2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Sampling Based Predictive Control with Frequency Domain Input Sampling for Smooth Collision Avoidance*

Hiroyuki Okuda[1], Yingqi Liang[1], Tatsuya Suzuki[1]

Abstract— **This paper presents the smooth path generation method based on the sampling based model predictive control. Non-linear model predictive control framework is applied together with considering the car dynamics and non-linear constraints to achieve collision avoidance to obstacles. Since the control input must be optimized with non-linear constraints every control step in real-time, the problem is solved by sampling based approach. One of big issues on the sampling based approach is efficient sampling of smooth control input because some applications require the smoothness of control input. In proposed method, series of input are sampled in frequency domain directly in order to generate the smooth input series instead of applying the digital filtering to sampled input from specified random process. The proposed method samples the magnitudes of frequency components directly and those samples are transformed to time domain signals by Inverse Discrete Cosine Transform (IDCT.) Proposed method is implemented in simulation experiment with car dynamics simulator for collision avoidance task, and its validity is tested from the viewpoint of path tracking performance and real-time computation capability in several collision avoidance scenarios.**

I. INTRODUCTION

Demand for autonomous driving is growing and spreading. Needless to say, the environmental recognition of autonomous driving is necessary part. At the same time, vehicle control is also becoming essential technique for autonomous driving in order to realize the reliable motion control.

Motion control of autonomous vehicle usually consists of path planning part and control part. Path planner generates the reference path considering the objectives, such as the lane keeping, lane changing and so on, and the constraints, such as collision avoidance, input constraints and so on. There is a huge number of researches about path planning in automotive and robotic area [2]–[5]. Sampling-based approach such as rapidly-exploring random tree, lattice-based motion planning and so on, and optimization based approach with expressing the path as the parametric curve such as the spline, Bezier Curve and so on, are often used as path planner.

Vehicle control determine the operation(input) of targeting car to follow the generated reference path in path planner. There are also many researches in this area, and many control methods are proposed for path tracking problem for cars such as simple PID controller, model predictive controller and so on [6]–[10].

One of the concerns of this conventional approach is that targeting car sometimes can not follow the planned path exactly because of its dynamic property in path planning phase. However, there is some redundancy if the car dynamics is considered in both planning stage and path following stage. It becomes natural to consider the path planning and vehicle control problems in the unified framework. This framework is called 'simultaneous motion planning and control'(SMPC) problem these days [11], [12].

Model predictive control(MPC) [8]–[10] is the promising framework which can deal with the SMPC problem in real time. Especially, a sample based MPC solution, witch is called as 'Randomized MPC' is focused as the practical method which can obtain the sub-optimal input for the plant in real-time with considering the non-linear state equations and non-linear constraints. However, since it is quite difficult to expect the parameters of prior probability to find the optimal solutions, normal distribution or uniform distribution are tend to be applied as the sampling distribution. This results the un-smooth nature of sampled input series and it requires the de-noising filter to generate smooth operation in the autonomous driving application.

This paper utilize sample-based approach for the non-linear MPC problem that can generate smooth trajectory to improve ride comfort. Instead of applying the de-noising filter to the sampled input series, input series are sampled directly in frequency domain within restricted band-width. Sampled input series in frequency domain is transformed to the time-series input by the inverse discrete cosine transform(IDCT.) Thanks to fast computation nature of IDCT, smooth input are obtained without loosing real time computation capability. The validity of proposed method is checked in simulator experiments using car dynamics simulator. In addition, the computation time is also checked if proposed method can achieve real-time control toward the practical application.

II. OVERVIEW OF PROPOSED SYSTEM

A. Problems settings

A collision avoidance task shown in Fig.1 is approached in this paper. Ego-car drives the center of a narrow straight street with 6[m] width from the origin, i.e. $(p_x(t = 0), p_y(t = 0)) = (0, 0)$ where t is time index. N_O of cars are parked on left and right side of the street. The position of ith parked car is $(o_x^i, o_y^i), i \in \{1, 2, \cdots, N_O\}$. Ego-car have to steer right and left to avoid the collision to parked cars. Ego-car is assumed to be fully automated and to accept the tire angle command and the speed command as control input.

