Optimal Control Based Trajectory Planning Under Uncertainty

Shangyuan Zhang[1,2](✉), Makhlouf Hadji[2], and Abdel Lisser[1]

[1] CentraleSupelec, L2S, Université Paris Saclay, 3 Rue Curie Joliot, 91190 Gif-sur-Yvette, France
abdel.lisser@l2s.centralesupelec.fr

[2] Institut de Recherche Technologique SystemX, 8 Avenue de la Vauve, 91120 Palaiseau, France
{shangyuan.zhang,makhlouf.hadji}@irt-systemx.fr

Abstract. In this paper, we propose a constrained optimal control approach as a reference trajectory generator for a driving scenario with uncertainty. With a given scenario, this generator can produce a reference trajectory in order to make validations for autonomous vehicle's decision-making problems. The constrained optimal control problem guarantees obtaining a collision-free trajectory with safety and comfort based on the design of the objective function and the constraints of the vehicle. The uncertainty of environmental information provided by sensors is taken into account, and a stochastic optimization problem is proposed to limit the risk of violating safety requirements. Numerical experiments show that the stochastic model can better ensure the robustness of the obtained solutions.

Keywords: Autonomous vehicle · Trajectory planning · Stochastic optimization · Optimal control · Chance constraint

1 Introduction

Autonomous vehicles have been an active research area of both academia and industry in recent years as a way to achieve a safer and more effective mode of transportation. Various challenges are faced in different subtasks, and trajectory planning is one of them. An autonomous vehicle's decision-making module has to determine a collision-free and feasible trajectory that takes the vehicle from its current position to its destination in a dynamic and unpredictable driving environment.

Because of the complexity of the autonomous vehicle's system and the possible failure of coherence among the modules, the algorithm's performance may not be initially envisioned, and malfunctions may sometimes occur. It is, therefore, essential to validate the reliability and safety of autonomous vehicles prior to their commercialization. Furthermore, there is no practical way to conduct a

A. L. Martins et al. (Eds.): INTSYS 2022, LNICST 486, pp. 73–88, 2023.
https://doi.org/10.1007/978-3-031-30855-0_5

complete on-road vehicle-level test that covers all driving scenarios. A promising solution is to make the assessment through driving scenarios in simulations to reduce costs and effort. Thus, to evaluate the performance of the decision-making module of an autonomous vehicle, a reference generator is used to generate a reference trajectory and compare the original trajectory with this reference trajectory in order to make an assessment.

In addition, uncertainties arising from sensor noise, measurement fluctuations, and weather conditions must be taken into account to achieve maximum practical performance. These factors have a huge impact on safety assessment. The reliability of trajectory generators depends on their ability to model uncertainty effectively and ensure that the generated trajectory is robust and promising.

The main contribution of this paper is to propose a numerical optimal control method to formulate the reference generation problem under various driving scenarios. The model considers factors like safety, comfort, and effectiveness, as well as features such as adaptive cruise control and lane-keeping assist. In addition, we also address uncertainty and employ a chance constraint to assess safety.

The remainder of this paper is organized as follows: In Sect. 2, state-of-art trajectory planning techniques are discussed. Section 3 describes the formulation of our reference generation model and the stochastic component. Section 4 presents the numerical experiments and compares the performance of different approaches. Conclusions and future perspectives are provided in Sect. 5.

2 Related Work

There has been an increasing amount of trajectory planning techniques derived from robotics and adapted to the application of autonomous vehicle decision-making. A variety of approaches has been developed in order to meet different needs across a wide range of driving scenarios.

One widely applied approach is the sampling-based method, including the Rapidly-exploring Random Tree (RRT) [1] and its variants [2–4]. RRT explores the configuration space with a random search for connections in order to obtain a feasible trajectory. It's widely used in online path planning problems. In [5], improvement has been made by removing the steering function in RRT while providing asymptotic near-optimality for kinodynamic planning. Despite the fact that sampling in a semi-structured space allows for fast planning, there is no guarantee for the optimality and continuity of the results.

There is also interest in graph search-based algorithms at different planning levels. The A* algorithm uses heuristics to perform the process of online trajectory planning efficiently with a fast node search. While it is useful for searching areas known to the vehicle a priori, it is slow and memory-intensive for large areas. Its extensions like AD* [6], ARA* [7] and AIT* [8] aim to find a suboptimal solution quickly by executing an A* with inflated heuristics and then refine the solution incrementally. Typically, a heuristic rule for this type of algorithm is not straightforward to find in complex driving environments, and the trajectory is not always smooth.

