Multi-constrained robot motion generation

Masato Miyahara[1], **Tomoki Nakao**[2], **Tatsuya Ishihara**[1], **Motohiro Takagi**[1], **Kouhei Takahashi**[1],**Taichi Kanada**[1], **Narimune Matsumura**[1], **Hitoshi Seshimo**[1]
[1]NTT Human Informatics Laboratories, NTT Corporation,
[2]Graduate School of Informatics, Kyoto University
masato.miyahara@ntt.com

Abstract: In this paper, we propose a novel robot motion generation method that simultaneously considers multiple constraints. To operate safely in the real world, robots must meet several constraints at once. For example, when carrying a glass of water, the robot needs to adjust its speed to avoid spills and avoid obstacles. However, current generative model-based methods typically only account for one constraint, and tasks with multiple constraints remain challenging. To address this, we proposed Multi-Constrained-TREBI, which uses weighted cost functions to represent the importance of each constraint. Through a simulation of a walking robot, we show that utilizing our proposed method can generate trajectories that satisfy multiple constraints.

Keywords: multi-constrained motion generation, diffusion model, safe offline RL

1 Introduction

Generative models, such as vision language models(VLM) or diffusion models[1], have advanced significantly in recent years, enabling robots to interpret human language instructions and perform various physical tasks autonomously[2, 3, 4]. These models acquire the ability to respond to instructions by training on the robot motion data labeled by the task description and learning how to move the robot to complete the task.

However, motion generation methods based on these generative models cannot account for the multiple constraints necessary for executing a motion safely. For instance, methods based on the diffusion model only consider a single constraint, such as speed, permissible region[5, 6, 7], or robot workspace[8]. As a result, it is challenging for a robot to complete everyday tasks while considering multiple unknown constraints that are not included in the training data. To illustrate, when carrying a glass of water, it is difficult for a robot to adjust its trajectory while considering its position relative to obstacles and its range of motion while limiting its speed to avoid spilling the water.

In this study, we propose a robot motion generation method, Multi-Constrained-TREBI, which considers multiple constraints simultaneously. As shown in Fig 1, our proposed method simultaneously considers the given multi-constraints, including speed, passable region, and their weights, and generates the robot's trajectory. Through experiments using a walking robot in a simulation environment, we verified that our method can impose multiple constraints on the generated robot trajectory in accordance with the weights of constraints.

2 Method

To realize a motion generation method that simultaneously satisfies multiple constraints, we aim to solve the multi-constraint trajectory optimization problem using diffusion models based on the safe offline RL framework. Given that the number of constraints is denoted as N_C, this problem can be

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

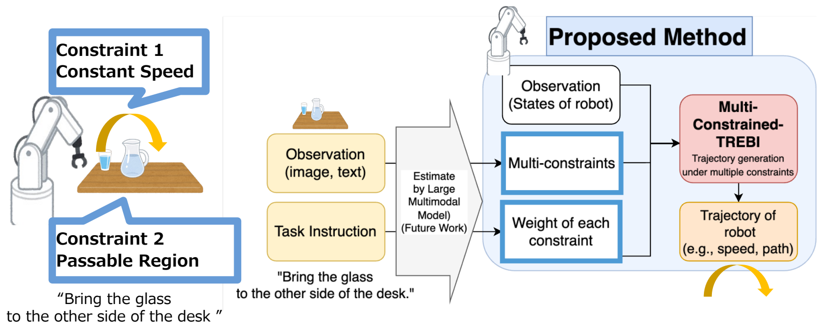


Figure 1: Framework of Proposal Method

formulated as follows.