*This work was not supported by any organization

[1]Hiroyuki Okuda, Yingqi Liang and Tatsuya Suzuki are with Nagoya University, Furo-cho, Chikusa, Nagoya, Aichi, Japan h_okuda@nuem.nagoya-u.ac.jp

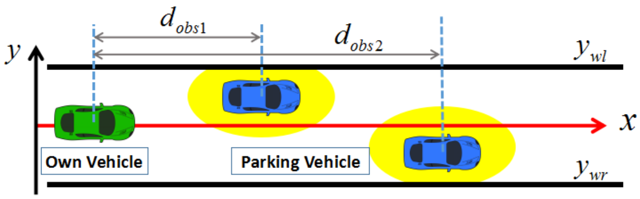

Fig. 1. Targeting environment

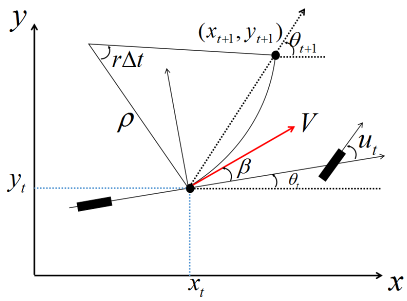

Fig. 2. Steady state circular model

B. Outline of proposed system

This paper approaches the collision avoidance problem in previous section by the simultaneous motion planning and control(SMPC) based on MPC framework with constraint. This means that the control input is directly computed in one unified framework with considering the car dynamics, physical/safety constraints, and driving objective at the same time instead of making a reference path to avoid the obstacles in advance.

An equivalent bicycle model(sectionII-C) are considered as the car dynamics model. In addition, input constraint and safety constraints(section II-D) to keep a margin to parked car are also considered as the constraints in the framework.

To realize this control framework in real-time, sample-based approach is used to solve the optimization problem at each control step in non-linear MPC framework. This control scheme is well known as 'Randomized Model Predictive Control(RMPC)' scheme and is attracting great attention recently [13]–[18].

In autonomous driving application, the most important factor for driving comfort of the passenger is smoothness of the steering operation. While some previous literatures utilized the random walk process with filtering, the input series are sampled in frequency domain with bandwidth limitation and transformed to time-domain data by Inverse Discrete Cosine Transform(IDCT).

C. State equation for car dynamics

As the vehicle dynamics model, a steady state circular vehicle behavior is assumed in each time instance [1]. This can be regarded as the simplified equivalent bicycle model with assuming that the time derivatives of both front tire slip angle and rotation speed are zero. This simplification is used for fast computation to achieve real-time control while this may lead slight deterioration of the accuracy compared to dynamic bicycle model. The equation of vehicle motion can be expressed as follows:

TABLE I

DEFINITION OF PARAMETERS AND VARIABLES

$p_x(t), p_y(t)$	ego-car position
$\beta(t)$	body slip angle
$r(t)$	yaw rate
$u(t)$	front tire angle
K_f	cornering stiffness of front tire
K_r	cornering stiffness of rear tire
l_f	length to front axle from CoG
l_r	length to rear axle from CoG
m	Vehicle mass

$$2K_f u_t = 2(K_f + K_r)\beta_t + \left\{ mV_t + \frac{2}{V_t}(l_f K_f - l_r K_r) \right\} r_t, \tag{1}$$

$$2l_f K_f u_t = 2(l_f K_f - l_r K_r)\beta_t + \frac{2(l_f^2 K_f + l_r^2 K_r)}{V_t} r_t. \tag{2}$$

where β_t is body slip angle and $r(t)$ is yaw rate at the time of t. Control input u_t is front tire angle, $V(t)$(assumed to be constant $V(t) = V$ in this paper) is the driving speed of ego-car. Definitions of all parameters and variables are listed in Table I. From Eq.2, $\beta(t)$ and $r(t)$ can be derived as:

$$\beta_t = \left(\frac{1 - \frac{m}{2l}\frac{l_f}{l_r K_r} V_t^2}{1 - \frac{m}{2l^2}\frac{l_f K_f - l_r K_r}{K_f K_r} V_t^2} \right) \frac{l_r}{l} u_t, \tag{3}$$

$$r_t = \left(\frac{2l^2 K_f K_r}{2l^2 K_f K_r - m(l_f K_f - l_r K_r) V_t^2} \right) \frac{V_t}{l} u_t, \tag{4}$$

respectively. When the car is under the steady state circular behavior with the constant speed $V(t)$ and constant yaw rate r_t, the turning radiusρ is computed as:

$$\rho_t = \frac{V_t}{r_t} = \left(1 - \frac{m}{2l^2}\frac{l_f K_f - l_r K_r}{K_f K_r} V_t^2 \right) \frac{l}{u_t}. \tag{5}$$