In order to generate a smooth trajectory that respects the precise constraints, optimization methods have demonstrated their effectiveness. In [9], The optimal control solutions are presented and analyzed for various maneuvers for models of different complexities. Another example is [10], in which numerical optimal control and homotopy methods are combined for motion planning in nonconvex environments, the problem is formulated as Sequential Quadratic Programming (SQP), and the obstacles are categorized according to their topological properties. The optimization methods typically encode constraints and objectives in continuous optimal control, transform them into a nonlinear optimization problem, and then develop efficient and reliable algorithms to solve it.

3 Problem Formulation

3.1 Driving Scenario

In this section, we break down all the elements of the input to our reference generator model. The driving scenario could be understood as such: a driving scenario is the precise description of all the components in the environment and the trajectory information of all other traffic actors, including vehicles, pedestrians, etc., over a period of time (typically $\leq$ 30s). It could be a simulator or a real-life driving scenario, and the information could be retrieved and stored properly to reproduce it.

The necessary information on the road and other moving objects should be available in a typical driving scenario as follows:

- The trajectory of the ith vehicle at time t: this is represented and noted by $X_i(t), Y_i(t)$.
- The ith center lane of the road is noticed by $C_i(x)$.
- The boundary of the road noted by $B_i(x)$.
- Regulations and code of the road, including maximal speed, which is indicated by v_{max}.

In order to produce the reference trajectory, we also need our ego vehicle's planning information as follows:

- The initial state of the ego vehicle: z_0.
- Predefined way-point as an indicator of the expected maneuver (lane change, overtaking, steady driving, etc.).
- The measure of the optimality that based on an evaluation function.
- Constraints of the vehicle: cinematic and dynamic constraints and consideration of passenger's comfort.

Our reference generator needs to take that information as input and output the reference trajectory for the ego car given the driving scenario.

3.2 Optimal Control Problem

The objective of our work is to develop a reference trajectory generator with various functionalities in a driving scenario. With a scenario as input, this reference path generator should output a list of commands to execute to create a reference trajectory for further evaluation. The reference trajectory must maintain a safe distance from the leading vehicle and drive in the middle of the lane, taking into account all relevant vehicle constraints. This problem can be well-defined under the optimal control framework.

In an optimal control problem, the optimal control input $u(t)$ is determined to minimize an objective function $\ell(\cdot)$ while respecting the system dynamics $f(\cdot)$ and proposed constraints. The problem is formulated as follows:

$$\begin{aligned} \min_{z(\cdot),u(\cdot)} \quad & \int_{t_0}^{t_n} \ell(z(t),\ u(t))dt \\ \text{s.t.} \quad & \dot{z}(t) = f(z(t), u(t)), \\ & c(z(t), u(t)) \leq 0, \\ & z(t_0) = z_{\text{init}}, \quad z(t_n) = z_{\text{term}}, \\ & z(t) \in \mathcal{Z}, \quad u(t) \in \mathcal{U}. \end{aligned} \tag{1}$$

where t_0 and t_n are start time and end time, z_{init} and z_{term} are inital and terminal states, $c(\cdot)$ is the inequality constraint function, $\mathcal{Z}$ and $\mathcal{U}$ are feasible sets of states and control inputs.

Various methods are available for solving this continuous optimal control problem, which can be classified into direct and indirect methods. Through indirect methods, the original problem is first transformed into a boundary value problem and then solved numerically, while direct methods solve the nonlinear optimization problem through the discretization of the integral form.

For a given period of time, the whole duration is equally divided in N phase $[t_0, t_1, ...t_i, \dots\ t_n]$ where $t_{i+1} = t_i + dt,\ \forall i \in \{0, 1, \dots\ n-1\}$, and dt is the duration of one frame during which the state and control inputs are constant.

$$\begin{aligned} \min_{\mathbf{u},\mathbf{z}} \quad & \sum_{k=1}^{k=n} \ell(z_k, u_k) \\ \text{s.t.} \quad & z_{k+1} = z_k + f(z_k, u_k)dt, \\ & c(z_k, u_k) \leq 0, \\ & z_0 = z_{\text{init}}, \quad z_n = z_{\text{term}}, \\ & z_k \in \mathcal{Z}, \quad u_k \in \mathcal{U}, \\ & k = 0, 1, \cdots n. \end{aligned} \tag{2}$$

Once this nonlinear optimization problem is solved, we can re-establish the reference trajectory using the initial state of the vehicle and the result commands.

3.3 An Example of Our Module

In the sequel, we provide an explicit example of the modeling of a reference trajectory generator.

We chose the unicycle kinematic model as the vehicle model for trajectory planning. The state of the ego vehicle at time k is given by:

$$z_k = [x_k, y_k, \theta_k, v_k]^T,$$

where x is the longitudinal position, y is the lateral position, θ is the heading angle, and v is the speed.

The control input at time k is given by

$$u_k = [a_k, \omega_k],$$

where a_k is the linear acceleration and ω_k is the angular velocity.