$$\max_{q(\tau)} \mathbb{E}_{\tau\sim q(\tau)}[R(\tau)] \tag{1}$$

$$\text{s.t.} \int_{\tau:\{C_j(\tau)\leq b_j\}_{j=1}^{N_c}} q(\tau)d\tau = 1 \tag{2}$$

$$D_{KL}(q(\tau)||p_{\pi_\beta}(\tau)) \leq \epsilon. \tag{3}$$

Offline RL is the problem for finding a target distribution of the trajectory $q(\tau)$ that maximizes the expected cumulative reward $\mathbb{E}_{\tau\sim q(\tau)}[R(\tau)]$ from the perspective of trajectory optimization, given an offline data set. Here, trajectory τ consists of s and action a, such that, $\tau = \{\tau_{s_t}, \tau_{a_t}\}_{t=0}^{L}$, where L is the maximum episode length. Safe offline RL is the problem of finding a trajectory distribution that satisfies $C(\tau) < b$ and maximizes $R(\tau)$. Here, $C(\tau) = \sum_{t=0}^{L} \gamma^t c(\tau_{s_t}, \tau_{a_t})$ is the cumulative cost function, $c(\tau)$ is the cost function, which is an evaluation value for the constraints of the current trajectory, and b is the constant budget. Eq. (2) represents the set of conditions satisfying the N_C cost functions, each corresponding to a different constraint, while Eq. (3) is a constraint derived from the out-of-distribution problem in offline RL.

Several methods have been proposed to impose one constraint on the Diffuser[3] by conditioning the diffusion model[5], projection of the generated trajectories [7], and learning the feasible region as a state value function[6], etc. Trajectory-based REal-time Budget Inference (TREBI)[5] is a soft constraint-observing method based on diffusion model conditioning, and the effect of the constraints can be changed in accordance with the weight. However, none of the existing methods can handle more than a single constraint, nor can they solve our multi-constraint trajectory optimization problem.

We aim to address these challenges by decomposing the target distribution, following similar approaches used in Diffuser and TREBI. The target trajectory distribution can decompose the data distribution and the guidance distribution, where the guidance distribution serves to impose constraints on the policy. When a solution exists that simultaneously satisfies Eq. (1)∼(3), the approximate optimal distribution $q_b(\tau)$ for the target trajectory distribution can be expressed as follows.

$$q_b^*(\tau) = p_{\pi_\beta}(\tau) \prod_{j=0}^{N_c} h_{b,n}^{c_j}(\tau) \tag{4}$$

In order to sample from $q_b^*(\tau)$, we sample from $g_b = \nabla_{\tau_i} \log h_{b,n}^{c_j}(\tau_i)$ similar to the TREBI method. By introducing w_j as the weight of constraint j, instead of a single constraint weight as in TREBI, g_b can be expressed as follows.

$$g_b = \begin{cases} \alpha \nabla_{\tau_i} R(\tau_i) & (C_j(\tau_i) \leq b_j \text{ for all } j) \\ \alpha \left(\nabla_{\tau_i} R(\tau_i) - \sum_{j=1}^{N_c} w_j \nabla_{\tau_i} C_j(\tau_i) \right) & \text{(otherwise)} \end{cases} \tag{5}$$

The estimation algorithm of Eq. (1)∼(3) can is outlined in Algorithm 1, which is based on Diffuser and TREBI method. We approximate μ_θ using a diffusion model.

Algorithm 1 Multi-Constrained-TREBI

Requirements: Diffuser model μ_θ, scale parameter α, covariances Σ^i, budget for all constraints $\mathbf{b} = (b_j)_{j=1}^{N_c}$, weights for all constraints $\mathbf{w} = (w_j)_{j=1}^{N_c}$;

$\mathbf{z}_0 = \mathbf{b}$;
for $t = 0, ..., T$ **do**
  Observe state s; initialize trajectories $\tau^N \sim \mathcal{N}(0, \mathcal{I})$;
  for $i = N, ..., 1$ **do**
    if $C_j(\tau^i) \leq \mathbf{z}_{t,j}$ for all j **then**
      $g = \alpha \nabla R(\tau^i)$;
    else
      $g = \alpha \left(\nabla_{\tau_i} R(\tau_i) - \sum_{j=1}^{N_c} \mathbb{I}(C_j(\tau_i) > b_j) w_j \nabla_{\tau_i} C_j(\tau_i) \right)$
    end if
    $\mu \leftarrow \mu_\theta(\tau^i)$;
    $\tau^{i-1} \sim \mathcal{N}(\mu + \Sigma^i g, \Sigma^i)$;
    $\tau^{i-1} \leftarrow s$;
  end for
  Execute the first action τ_{a_0} and get cost $\mathbf{c} = (c_j)_{j=1}^{N_c}$;
  $\mathbf{z}_{t+1} = (\mathbf{z}_t - \mathbf{c}_t)/\gamma$.
end for