As you can see in equations 2, 4 and 5, $\beta(t)$, $r(t)$ and $\rho(t)$ are linear to control input $u(t)$ by assuming constant $V(t) = V$.

In addition, ego-car position should be also included as the state variables of the model for consideration of safety constraints. So the state of ego-car is described as $x(t) = [p_x(t), p_y(t), \theta(t)]^T$ and state equations for positions($x(t)$ and $y(t)$) and yaw angle($\theta(t)$) of cars can be written by considering car geometry as follows:

$$\begin{aligned} p_x(t+1) =& p_x(t) + 2\rho(u_t)\sin(\frac{1}{2}r(u_t)\Delta t) \\ & \times \cos(\frac{1}{2}r(u_t)\Delta t + \beta(u_t) + \theta_t), \end{aligned} \tag{6}$$

$$\begin{aligned} p_y(t+1) =& p_y(t) + 2\rho(u_t)\sin(\frac{1}{2}r(u_t)\Delta t) \\ & \times \sin(\frac{1}{2}r(u_t)\Delta t + \beta(u_t) + \theta_t), \end{aligned} \tag{7}$$

$$\theta_{t+1} = \theta_t + r(u_t)\Delta t, \tag{8}$$

where Δt is control interval, respectively.

D. Input and safety constraints

To ensure the feasibility of the input and safety, input constraint and the safety constraint not to enter prohibited area are considered as the hard constraints in proposed method.

The control input u_t is simply constrained within the range of $\|u_t\| < 0.1745$ (≈ 10[degree]) to avoid the urgent steering operation. On the other hand, an ellipse is considered around each obstacle cars as prohibited area to take safety margin to them. The safety margin constraints for the ith parked car can be written as the following inequality:

$$\left(\frac{x_t - x_{obs}^i}{r_a^i}\right)^2 + \left(\frac{y_t - y_{obs}^i}{r_b^i}\right)^2 > 1 \quad \forall i \in \{1, 2, \cdots, N_O\}, \tag{9}$$

where N_O is the number of parked cars in the driving environment and x_{obs}^i and y_{obs}^i are the position of the ith parked car. r_a^i and r_b^i are the length of major axis and minor axis of the ellipse set for the ith parked car, respectively.

E. Cost function and reference state

The main goal of ego-car is to drive the center of street without any collision with obstacles. Here the basic cost function set for MPC $J_1(\boldsymbol{u}_t)$ for given input series $\boldsymbol{u}_t = [u(0|t), u(1|t), \cdots, u(N-1|t)]$ to be optimized at the time of t is set as follows:

$$J_1(\boldsymbol{u}_t) = \sum_{k=1}^{N-1} \left(\phi^T(k|t)Q\phi(k|t) + \Delta u(k|t)^T R \Delta u(k|t)\right), + \phi^T(N|t)Q_f\phi(N|t)$$

$$\text{where} \quad \phi(k|t) = x(k|t) - x^{ref}, \tag{10}$$

$$\Delta u(0|t) = 0, \ \Delta u(k|t) = |u(k|t) - u(k-1|t)|_1 \quad \forall k \in \{1, 2, \cdots, N-1\}, \tag{11}$$

$$x^{ref} = [0, 0, 0]^T, \tag{12}$$

where N is a number of steps of prediction horizon in MPC, ϕ the state error, Δu the time difference of control input and x^{ref} is the state reference, respectively. Q and R are the weight parameters to balance the effort and the control performance and Q_f is the penalty to residue at the last step in prediction horizon. From the view point of SMPC, path planning for state reference, i.e. reference path, to avoid obstacles is skipped, but a simple goal, $p_y^{ref} = 0$ and $\theta^{ref} = 0$, are given as the reference state instead. Note that the car position along the street, p_x, is not penalized by setting first column and first row of Q to zero while the reference state for p_x is set to zero. In the cost function, the time difference of the control input is penalized in second term to improve the smoothness of steering operation.