And the ego vehicle's control-state relationship is :

$$z_{k+1} = z_k + f(z_k, u_k)dt, \tag{3}$$

where $f(z_k, u_k) = [v_k \cos\theta_k, v_k \sin\theta_k, \omega_k, a_k]^T$.

We can formulate the reference trajectory generation as an optimal control problem in discrete form, which gives the following nonlinear optimization problem (NLP).

$$\begin{aligned} \min_{\mathbf{u},\mathbf{z}} \quad & \sum_{k=1}^{k=n} \{w_g D_k^2(x_k, y_k) + w_v(v_r - v_k)^2 + w_a a_k^2 \\ & + w_\omega \omega_k^2 + w_j(a_k - a_{k-1})^2 + w_h H(\theta_k)^2 \\ & + w_p P(x_k^{tgt}, y_k^{tgt}, x_k, y_k)\} \end{aligned} \tag{4}$$

$$\begin{aligned} \text{s.t.} \quad & z_{k+1} = z_k + f(z_k, u_k)dt, && \text{(4a)} \\ & L_(x_k, y_k) <= 0, && \text{(4b)} \\ & |v_k| \leq v_{max}, && \text{(4c)} \\ & |\omega_k| \leq \omega_{max}, && \text{(4d)} \\ & |a_k| \leq a_{max}, && \text{(4e)} \\ & |a_k - a_{k-1}| \leq j_{max}, && \text{(4f)} \\ & K(x_k^{tgt}, y_k^{tgt}, x_k, y_k) \geq d_{min} \\ & k = 0, 1, \cdots n. && \text{(4g)} \end{aligned}$$

where $\mathbf{u}$ and $\mathbf{z}$ are vectors, including all the discrete control inputs and states during the scenario.

The objective function (4) consists of several different terms to regulate the behavior of the ego vehicle. $D_k^2(x_k, y_k)$ is the distance to the waypoint at time

k. Minimizing their sums allows the vehicle to travel at the desired speed while staying at the center line. $(v_r - v_k)^2$ regulates the vehicle's actual speed to the desired speed. a_k^2 and ω_k^2 penalize the large control input, and minimizing the jerk term $(a_k - a_{k-1})^2$ improves the comfort of passengers in the vehicle. $H(\theta_k)^2$ drives the vehicle to align its heading with the curvature of the center lane. The potential field function $P(d_k)$ is based on the headway distance d_k to the leading vehicle. This term is used to regulate the headway distance in order to achieve ACC functionality. The weights w are chosen according to needs. They represent the importance of each factor and the trade-off among comfort, security, and effectiveness.

Constraint (4a) comes from the vehicle's kinematic model. Constraint (4b) guarantees the vehicle to drive within the road range. By interpolating polynomials, we can represent the boundaries of roads and restrict the reach of ego vehicles. Constraints (4c, 4d, 4e, 4f) present the speed limit, the actuator limits of the vehicle, and the range of jerk. The safety and comfort of passengers are ensured by those terms. Constraint (4f) is the collision avoidance constraint. The minimum distance between the ego vehicle and the heading vehicle should exceed a threshold d_{min}.

3.4 Stochastic Model

The model above assumed that the ego vehicle could obtain exact environmental information. In real-life scenarios, sophisticated sensors in ego vehicles do not always provide accurate information in complex driving scenarios. The inability to handle the involved uncertainty may lead to a security failure for autonomous vehicles. In our paper, a chance-constraint stochastic optimization model [11] is proposed to solve problems with uncertainty in order to achieve better performance.

In our stochastic model, the leading car's position at time k, (x_k^{tgt}, y_k^{tgt}), contains random noises following normal distributions due to sensor inaccuracy. Hence, we consider $x_k^{tgt} \sim N(\mu_{xk}, \sigma_{xk}^2)$ and $y_k^{tgt} \sim N(\mu_{yk}, \sigma_{yk}^2)$. Adding the random variable in the optimization problem, we need to treat the objective function and the constraints independently.

In objective function (4), the uncertainty part lies in the term $P(x_k^{tgt}, y_k^{tgt}, x_k, y_k)$, so we can replace x_k^{tgt} and y_k^{tgt} with μ_{xk} and μ_{yk} to get the approximate expectation.

The constraint (4g) is the only constraint involving uncertainty.

$$|x_k^{tgt} - x_k| + |y_k^{tgt} - y_k| \geq d_{min}$$

.

Applying triangle inequality to the left side, we have

$$|x_k^{tgt} - x_k| + |y_k^{tgt} - y_k| \geq |x_k^{tgt} - x_k + y_k^{tgt} - y_k|$$

.

Thus the constraint (4g) can be replaced by a more strict constraint

$$|x_k^{tgt} - x_k + y_k^{tgt} - y_k| \geq d_{min}$$

.

