User Command Correction for Safe Remote Manipulation in Dynamic Environments

Mincheul Kang[1], Minsung Yoon[1] and Sung-Eui Yoon[2]

***Abstract*— Collision avoidance is an important factor for safe robot movement. In remote manipulation, a user's role is huge in avoiding obstacles because a robot follows a user's command. Especially, dynamic obstacle avoidance requires more user judgment. Unfortunately, a human sometimes decides an unrealizable command with the possibility of collision with obstacles. To cope with the unrealizable command, we present a learning-based user command correction method. Our method predicts the risk of dynamic obstacles and corrects a user's unrealizable command to avoid collision risk with dynamic obstacles. In this paper, we define the problem to be solved and introduce the proposed method briefly.**

I. INTRODUCTION

Remote manipulation has been used to perform dangerous and sophisticated tasks, such as works in nuclear power plants [1] and telesurgery [2], instead of humans. Recently, it has been expanding the scope to environments around us. For example, Telexistence[1] has been researching for remote manipulation to reduce the workforce going to convenience stores far away. In environments around us like a convenience store, there are various and dynamic obstacles. Therefore, handling such obstacles in the progress of manipulation planning is important.

Motion planning essentially considers avoiding collision with obstacles because it is an important factor in terms of safety. Although most motion planners can compute a collision-free trajectory for static obstacles, there is no way to guarantee collision avoidance for dynamic obstacles [3]. Therefore, dynamic obstacle avoidance is a difficult problem and important for safe robot movement.

Recent inverse kinematics (IK) methods have been proposed for remote manipulation with handling dynamic obstacles. CollisionIK [4] quickly computes the distance between robot links and dynamic obstacles by applying a fast convex shape representation method. The computed distances are applied to their optimizer to solve the IK problem for remote manipulation. RCIK [5] is a sampling-based approach that generates IK candidates and then selects one IK candidate away from obstacles. This method utilizes deep learning to quickly compute collision costs of IK candidates from sensor data in real-time.

Even though these methods consider avoiding dynamic obstacles, they depend more on a user's judgment as it has

[1]Mincheul Kang (mincheul.kang@kaist.ac.kr), [1]Minsung Yoon (minsung.yoon@kaist.ac.kr) are with the School of Computing, and [2]Sung-Eui Yoon (Corresponding author, sungeui@kaist.edu) is with the Faculty of School of Computing, KAIST at Daejeon, Korea 34141

[1]https://tx-inc.com

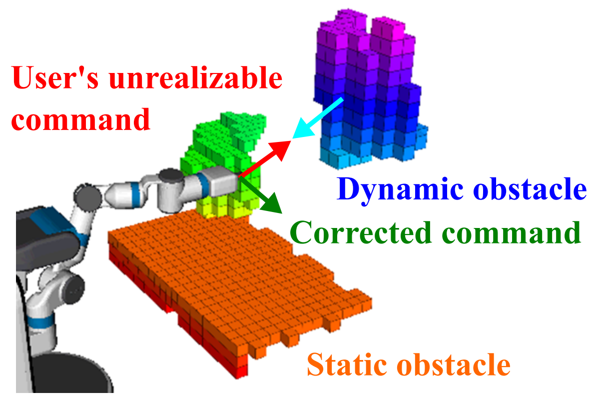

Fig. 1. Our goal is to correct a user's unrealizable command with a collision risk of dynamic obstacles to moving to a safe region.

to follow a user's command. In particular, sampling-based IK methods like RCIK are more affected because they aim to follow a user's commands exactly. In the case of static obstacles, even if a user's command is wrong, an IK solver considering collision avoidance can stop a robot to ensure safety. On the other hand, for dynamic obstacles, simply stopping is not enough. Since a human does not always decide a perfect command, we need a way to correct a user's unrealizable command that has the risk of collision with dynamic obstacles.