F. Modified cost function for smooth collision avoidance

The cost function shown in previous section may work well. However, the collision avoidance constraints and the penalty to the deviation from street center sometime conflict to each other. In addition, ego-car should not drive the most closest point to the parked car for improving the system's reliability even the minimum safety margin is assured by safety constraints. In this paper, the potential field to enable the smooth collision avoidance is introduced to the cost function, and the weight balance between the term to track reference state and the term to avoid collision based on the potential field is changed depending on the distance from the parked car. Additionally, the potential field to leave from the side walls is also introduced. Here the modified cost function is as follows:

$$J_2(\boldsymbol{u}_t) = \phi^T(N|t)Q_f\phi(N|t) + \sum_{k=1}^{N-1} s_0(k|t)\left(\phi^T(k|t)Q\phi(k|t) + \Delta u(k|t)^T R \Delta u(k|t)\right), + Q_{obs}\sum_{k=1}^{N-1} s_j P_{obs}^j(k|t) + Q_{wall}\sum_{k=1}^{N} P_{wall}(k|t), \tag{13}$$

where the switching parameter s_0 and s_i are computed as:

$$s_i(k|t) = \begin{cases} \dfrac{d_{th}}{\|p(k|t) - \boldsymbol{o}^i\|_2} & \text{if} \quad \|p(k|t) - \boldsymbol{o}^i\|_2 > d_{th} \\ 1 & \text{other} \end{cases}, \tag{14}$$

$$s_0(k|t) = \prod_{j=1}^{N_O}(1 - s_j(k|t)), \tag{15}$$

and Q_{obs} and Q_{wall} are the weight parameters for the potential field of parked cars and walls, respectively. d_th is threshold on distance between ego car and ith obstacle. s_i represents a smooth membership function whether the ith obstacle affect on the ego vehicle trajectory. P_{obj}^j is the potential function for jth parked car and given as follows:

$$P_{obj}^j(k|t) = C\exp\left(-\left(\frac{p_x(k|t) - o_x^j}{r_a^j}\right)^2 - \left(\frac{p_y(k|t) - o_y^j}{r_b^j}\right)^2\right), \tag{16}$$

where C, r_a^i, r_b^i are the parameters to adjust the magnitude and area of influence of the potential field, $\boldsymbol{p}(k|t) = [p_x(k|t), p_y(k|t)]$ is the predicted ego-car position at kth step in prediction horizon at the time of t, and $\boldsymbol{o}^j = [o_x^i, o_y^i]$ is position of ith parked car. P_{wall} is the potential field of side walls and given as follows:

$$P_{wall}(k|t) = (\log|y_{wl}| + \log|y_{wr}|) - (\log(y_{wl} - p_y(k|t) + \log(p_y(k|t) - y_{wr})), \tag{17}$$

where $y_{wl} = 3$ and $y_{wr} = -3$ are the side wall positions.

In (Eq. 13), the switching parameter s_i for ith parked car becomes close to 1 when ego-car approaching the car. On the other hand, the deviation from reference state is more emphasized if ego-car leave the parked car. This means that the repulsive force from the parked car become more significant to achieve smooth collision avoidance and the path tracking is focused otherwise. The cost function and those parameters introduced here were set by try and error.

G. Formulation of input optimization problem

Finally the optimization problem to obtain the control input series $\boldsymbol{u}_t = [u(0|t), u(1|t), \cdots, u(N-1|t)]$ at the time of t can be summarized as follows:

given:

$$x(t) = x(0|t),\ x^{ref}, \tag{18}$$

find:

$$u(k|t), x(k|t), \qquad (k \in \{1, 2, \cdots, N\}) \tag{19}$$

which minimize:

$$J_2(\boldsymbol{u_t} = \{u(k|t)\}), \qquad (k \in \{0, 1, 2, \cdots, N-1\}) \tag{20}$$

subject to:

Prohibited area constraint (Eq. 9)

Car dynamics (Eq. 6), (Eq. 7), (Eq. 8). (21)

This problem is solved every control cycle by sampling based method explained in next section and receding horizon method is applied in real-time.