Based on the property of normal distribution, we have $r = x_k^{tgt} + y_k^{tgt}$ following normal distribution $N(\mu_{xk} + \mu_{yk}, \sigma_{xk}^2 + \sigma_{yk}^2)$ with a given a threshold α. The chance constraint can be transformed as follows:

$$\begin{aligned}
&\mathbb{P}(|x_k^{tgt} - x_k + y_k^{tgt} - y_k| \geq d_{min}) \geq \alpha, \ \forall k \\
=&\mathbb{P}(|x_k^{tgt} - x_k + y_k^{tgt} - y_k| \leq d_{min}) \leq 1 - \alpha, \\
=&\mathbb{P}(\frac{-d_{min} - \mu_{xk} - \mu_{yk} + x_k + y_k}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}} \\
&\leq \frac{x_k^{tgt} + y_k^{tgt} - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}} \leq \frac{d_{min} - \mu_{xk} - \mu_{yk} + x_k + y_k)}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}}) \quad \leq 1 - \alpha \\
=&F_N(\frac{x_k + y_k + d_{min} - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}}) \\
&- F_N(\frac{x_k + y_k - d_{min} - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}}) \leq 1 - \alpha.
\end{aligned} \tag{5}$$

This is equivalent to

$$\begin{aligned}
&\mathbb{P}(x_k^{tgt} - x_k + y_k^{tgt} - y_k \leq d_{min}) \geq \beta_1 \\
&\mathbb{P}(x_k^{tgt} - x_k + y_k^{tgt} - y_k \geq -d_{min}) \geq \beta_2 \\
&1 \leq \beta_1 + \beta_2 \leq 2 - \alpha \\
&\beta_1, \beta_2 \in [0, 1].
\end{aligned} \tag{6}$$

A sufficient condition is to consider $\beta_1 = \beta_2 = 1 - \alpha/2$. Then the above constraints can be transformed to

$$\begin{aligned}
&\mathbb{P}(x_k^{tgt} - x_k + y_k^{tgt} - y_k \leq d_{min}) \geq 1 - \alpha/2, \ \forall k \\
=&\mathbb{P}(\frac{x_k^{tgt} + y_k^{tgt} - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}} \leq \frac{d_{min} + x_k + y_k - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}}) \\
&\geq 1 - \alpha/2 \\
=&\frac{d_{min} + x_k + y_k - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}} \geq F_N^{-1}(1 - \alpha/2) \\
=&x_k + y_k \geq \mu_{xk} + \mu_{yk} - d_{min} + \sqrt{\sigma_{xk}^2 + \sigma_{yk}^2} \cdot F_N^{-1}(1 - \alpha/2).
\end{aligned} \tag{7}$$

In a similar way, we have:

$$\begin{aligned}
&\mathbb{P}(x_k^{tgt} - x_k + y_k^{tgt} - y_k \geq -d_{min}) \geq 1 - \alpha/2, \ \forall k \\
=&\mathbb{P}(\frac{x_k^{tgt} + y_k^{tgt} - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}} \leq \frac{-d_{min} + x_k + y_k - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}}) \\
&\leq \alpha/2 \\
=&\frac{-d_{min} + x_k + y_k - \mu_{xk} - \mu_{yk}}{\sqrt{\sigma_{xk}^2 + \sigma_{yk}^2}} \leq F_N^{-1}(\alpha/2) \\
=&x_k + y_k \leq \mu_{xk} + \mu_{yk} + d_{min} + \sqrt{\sigma_{xk}^2 + \sigma_{yk}^2} \cdot F_N^{-1}(\alpha/2),
\end{aligned} \tag{8}$$

where F_N is the cumulative distribution function of standard normal distribution.

Using the last constraint instead of the previous one (4g), we can get a new optimization problem to generate a reference trajectory considering sensor uncertainties.

4 Numerical Experiments

In this section, numerical tests are conducted to prove the efficiency of the reference trajectory generation model under various driving scenarios. Our driving scenarios are generated with SCANeR Studio [12], which is a commercial driving simulation software that helps develop and validate ADAS. The modules in SCANeR Studio for vehicle dynamics, environment building, and sensor modeling offer us the flexibility to define road states and conditions based on our requirements. The driving data in SCANeR Studio could be exported for further analysis. We solve nonlinear programming models with the help of the Python package GEKKO [13], which has an active set sequential quadratic programming solver for our constrained nonlinear optimization problem.

Following are descriptions of our experimental set-up, examples of results, and a comparison of the performance of our deterministic and stochastic models.

4.1 Experimentation Set-Up

Our model consists of driving-related parameters, such as the maximum speed and the minimum inter-vehicle distance, etc. Those parameters should be adjusted to reflect real-world regulations and traffic rules. In response to changing scenarios, they should be adjusted as well. For example, the reference speed, and the maximum speed in highway scenarios must be higher than those in urban driving scenarios.

