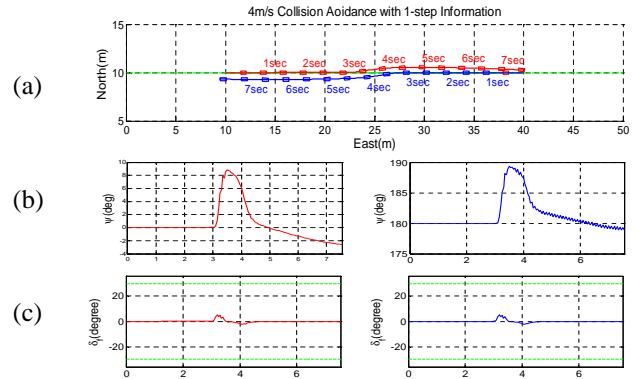


Fig. 5. (a), The trajectory of the robots with initial speeds (4 m/s). (b), heading angles(ψ) and (c), steering command(δ_f) of the robot (solid lines) and saturation value (dashed green line) with one step information sharing. The red and blue curve mean the red and blue robot. And x-axis of (b) and (c) is time(sec).

Fig. 5(a), (b), (c) are results for the two robots moving with the initial speed of 4 m/s. The left(red) and the right(blue) figures in Fig. 5(b-c) correspond to the red robot starting from the left side and the blue robot starting from the right side, respectively. As can be seen in Fig. 5(a), the performances in terms of collision-free navigation are satisfactory. Both robots adjust the steering commands to avoid each other about 5 deg in counterclockwise and after that instance, their steering commands become negative (i.e. clockwise) to rejoin the

reference (See Fig. 5(c)). But the robots approach too close because of imperfect information. If initial speed increases, collision avoidance is not easy. To compare computation time, the ratio between CPU time and real time r is defined and $r_{4\text{ m/s}}$ was 4.59.

B. Two robots : sharing full information

In this simulation, we assume that the robots share the information throughout the look-ahead horizon. (See Fig. 6).

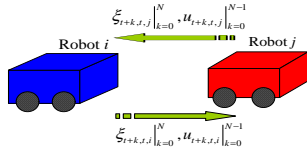


Fig. 6. Flow of communication network sharing all information between the robots.

That is, they know all the predicted state variables and steering commands over the horizon length of the other robot exactly as the following:

$$\xi_{t+k+1,t,j} = f(\xi_{t+k,t,j}, u_{t+k,t,j}), k = 0 \cdots N - 1 \quad (15)$$

The initial state and weighting parameters are identical to the simulation in Sec. IV-A.

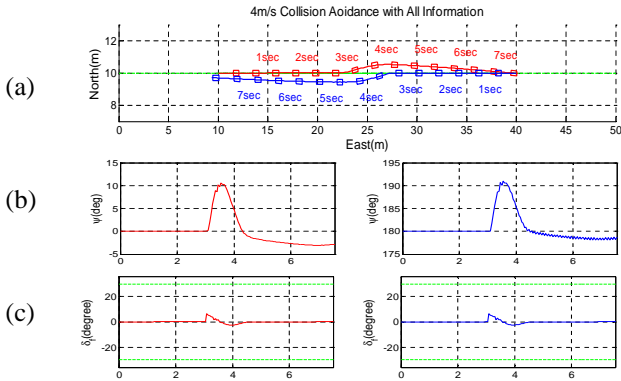


Fig. 7. (a), The trajectory of the robots with initial speed (4 m/s). (b), heading angles(ψ) and (c), steering commands(δ_f) of the robot (solid lines) and saturation value (dashed green line) with all information sharing. The red and blue curve mean the red and blue robot. And x-axis of (b) and (c) is time(sec).

The robots avoided each other successfully (See Fig. 7(a)). The both robots adjust the steering command to avoid each other about 7 deg in counterclockwise and after that instance, their steering commands become negative (i.e. clockwise) to rejoin the given reference for a longer time than those in 1-step sharing (See Fig. 5(c) vs Fig. 7(c)). As a result, the robots rejoin the reference trajectory more rapidly (See Fig. 5(a) vs Fig. 7(a)). In Sec. III-A, in order to decrease computation time

u_k is initialized at the beginning of the optimization at each step with the u_k of the previous optimization [9]. Because of this, sharing all-step information over the horizon corresponds to sharing their future plan. $r_{4\text{ m/s}}$ (defined in IV-A) is 3.42. These results mean that computation time can be reduced with more information. Even if initial speed increases collision can be avoided according to further simulation. In other words, control can be easier with more information about the other.

V. CONCLUSION

A model predictive approach for trajectory generation of a car-like robot was presented. An optimal problem while avoiding collision was formulated in terms of cost minimization under constraints. We solved this with nonlinear programming and the constraints were incorporated as a penalty function with KKT variables. We tested two scenarios. In the first scenario, the robots share only the one-step information about each other. In the second scenario, the robots share all-step information throughout the look-ahead horizon. In comparison of performances with different levels of communication, the case of sharing all-step information is prior to the case of sharing only 1-step information in the view of avoiding collision and computation time.

ACKNOWLEDGMENT

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD)(KRF-2005- 204-D00002), and by the Agency for Defense Development under the award 0498-20060021, administered via the Institute of Advanced Aerospace Technology at Seoul National University, Korea.

REFERENCES

- [1] E.Rimon,D.Kodischek,“Exact robot navigations using artificial potential functions,” *IEEE Transaction on Ronoics and Automation* 8(5)501-518, 1992.
- [2] D.Fox,W.Burgard and S.Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*4(1)23-33, 1997.
- [3] R.G.Simmon,“The curvature velocity method for local obstacle avoidance,” in *Proceeding of the IEEE International Conference on Robotics and Automation*,vol 4, pp.2275-2282, 1996.
- [4] J.Minguez and L.Montano,“Nearness diagram navigation(ND): a new real-time collision avoidance approach,”in *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and System(IROS)*,2000.
- [5] I.Ulrich and J.Borenstein,VFH+“reliable obstacle avoidance for fast mobile fonots,”in *Proceeding of the IEEE International Conference on Robotics and Automation*,pp.1572-1577, 2000.
- [6] E.Bakker,L.Nyborg and H.B.Pacejka, “Tyre modeling for use in vehicle dynamics studies,”*SAE paper* # 870421, 1987.
- [7] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari and D. Hrovat, “ MPC-based approach to active steering for autonomous vehicle systems,”*Int. J. Vehicle Autonomous Systems*, 3(2/3/4), 265-291, 2005.
- [8] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng and D. Hrovat, “ Predictive active steering control for autonomous vehicle systems,” *Int. J. IEEE Transactions on Control Systems Technology* , 2007.
- [9] H. J. Kim, D. H. Shim and S. Sastry, “Nonlinear model predictive tracking control for rotorcraft-base unmanned aerial vehicles,” *Proceedings of the American Control Conference*, Anchorage, AK, pp 3576-3581, May 2002.
- [10] J.H. Shin and H. J. Kim, “The decentralized Nonlinear Model Predictive Formation Flight,” *17th IFAC Symposium on Automatic Control in Aerospace*, Toulouse, France, June 2007.