III. Randomized approach for semi-global solution

Since the optimization problem introduced in previous section includes nonlinear constraints and must be solved in real-time in MPC frameworks, sample based approach is applied to obtain the semi-optimal solution in this paper. At first, multiple series of control input, $\boldsymbol{u}_t$, are sampled from specified probability distribution and the state sequence in prediction horizon are computed by a state space model for each sampled input. Infeasible control input which violate the constraints are rejected and the others are evaluated by the defined cost function based on current measurement. Next, one of sampled input series which provide the most minimum cost function is selected as the semi-optimal solution. Then the first element of the semi-optimal input series, $u^*(0|t)$, is applied to the ego-car. These steps are ran iteratively every control cycle like usual MPC control frame work.

A. Random input sampling from random walk process

One of the simple ideas to sample the input series is to use the random walk process. N_s of control input series, $\boldsymbol{u}^i_{RW}(t)$ $(i \in \{1, 2, \cdots, N_s\})$ are computed as follows:

$$u^i_{RW}(0|t) = u(t-1), \tag{22}$$

$$u^i_{RW}(k|t) = u^i_{RW}(k-1|t) + \alpha * \Delta u^i_{RW}(k|t) \quad \forall k \in \{1, \cdots, N-1\}, \tag{23}$$

$$\Delta u^i_{RW}(k|t) \sim \mathcal{N}(0, 1), \tag{24}$$

where $\Delta u^i_{RW}(k|t)$ is corresponding to the time difference of the input and sampled from the normal distribution $\mathcal{N}(0, 1)$ and $\alpha (= 2$ in this paper) is the scaling parameter. Constraints on control input can be also considered by rejecting the sampled series $\boldsymbol{u}^i_{RW}(t)$ which violates the constrains. Examples of sampled input series with assuming $u(t-1) = 0$ are depicted in Fig.3. The figure shows the generated input is not smooth and contain much high frequency.

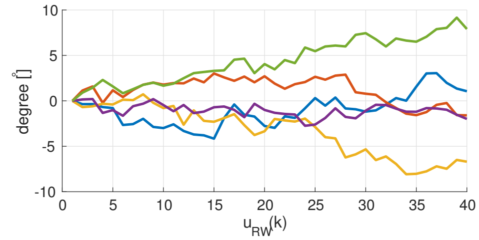

Fig. 3. Examples of generated control input based on random walk process

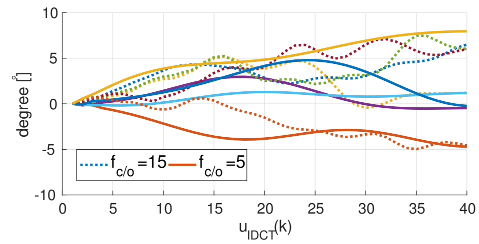

Fig. 4. Examples of generated control input based on frequency domain sampling

B. Random input sampling in frequency domain

Input sampling by random walk process is simple and it enables quick computation, however, there are two drawbacks in the generation process. One of those drawbacks is the difficulty on estimation of resulting input signal magnitude. The scaling parameter α is only adjustable parameters in random walk process, and it is hard to estimate the maximum and/or minimum values of resulting input series.

Another problem on the random walk process is a noisy nature of generated series. This is quite natural because the signal source of $\Delta u^i(k|t)$ is sampled from a normal distribution, i.e. white Gaussian noise. Since the smooth control input is preferable when we consider the application to vehicle control, the input series are sampled in frequency domain instead of direct sampling in time domain. In this paper, Inverse Discrete Cosine Transform(IDCT) is utilized to realize the frequency domain sampling instead of sampling input as time domain data. Here the sampled input series $\boldsymbol{u}^i_{DCT}$ by using IDCT transform are computed as follows:

$$u^i_{IDCT}(0|t) = u(t-1), \tag{25}$$

$$u^i_{IDCT}(k|t) = u^i_{IDCT}(k-1|t) + \Delta u^i_{IDCT}(k|t) \quad \forall k \in \{1, \cdots, N\}, \tag{26}$$