Our experiment is based on an urban driving scenario. Parameters like the reference velocity, minimal distance and maximum velocity should be inferior to those in highway driving. The table below lists the values of parameters during the numerical experiments:

Table 1. Parameters' values during the simulation.

Parameter	meaning	Value
v_r	reference velocity	12 m/s
dt	time step	0.05 s
d_{min}	minimum distance	5 m
v_{max}	maximum velocity	40 m/s
ω_{max}	maximum angular velocity	$\pi/6\ s^{-1}$
j_{max}	maximum jerk	0.6 m/s^2

Another set of parameters is weights in the objective function, representing the importance of corresponding terms to optimize. In order to achieve the best performance of the model, the weights should be fine-tuned from an engineering perspective.

The stochastic model takes into account the uncertainty of the sensors, which results in errors in the position of the leading vehicle. Thus the detected leading vehicle's position is equal to the real leading vehicle's position plus a random noise with a normal distribution $N(0, 1)$.

4.2 Examples of Solutions

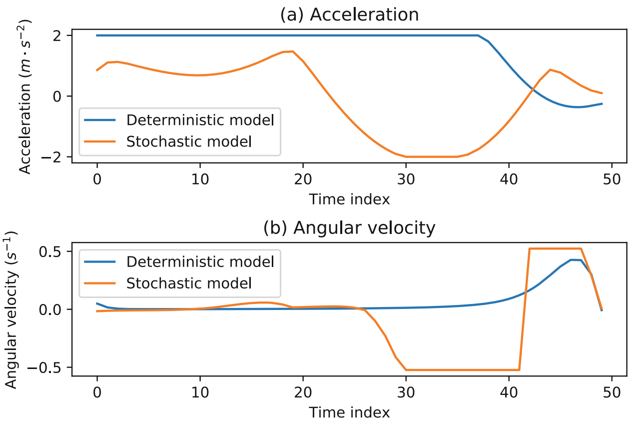

Fig. 1. Result of an example scenario.

The following result has been obtained in a driving scenario generated in SCANeR, using the deterministic and stochastic models with the previously indicated parameters.

In Fig. 1, we show the generated acceleration and angular velocity profile during the scenarios to get the optimal reference trajectory.

Table 2. Comparison under different configurations.

$w_g : w_v : w_a : w_\omega : w_j : w_h : w_p$	5:1:1:1:1:1:1			1:5:1:1:1:1:1			1:1:5:1:1:1:1			1:1:1:5:1:1:1		
N	20	40	60	20	40	60	20	40	60	20	40	60
Average CPU time(s)	0.36	0.68	2.38	1.34	5.93	3.52	1.18	12.48	3.39	0.73	1.05	2.66
Average acceleration	1.62	1.84	1.88	0.57	0.74	0.72	0.34	1.24	1.47	1.05	1.53	1.59
Average angular velocity	0	0	0.09	0	0	0.04	0	0	-0.07	0	0	0.08
Average velocity	14.80	15.82	16.82	14.36	15.32	16.19	14.13	15.54	16.63	14.61	15.74	16.73
Average distance	20.72	19.80	18.50	20.82	19.94	18.52	20.92	19.88	18.49	20.76	19.81	18.50
$w_g : w_v : w_a : w_\omega : w_j : w_h : w_p$	1:1:1:1:5:1:1			1:1:1:1:1:5:1			1:1:1:1:1:1:5			1:1:1:1:1:1:1		
N	20	40	60	20	40	60	20	40	60	20	40	60
Average CPU time(s)	1.24	2.56	3.92	0.55	4.07	3.82	0.82	3.51	3.54	0.95	3.06	3.25
Average acceleration	1.04	1.53	1.59	1.03	1.55	1.61	1.05	1.53	1.60	1.05	1.53	1.60
Average angular velocity	0	0	0.08	0	0	0.08	0	0	0.08	0	0	0.08
Average velocity	14.61	15.74	16.73	14.56	15.74	16.74	14.61	15.74	16.73	14.61	15.74	16.73
Average distance	20.76	19.81	18.50	20.78	19.82	18.50	20.76	19.81	18.50	20.76	19.81	18.50

4.3 Effects of Different Configurations

In Table 2, we considered various values of the objective weights proportion and the number of sampling times N to analyze its impact on the results and calculation.

We can observe that the average CPU time is always constant, and when the weight w_a, w_v is high, the acceleration is low, and high weight w_ω leads to low angular velocity. The average CPU time increases proportionally as the number of time frames increases.

In real-life applications, the proportion of weights should be adjusted to achieve optimal performance under specific criteria.

4.4 Comparison of Robustness

This part presents two experiments that illustrate the robustness of the stochastic model under data uncertainty. Specifically, robustness is the ability to produce near-optimal solutions with fewer violations of constraints while facing the uncertainty of data. In the context of autonomous driving, it can be illustrated in two ways.