3 Experiments

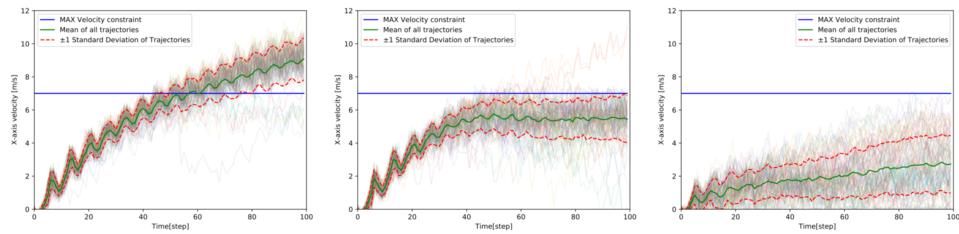
We conducted qualitative evaluation by visualizing the trajectory and calculating the average violation rate V and cumulative reward value R under the same conditions across all 100 episodes. The violation rate V is the ratio of times the constraint was violated during all time steps in each trial episode. We conducted the experiment using simulated data to evaluate whether the proposed method can simultaneously satisfy multiple constraints. We used the HalfCheetah-medium-expert-v2 of the D4RL dataset[9], the benchmark for offline RL. HalfCheetah is a task in which an agent moves at high speed in the x-axis direction on a two-dimensional plane.

We use two constraints in this study: velocity and height. The velocity constraint is to keep the velocity V_x of the agent moving in the x-axis direction consistently below the maximum velocity V_{max}, and we set $V_{max} = 7.0$m/s. The cost function of this constraint is defined as follows: $\|\max(v - V_{max}, 0.0)\|^2$. The height constraint keeps the agent's z-coordinate z consistently below the upper bound Z_{max}, and we set $Z_{max} = -0.05$m. The cost function of this constraint is as follows: $\|\max(0.0, z - Z_{max} + 0.01) \times 20\|^2$. We calculate the budget b for each constraint using the Optuna [10] for prior optimization. We set $b_1 = 141.1$ for the speed constraint and $b_2 = 284.2$ for the height constraint.

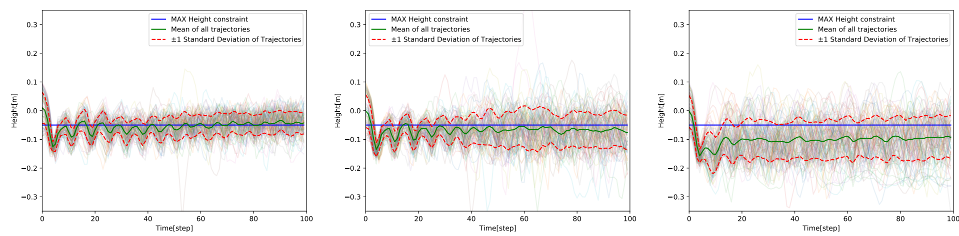
4 Results

First, we show the trajectory changes before and after constraint adaptation by the proposed method in Figure 2. The first row of Figure 2 shows the speed progressions, and the second row shows the height progressions. The left column shows the case with no constraints, and the center column shows when both constraint weights are $w_1 = w_2 = 1$, and the right column shows the case where both are $w_1 = w_2 = 100$. The horizontal axis of each graph is time, and the vertical axis corresponds to the first column for velocity changes and the second for height changes. In each figure, we plot 100 evaluation trajectories in light. The average of all trajectories is plotted in a green line, and the standard deviation interval of 1 is shown as a red dotted line.