$$\Delta u^i_{IDCT}(t)^T = \gamma D U^i_{IDCT}(t)^T, \tag{27}$$

$$U^i_{IDCT}(k|t) \begin{cases} \sim \mathcal{N}(0,1) & \text{if } \ k \le F_{c/o} \\ = 0 & \text{other} \end{cases}, \tag{28}$$

where the $D \in R^{N \times N}$ is the coefficients matrix of IDCT transform and an element of D is defined as:

$$D_{ij} = \sqrt{\frac{2}{n}} k_i \cos\left(\frac{(i-1)(j-1/2)\pi}{n}\right),$$
$$i \in \{1, 2, \cdots, N\}, j \in \{1, 2, \cdots, N\}, \tag{29}$$

where N is length of the control input series to be sampled. Normalizing factor k_i $(i = 1, 2, ..., n)$ is also defined as:

$$k_i = \left\{ \begin{array}{cc} \frac{1}{\sqrt{2}} & i = 1 \\ 1 & i \neq 1 \end{array} \right\}. \tag{30}$$

$U^i_{IDCT}(t)$ are the magnitude of each frequency components and they are sampled from normal distribution. γ is the parameter to adjust the resulting input and $F_{c/o}$ is the cut-off threshold to eliminate the high-frequency components. Generated input become more smooth when $F_{c/o}$ set to be small. Examples of generated input by using IDCT with assuming $u(t-1) = 0$ are depicted in Fig.4. The figure shows the generated input by proposed method become more smooth than the one in Fig.3. In addition, the proposed sampling method consists only of independent random sampling step and matrix multiplication step once matrix D is created. Great acceleration of computation speed can be expected by applying the parallel computation technique.

IV. Validation of Proposed Sampling Method

A. Comparison of trajectories and input

In this section, two of sampling method, input series generation by random walk process(conventional) and input series generation by frequency domain sampling with IDCT(proposed), are compared in simulator experiment. The vehicle motion is simulated by vehicle dynamics simulator Carsim (Virtual Mechanics Inc.) in this simulation. The control inputs are computed by RMPC and input to the vehicle dynamics simulator. The used parameters in cost function are $Q_f = 1$, $Q = 10$, $R = 3000$, $Q_{obs} = 3000$, $Q_{wall} = 5$ and $N = 40$, respectively. Because the control interval Δt was set to 0.1, $N = 40$ means that the control system predicts the 4 second future to determine the optimal control input in MPC framework. While α is set to 2 for random walk process, γ was set to 1 by cut and try considering the coverage of input space in this paper. In this section, the parked cars' position are set as $\boldsymbol{o}_1 = (50, 0.85)$ and $\boldsymbol{o}_1 = (80, -0.85)$. For the number of sampling, N_s should be set to $N_s = 500$ because the condition $N_s > 458$ is obtained from the discussion on the probabilistic optimal condition [15] under the condition of confidence level $\alpha_{believe} = 0.01$ and the confidence $\delta_{believe} = 0.01$. Please see [15] for the detail of estimation of N_s. The resulting trajectories by setting $N_S = 50$, 200 and 500 are shown in Fig.5 to check the effect of changing the number of sampling. The comparison between trajectories of ego-car with IDCT and with out IDCT is also shown in Fig.6. Horizontal axis of Figs. 5 and 6 shows $p_x(t)$ and vertical axises are the $p_y(t)$, $u(t)$ and $theta(t)$, respectively.

In Fig.6, we can see that the trajectory is approaching to the prohibited area which is shown as the green ellipse

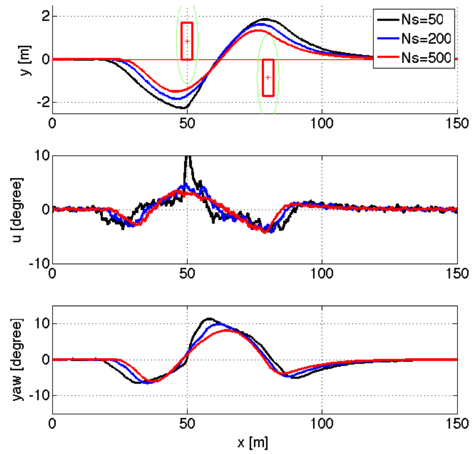

Fig. 5. Comparison of the number of sampling(with IDCT)