Firstly, the model needs to be robust to an unchanged driving scenario with numerous realizations of the uncertainty of data. Secondly, the robustness needs to be proved in various driving scenarios. Two experiments are designed based on these two aspects.

Our first experiment fixes the driving scenario and generates 100 realizations of its random variables X_{tgt}, Y_{tgt}. Next, we run the deterministic model and stochastic model over 100 instances and compare the number of the violated constraints.

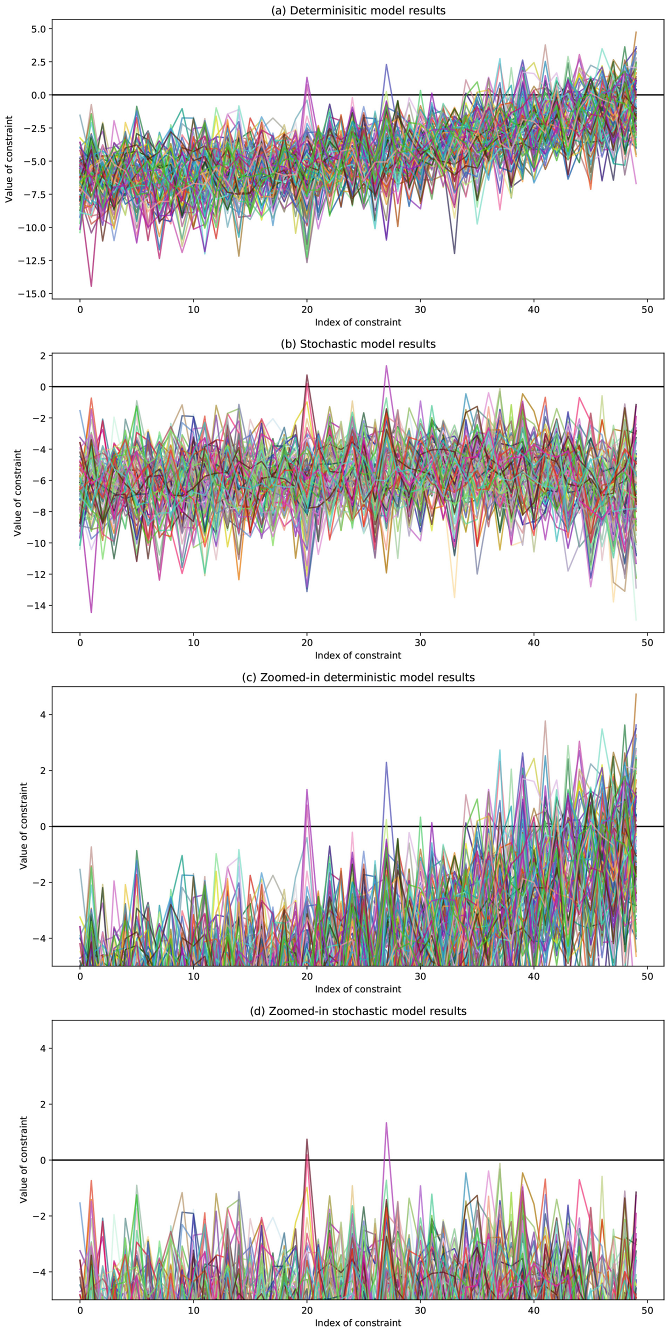

Fig. 2. Constraint function values of all instances for deterministic and stochastic models.

Figure 2 visualizes the constraint violation value $d_{min} - K(x_k^{tgt}, y_k^{tgt}, x_k, y_k)$, adapted from constraint (4g), for the whole results of the two models. Figure 2(a) and Fig. 2(b) show the constraint violation value for the whole constraints, whilst Fig. 2(c) and Fig. 2(d) show a zoom-in on a subset of constraints for better readability. In Fig. 2, each curve in its own color displays the constraint violation values of a driving scenario result, and the x-axis represents the index of constraints. If the value at constraint index i exceeds 0, it means that $d_{min} > K(x_k^{tgt}, y_k^{tgt}, x_k, y_k)$, i.e., the constraint (4g) is violated at this sampling time.

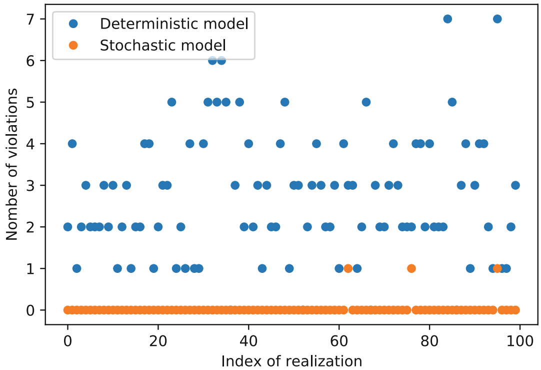

Fig. 3. Number of violated constraints of our two models.

