Responses to reviewer 12

Corrections to your comments are highlighted in colors ■ and ■. In the revised version, we have made the following changes to the symbols and terminologies used in the manuscript.

Original version	Revised version
Controller,disturbance	Player I, Player II
$a \in \mathcal{A}$, $d \in \mathcal{D}$	$a \in \mathcal{A}$, $b \in \mathcal{B}$
$\mathscr{A}$, $\mathscr{D}$	$\mathscr{A}_0$, $\mathscr{B}_0$
Given time horizon $\bar{T}$	T or $T_1, ..., T_M$
Given admissible cost $\bar{J}$	J or $J_1, ..., J_M$
$\phi_{t_0}^{t_1}(., s_0, a, d)$	$\phi_{s,t}^{a,b}(.)$
$\min\limits_{a(.)} \max\limits_{d(.)} \; t_f$	$\sup\limits_{\beta(.)} \inf\limits_{a(.)} \; t_f$ (Emphasize that Player II plays nonanticipative strategies)
$J_{t_0}^{t_1}(s_0, a, d)$	$\mathcal{J}_{t_0}^{t_1}(s, a, b)$
$\min\limits_{a(.)} \max\limits_{d(.)} \; J_{t_0}^{t_1}(s_0, a, d)$	$\sup\limits_{\beta(.)} \inf\limits_{a(.)} \; \mathcal{J}_{t_0}^{t_1}(s, a, \beta[a])$
Given final time t_f	$\bar{T}$ (The symbol of the free final time t_f remains unchanged)
$\widehat{W}_{t_f}^c(.)$	$W_{\bar{T}}^c(., 0)$, $\left(W_{\bar{T}}^c(s, \tau) = \sup\limits_{\beta(.)} \inf\limits_{a(.)} \; \mathcal{J}_{\tau}^{\bar{T}}(s, a, \beta[a]) \right)$

1. The paper introduces an interesting extension to computing backward reachable tubes (BRT) via an optimal control problem. The proposed method is presented and explained well, however, the advantages compared with traditional methods of computing BRTs are not clear.

Response: The most important difference between our method and the level set method is the difference in the constructed value functions. In the level set method, since the value function $V(., .)$ is Lipschitz continuous, it can be represented by the viscosity solution of the following HJ PDE.

$$\begin{cases} \dfrac{\partial V}{\partial t}(s,t) + \min\left[0, \min\limits_{a \in \mathcal{A}} \max\limits_{b \in \mathcal{B}} \dfrac{\partial V}{\partial s}(s,t) f(s,a,b)\right] = 0 \\ \text{s.t.} V(s,T) = l(s) \end{cases} \tag{1}$$

It can also be written in the following backward recursive form:

$$\begin{aligned} &V(s,T) = l(s) \\ &V(s, t - \Delta t) = \min\left[V(s,t), \min_{a \in \mathcal{A}} \max_{b \in \mathcal{B}} V\left(F(s,a,b), t\right)\right] \end{aligned} \tag{2}$$

where $s(t + \Delta t) = F(s(t), a(t), b(t))$ is the time discrete form of $\dot{s} = f(s, a, b)$. Our recursive form is:

$$\begin{aligned} &W_{\bar{T}}^c(s, \bar{T}) = 0 \\ &W_{\bar{T}}^c(s, t - \Delta t) = \min_{a \in \mathcal{A}} \max_{b \in \mathcal{B}} \left[\widehat{C}(s,a,b) + W_{\bar{T}}^c(\widehat{F}(s,a,b), t)\right] \end{aligned} \tag{3}$$

This value function brings great benefits, one of which is the small storage space consumption. In the level set method, the BRT is:

$$R_K(T) = \{s_0 \in \mathbb{R}^n | V(s_0, 0) \leq 0\} \tag{4}$$

This means that to save $R_K(T)$, one needs to save value function $V(., 0)$. In this case, the storage space consumed is proportional to $\prod_{i=1}^n N_i$, where N_i is the number of grid points in the ith dimension of the Cartesian grid. For $T_1, ..., T_M \in [0, \infty)$, the representations of the BRTs of these time horizons are:

$$\begin{aligned} &R_K(T_1) = \{s_0 \in \mathbb{R}^n | V(s_0, T - T_1) \leq 0\} \\ &... \\ &R_K(T_M) = \{s_0 \in \mathbb{R}^n | V(s_0, T - T_M) \leq 0\} \end{aligned} \tag{5}$$

Since the value functions $V(., T-T_1), ..., V(., T-T_M)$ are each different, the storage space consumption required to save these BRTs is proportional to $M \prod_{i=1}^{n} N_i$, which is M times as much as the original. In the proposed method, only $W_{\bar{T}}^c(., 0)$ ($\bar{T} > \max(T_1, ..., T_M)$) needs to be saved to save $R_K(T_1), ..., R_K(T_M)$. These BRTs are represented as:

$$\begin{aligned} &R_K(T_1) = \left\{s_0 \in \mathbb{R}^n | W_{\bar{T}}^c(s_0, 0) < T_1\right\} \\ &... \\ &R_K(T_M) = \left\{s_0 \in \mathbb{R}^n | W_{\bar{T}}^c(s_0, 0) < T_M\right\} \end{aligned} \tag{6}$$

The storage space consumed does not vary with M and is always proportional to $\prod_{i=1}^{n} N_i$, which is one part in M of that of the level set method. We visually display the superiority of this storage approach in the first example of the revised manuscript.