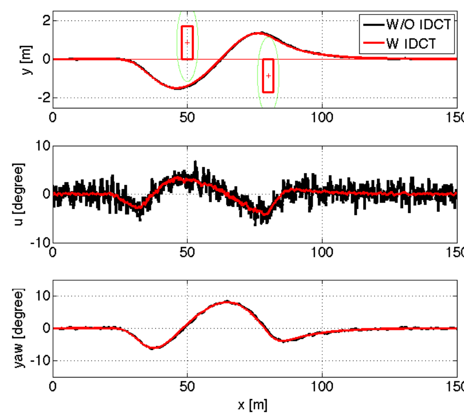

Fig. 6. Simulated trajectory with MPC($Ns = 500$)

around the parked cars as N_S increasing. It is obvious that the optimal trajectory touches the prohibited area because the input signal is minimized in optimization process. The result shows that the obtained result converges to the optimal solution with high probability by setting larger value of N_S and $N_S = 500$ is enough to get approximately optimal solution.

On the other hand, we can see that the smooth control input are generated by using frequency sampling method in Fig.6 while the obtained trajectory is quite similar to each other.

B. Comparison by computational burden

The computational burden is another big concerning to real application of the RMPC framework. The computation time using the conventional method based on random walk process and using proposed method are compared in Table II together with the average of deviation from the center of lane during $s_0 \approx 1$. $s_0 \approx 1$ means that the area

TABLE II
MEASURED COMPUTATIONAL TIME

	$N_s = 200$		$N_s = 500$	
	w IDCT	w/o IDCT	w IDCT	w/o IDCT
comp. time [s]	0.029	0.024	0.071	0.062
avg. deviation [m]	0.014	0.023	0.011	0.013

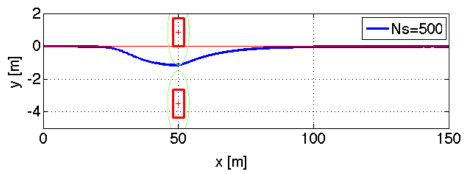

Fig. 7. Driving through small gap scenario(with IDCT)

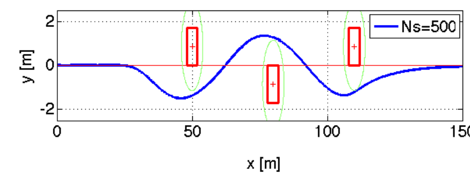

Fig. 8. Three cars scenario (with IDCT)

where ego car is not affected by any repulsive potential of parked cars. It is confirmed that the proposed method provides smaller deviation from the lane center because of the smoothness of generated input. On the other hand, although the computational time was increased in proposed method, it is also confirmed that the computational time is enough small to solve the problem within the control interval Δt. Note that the computation time is including not only the necessary time for input series sampling but includes all computation time, for state prediction, for checking the feasibility and for vehicle dynamics simulation.

C. Application to different driving scenarios

Finally the proposed method is tested to two different driving scenarios. Figure 7 shows the driving scenarios with setting $\boldsymbol{o}_1$=(50,0.85) and $\boldsymbol{o}_2$=(50,-3.5). The gap between two parked car is 0.35[m] considering the area of prohibited area in this scenario. Figure 8 shows the driving scenarios with 3 parked cars by setting $\boldsymbol{o}_1$=(50,0.85), $\boldsymbol{o}_2$=(80,-0.85) and $\boldsymbol{o}_3$=(110,0.85), respectively. The validity of the proposed method is also confirmed even in relatively difficult driving scenario.

V. CONCLUSION

This paper presented sample-based approach for the Model Predictive Control(MPC) containing non-linear constraints that can generate smooth control input. Candidate input series were sampled in frequency domain by utilizing the Inverse Discrete Cosine Transform(IDCT) to transform the frequency domain samples to time domain input series with band-width limitation. The validity of proposed method was checked through the simulation experiment using vehicle dynamics simulator. Thanks to fast computation feature of IDCT, it is also confirmed that the computation time of proposed method is enough small to realize real-time control and the proposed method is efficient and practical. In addition, the real car experiment with proposed method for collision avoidance using small EV platform will be introduced in our presentation.