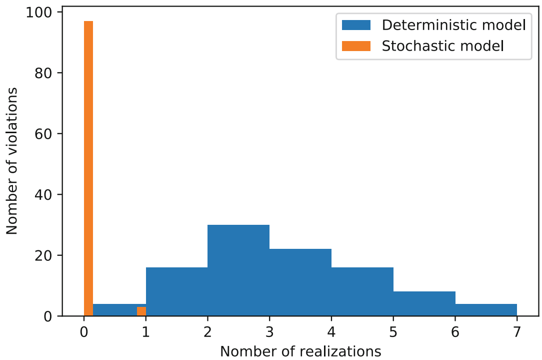

Fig. 4. Histogram of the number of violated constraints of our two models.

Figure 3 shows the number of violated constraints for simulations with 100 realizations. The blue dots represent the number of violated constraints for our

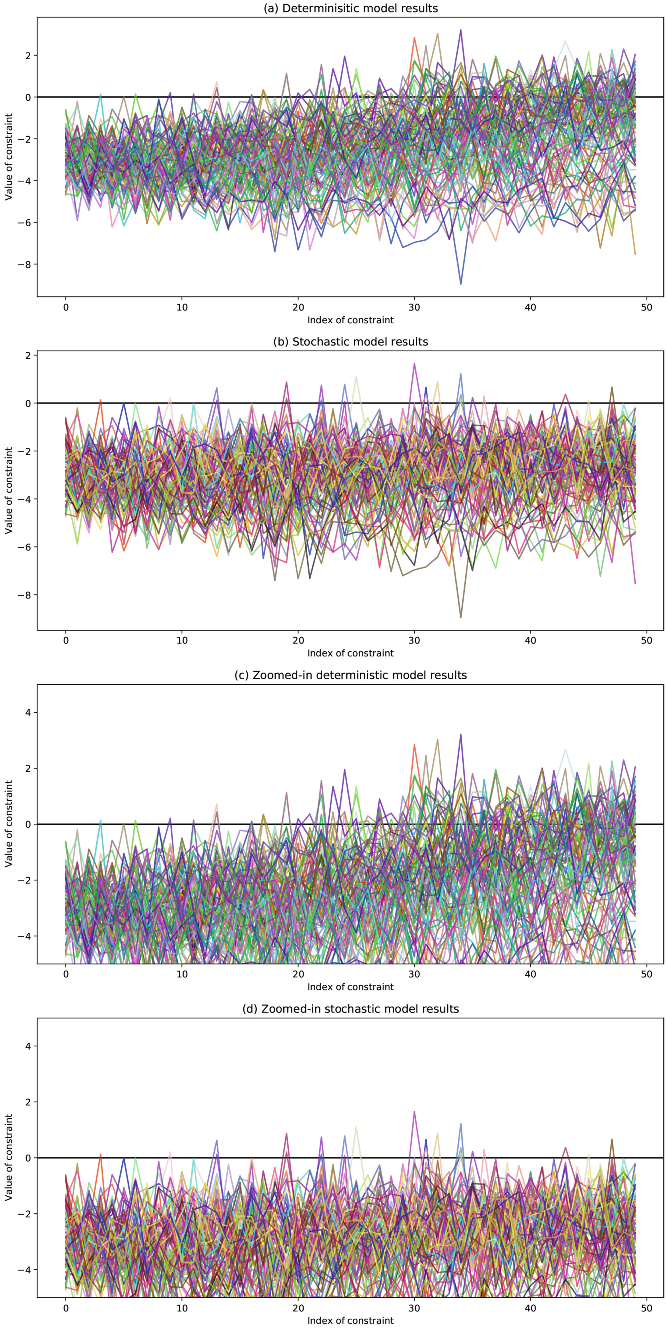

Fig. 5. Constraint function values of all instances for deterministic and stochastic models.

deterministic model, and the orange dots represent the number of violated constraints in the stochastic model. In Fig. 4, the distribution of the violated constraints is presented, and we observe that most of the stochastic models results are feasible, i.e., no constraints are violated; there are only three instances with a single constraint violation. Conversely, the constraint violations in the deterministic model are typically around two and three.

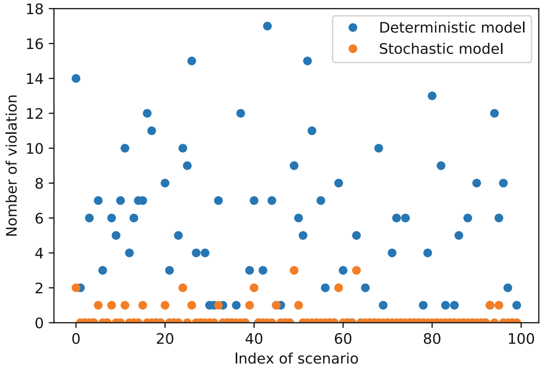

Fig. 6. Number of violated constraints of two models.