In addition, in the first example of revised version, we compare the computational accuracy of the proposed method and the level set method, and the variations of the computational error with the number of grids are demonstrated, see Fig. 4 of the revision. We also visualize the difference between the proposed method and the level set method in terms of storage form, see Fig. 5 of the revision. The first case of the second example is also about BRTs, and we also compare the results of the proposed method with the level set method.

2. I would suggest a broader discussion on the advantages and disadvantages of the proposed methods. How does the integration scheme affect the accuracy, what is the runtime compared to solving a HJ-PDE on the same grid?

Response: In the revision, we replaced the time-discrete form from forward Euler method to fourth-order Runge-Kutta method. The results show that the error decreases as the grid size decreases, and the convergence rate is comparable to that of the level set method, and compared to the level set method, the proposed method is slightly inferior in terms of computational accuracy, see Fig. 4 of the revision. However, the proposed method can significantly reduce the storage space consumption as described above.

The runtime is related to the performance of the algorithm itself, but also to the performance of the programming language, and to the programming skill. The method in this article is implemented in C++. The level set toolbox has Matlab and C++ versions. The former is limited by the performance of the programming language, which is significantly less computationally efficient than our program. The latter makes extensive use of vectorized operations and even GPU acceleration (Limited by programming skills, our program uses only simple loops), and is therefore significantly more computationally efficient than our program. Therefore, runtime is not an appropriate measure of the performance of the algorithm itself. In the revision, we analyze the time complexity of these two algorithms. In fact, the time complexity of both is the same, and both are proportional to grids number multiplied by time steps number. The revised version provides a simple description of the time complexity of the algorithm.

3. In equation 4 you reference the sub-zero level set, however, refer to this as the zero level set, which implies an equality instead of an inequality.

Response: In the revised version, "zero level set" in the original manuscript is replaced with "zero sublevel set" and "non-zero level set" is replaced with "non-zero sublevel set". Such terminology is consistent with literature [4].

4. In equation 5 you use the minimum and maximum, however, if this PDE is to reference the Value function described in equation 3, it should refer to the infimum and supremum respectively. Furthermore, there is no information given on the information pattern between the controller and the disturbance. I believe in your context the controller should probably be playing a nonanticipative strategy. (I recommend taking a look at this paper: https://ieeexplore.ieee.org/document/5685555/). This implies that the min and max terms in equation 5 should be flipped.

Response: Regarding the choice of symbols "sup, inf" and "max, min", this paper refers to the literatures [1–4]. When the decision variable is $a(.)$ or $b(.)$, they use "sup" or "inf". When the decision variable is a or b, they use "max" or "min". In literatures [1–4], when symbols "sup, inf" are used, the symbol corresponding to the player using nonanticipative strategies is placed on the left, and when symbol "$\max, \min$" are used, the symbol corresponding to the player using nonanticipative strategies is placed on the right.

The description of the information pattern has been added in the revision. In this manuscript, Plyaer II, the player trying to drive the system state away from the target set, plays a nonanticipative strategy. There are three reasons why this information pattern is chosen:

(1) In this manuscript, the target set represents a safe portion of the state space. For example, the target set of the aircraft in Example 2. This target set is the region near the equilibrium point in the aircraft state space. In general, the state of the aircraft is always kept in the neighborhood of equilibrium. Therefore, the BRT has the meaning of "safe set".

(2) The purpose of Player I is to drive the system's state to the target set, while the purpose of Player II is to avoid or delay the entry of the system state into the target set.

(3) The player who takes nonanticipative strategy has an advantage over its opponent [4]. Player II taking nonanticipative strategy can result in an underapproximation of BRT. Since a BRT is a safe set, a favorable set, to take the worst case into account, we prefer to underapproximate it rather than overapproximate it. This treatment is similar to that of Paper [4], except that in Paper [4], the reachable set represents an unsafe set, the authors prefer to overapproximate it rather than underapproximate it.

5. The same concerns as in equation 5 apply to equation 6, what are the information patterns between the controller and the disturbance?

Response: As with the answer to the previous question, Player II plays a nonanticipative strategy. The description of the information pattern has been added in the revision.

6. In Figure 1, the CBRS and BRS are mentioned, is this meant to refer to the Backward reachable set? Furthermore, in the introduction, the RBS is mentioned, is this also meant to refer to the backward reachable set?

Response: We are very sorry, these are all slip-ups, our method cannot compute reachable sets. These mistakes are fixed in the revised version.

7. Overall, the paper needs more extensive English copy editing, but the presented method is interesting and a good contribution to the field of HJ reachability analysis.

Response: Thank you for your recognition of our work. We have double-checked the spelling of every word and the grammar of every sentence. In addition, we have sought Language Editing services from Elsevier, the Language Editing Certificate is provided as an attachment.

References

[1] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253, 2017.

[2] Jaime F Fisac, Mo Chen, Claire J Tomlin, and S Shankar Sastry. Reach-avoid problems with time-varying dynamics, targets and constraints. In *Proceedings of the 18th international conference on hybrid systems: computation and control*, pages 11–20, 2015.

[3] P. Souganidis L. Evans. Differential games and representation formulas for solutions of hamilton-jacobi-isaacs equations. *Indiana Univ. Math. J.*, 33:773–797, 1984.

[4] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.