Motion planning and control by using proposed method for the collision avoidance problem to moving obstacles is one of our future works.

REFERENCES

[1] M. Abe,*Vehivle Handling Dynamics -Theory and Application-*, Butterworth-Heinemann, 2009

[2] B. Paden, M. Cap, S. Z. Yong, D. Yershov and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,” IEEE Trans. on Intelligent Vehicles, Vol. 1, No. 1, pp. 33–55, 2016.

[3] J. P. Laumond, P. E. Jacobs, M. Taix and R. M. Murray, “A motion planner for nonholonomic mobile robots,” IEEE Trans. on Robo. and Auto., Vol. 10, No. 5, pp. 577–593, 1994.

[4] L. Han, H. Yashiro, H. Tehrani Nik Nejad, Q. H. Do and S. Mita, “Bezier curve based path planning for autonomous vehicle in urban environment,” 2010 IEEE Intelligent Vehicles, pp. 1036–1042, 2010

[5] S.M.LaValle, “Rapidly-exploring random trees: A new tool for path planning,” Int'l J. of Robotics Research, vol.20, no.5, pp. 378–400, 2001.

[6] S. A. Arogeti and N. Berman, “Path Following of Autonomous Vehicles in the Presence of Sliding Effects,” in IEEE Trans. on Vehicular Tech., vol. 61, no. 4, pp. 1481–1492, 2012.

[7] A. Koga, H. Okuda, Y. Tazaki, et al., “Autonomous Lane Tracking Reflecting Skilled/Un-skilled Driving Characteristics,” 42th Annu. Conf. of the IEEE Industrial Ele. Soc., pp.3175–3180, 2015.

[8] M.Oshima and M.Ogawa, “Model Predictive Control-I : Basic Principle : history & present status,” Trans. on the Institute of Sys., Cont. and Info. Eng., vol.46, NO.5, 2002

[9] M.Kano and M.Oshima, “Model Predictive Control-II : Linear Model Predictive Control,” Trans. on the Institute of Sys., Cont. and Info. Eng., vol.46, NO.7, 2002

[10] P. Falcone, F. Borrelli, J. Asgari, et al., “Predictive Active Steering Control for Autonomous Vehicle Systems,” IEEE Trans. on Control Sys. Tech., Vol.15, No.3, 2007.

[11] H. Okuda, N. Sugie and T. Suzuki, “Real-Time Collision Avoidance Control Based on Continuation Method for Nonlinear Model Predictive Control with Safety Constraint,” 2017 Asian Control Conf., pp. 1086–1091, 2017

[12] C. Gotte, M. Keller, C. Hass, K. H. Glander, A. Seewald and T. Bertram, “A model predictive combined planning and control approach for guidance of automated vehicles,” 2015 IEEE Int'l Conf. on Vehicular Electronics and Safety, pp. 69–74, 2015

[13] J. L. Piovesan and H. G. Tanner, “Randomized model predictive control for robot navigation,” 2009 IEEE Int'l Conf. on Robotics and Automation, pp. 94–99, 2009.

[14] G. Schildbach, G. C. Calafiore and L. Fagiano, “Randomized Model Predictive Control for Stochastic Linear Systems,” 2012 American Control Conf., pp. 417–422, 2012.

[15] M. Vidyasagar, “Randomized algorithms for robust controller synthesis using statistical learning theory,” Automatica, Vol. 37, No.10, pp. 1515–1528, 2001.

[16] D. D. Dunlap, E. G. Collins Jr. and C. V. Caldwell, “Sampling Based Model Predictive Control with Application to Autonomous Vehicle Guidance,” 2008 Florida Conf. on Recent Advances in Robotics, 2008.

[17] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad and I.V. Kolmanovsky, “A Stochastic Model Predictive Control Approach for Series Hybrid Electric Vehicle Power Management,” 2010 American Control Conf., pp. 5844–5849, 2010.

[18] X. Zhang , S. Grammatico, G. Schildbachc, P. Goulart and J. Lygeros, “On the Sample Size of Random Convex Programs with Structured Dependence on the Uncertainty,” Automatica, Vol. 60, pp. 182–188, 2015.