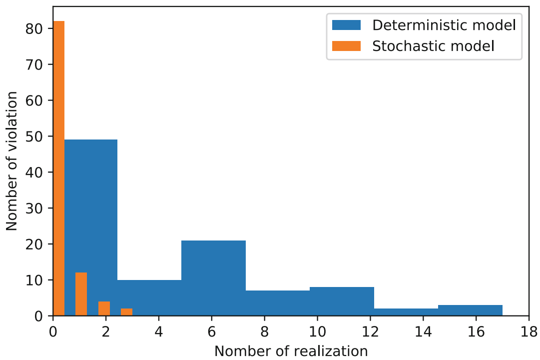

Fig. 7. Histogram of the number of violated constraints of two models.

Similarly, we conduct our second experiment with 100 different scenarios and compare the performance of our models, in the same way, to further prove our model's robustness under different scenarios. Figure 5 visualize the constraint violation value as in Fig. 2. Figure 6 and Fig. 7 present the number of violated constraints and their distribution, respectively. As a result of the diversity of scenarios, the number of violated constraints is more divergent compared with the precedent experiment. Meanwhile, the stochastic model still yields fewer constraint violations than the deterministic model.

We can conclude from the two experiments above that the stochastic model produces more robust solutions both in terms of stability and diversity.

5 Conclusions and Future Work

In this paper, an optimal control based reference trajectory generator has been proposed and implemented. With a scenario and prior waypoints, this generator can find an optimal collision-free trajectory for further validation. Stochastic programming has been used to address the uncertainty of autonomous vehicles. We also thoroughly analyse the performance of the deterministic model and the stochastic model under different configurations. Based on the comparison, it appears the stochastic model can produce more robust solutions than the deterministic model when uncertainty is present.

Future work includes the development of an increasingly sophisticated vehicle model and modeling of uncertainty involving dependent random variables. Another extension is to solve online autonomous driving planning problems with an adapted similar optimization-based framework.

Acknowledgement. This work was supported by the French government under the "France 2030" program, as part of the SystemX Technological Research Institute.

References

1. LaValle, S.M., Kuffner, J.J., Jr.: Randomized kinodynamic planning. Int. J. Robot. Res. **20**(5), 378–400 (2001)
2. Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2997–3004. IEEE (2014)
3. Perez, A., Platt, R., Konidaris, G., Kaelbling, L., Lozano-Perez, T.: LQR-RRT: optimal sampling-based motion planning with automatically derived extension heuristics. In: 2012 IEEE International Conference on Robotics and Automation, pp. 2537–2542. IEEE (2012)
4. Chen, J.: R2-RRT: reliability-based robust mission planning of off-road autonomous ground vehicle under uncertain terrain environment. IEEE Trans. Autom. Sci. Eng. **19**(2), 1030–1046 (2021)
5. Li, Y., Littlefield, Z., Bekris, K.E.: Sparse methods for efficient asymptotically optimal kinodynamic planning. In: Akin, H.L., Amato, N.M., Isler, V., van der Stappen, A.F. (eds.) Algorithmic Foundations of Robotics XI. STAR, vol. 107, pp. 263–282. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16595-0_16
6. Likhachev, M., Ferguson, D.I., Gordon, G.J., Stentz, A., Thrun, S.: Anytime dynamic A*: an anytime, replanning algorithm. In: ICAPS, vol. 5, pp. 262–271 (2005)

7. Likhachev, M., Gordon, G.J., Thrun, S.: ARA*: anytime A* with provable bounds on sub-optimality. In: Advances in Neural Information Processing Systems, vol. 16 (2003)
8. Strub, M.P., Gammell, J.D.: AIT* and EIT*: asymmetric bidirectional sampling-based path planning. arXiv preprint arXiv:2111.01877 (2021)
9. Berntorp, K., Olofsson, B., Lundahl, K., Nielsen, L.: Models and methodology for optimal trajectory generation in safety-critical road-vehicle manoeuvres. Veh. Syst. Dyn. **52**(10), 1304–1332 (2014)
10. Bergman, K., Axehill, D.: Combining homotopy methods and numerical optimal control to solve motion planning problems. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 347–354. IEEE (2018)
11. Prékopa, A.: Stochastic Programming, vol. 324. Springer Science & Business Media (2013)
12. That, T.N., Casas, J.: An integrated framework combining a traffic simulator and a driving simulator. Procedia Soc. Behav. Sci. **20**, 648–655 (2011)
13. Beal, L., Hill, D., Martin, R., Hedengren, J.: Gekko optimization suite. Processes **6**(8), 106 (2018)