Figure 2a and 2d are the same as when running Diffuser[3]. Figure 2a shows that the agents accelerate as the episode progresses in almost all trajectories. In Figure 2a, we can see that the robot is walking while swinging up and down. As shown in figure 2b, 2c, 2e and 2f, the generated trajectory shifts downward when the constraint is applied.



(a) Speed with no constraints (b) Speed with both constraints $w_1 = 1.0$, $w_2 = 1.0$ (c) Speed with both constraints $w_1 = 100$, $w_2 = 100$



(d) Height with no constraints (e) Height with both constraints $w_1 = 1.0$, $w_2 = 1.0$ (f) Height with both constraints $w_1 = 100$, $w_2 = 100$

Figure 2: Results of trajectory changes before and after constraint adaptation

Table 1: Results of rewards, violation rates change after constraint adaption

Constraints	w_1	w_2	$R(\uparrow)$	$V_1(\downarrow)$	$V_2(\downarrow)$
No (Diffuser[3])	–	–	**553.9**	39.7%	45.9%
Only speed constraint	1.0	–	419.4	3.25 %	34.5%
Only height constraint	–	1.0	422.4	3.37%	35.7%
Both constraints	1.0	1.0	417.9	2.8%	34.8%
Both constraints	1.0	100	348.0	3.9%	34.0%
Both constraints	100	1.0	343.1	0.5%	34.9%
Both constraints	100	100	151.8	**0.02**%	**22.8**%

Finally, we show the average cumulative reward value R and violation rates for the speed and height V_1,V_2 under each constraint condition in Table 1. Overall, increasing the weight of the constraint reduces the violation rate for both. Regarding the change in violation rates due to changes in the weight of individual constraints, the violation rate for both constraints decreases, but the height constraint is relatively less effective. In addition, in some cases, the height violation rate decreases when we increase the weight of the speed constraint because the two parameters of the speed and height constraints are related to the dynamics. The reward R decreases when the constraints are tightened, which may be because the agents are searching for the optimal strategy within the space that satisfies the constraints, which is called "temptation" in previous research[11].

5 Conclusion

We have proposed Multi-Constrained-TREBI, a robot motion generation method that can simultaneously satisfy multiple constraints and is effective for safely operating robots in real-world environments. To verify the proposed method, we used a simulation environment for walking robots and experimentally demonstrated that the trajectory generation can satisfy multiple constraints. As illustrated in Fig.1, for future work, we plan to develop an end-to-end motion generation framework using a large multimodal model such as CLIP[12] to infer the relevant constraints and their importance from images or text inputs.

Acknowledgments

This research was conducted during a 2024 summer internship at the Human Informatics Laboratories of Nippon Telegraph and Telephone Corporation (NTT) Yokosuka Research Center, Japan.

References

[1] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.*, 33:6840–6851, 2020.

[2] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, Dec. 2023.

[3] M. Janner, Y. Du, J. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pages 9902–9915. PMLR, June 2022.

[4] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, Mar. 2023.

[5] Q. Lin, B. Tang, Z. Wu, C. Yu, S. Mao, Q. Xie, X. Wang, and D. Wang. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning*, pages 21127–21152. PMLR, July 2023.

[6] Y. Zheng, J. Li, D. Yu, Y. Yang, S. E. Li, X. Zhan, and J. Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. In *The Twelfth International Conference on Learning Representations*, Oct. 2023.

[7] R. Römer, L. Brunke, M. Schuck, and A. P. Schoellig. Safe offline reinforcement learning using trajectory-level diffusion models. In *ICRA 2024 Workshop{—}Back to the Future: Robot Learning Going Probabilistic*, Apr. 2024.

[8] X. Ma, S. Patidar, I. Haughton, and S. James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. *arXiv [cs.RO]*, Mar. 2024.

[9] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, Oct. 2020.

[10] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, July 2019. ACM.

[11] Z. Liu, Z. Guo, Z. Cen, H. Zhang, J. Tan, B. Li, and D. Zhao. On the robustness of safe reinforcement learning under observational perturbations. In *The Eleventh International Conference on Learning Representations*, Sept. 2022.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, July 2021.