In this paper, we present a learning-based user command correction method in dynamic environments for safe remote manipulation (Fig. 1). Our method consists of two networks, a risk prediction network (RPN) for dynamic obstacles and a command correction network (CCN). The RPN predicts the risk of dynamic obstacles from successive occupancy grids and joint positions, and the CCN generates a command to move away from the obstacle. According to the predicted risk, we determine whether to follow the user's command or the CCN's command.

II. APPROACH

In this section, we introduce a problem definition and brief our approach.

A. Problem definition

In our problem, we deal with remote manipulation where a user consecutively gives a command, $x_u \subset \mathbb{R}^6$, to the end-effector in Cartesian space. For the command, a real-time IK solver, such as RCIK, synthesizes a joint configuration, q. We also target a redundant manipulator with multiple joint

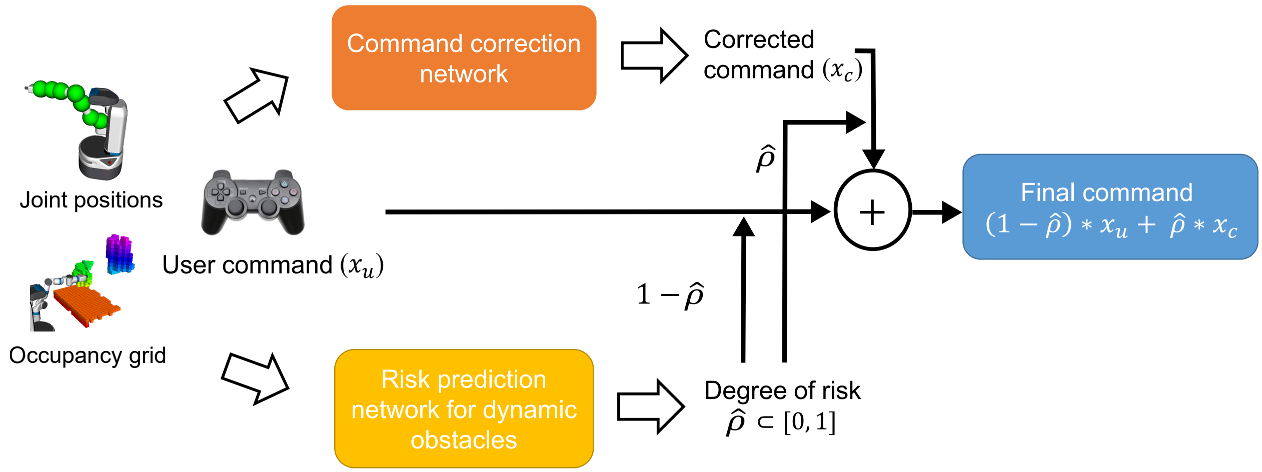

Fig. 2. This shows the system flow of our approach.

configurations for one end-effector pose due to having greater than six degrees of freedom (DoF).

Our goal is to perform safe remote manipulation in environments with dynamic obstacles. An IK solver considering collision avoidance can ensure safety for static obstacles by stopping a robot but cannot guarantee safety for dynamic obstacles. Even though a robot can be moved to a safe place through the user's judgment, a human can make mistakes. Therefore, we need a way to protect a robot from an unrealizable command causing a collision with obstacles.

In this work, we aim to correct unrealizable user commands by predicting the risk of collision with dynamic obstacles.

B. *User command correction*

Since the main purpose of remote manipulation is for a robot to follow a user's command, the command should be corrected only when it is judged to be unrealizable. We judge that the case of having the risk of collision with dynamic obstacles is highly likely to have unrealizable commands. Hence, we predict the risk of dynamic obstacles and correct an unrealizable command to move to a safe region. To do that, we use two kinds of networks. One is a risk prediction network (RPN) for dynamic obstacles, and the other is a command correction network (CCN).

Fig. 2 shows our system flow. The RPN predicts the degree of risk, $\hat{\rho} \subset [0,1]$, for dynamic obstacles from consecutive occupancy grids and joint positions. The CCN generates a corrected command, $x_c \subset \mathbb{R}^6$, to move away from the obstacle by taking the current environment and robot state into account. Finally, we decide the final command, x_f, between a user command x_u and a corrected command x_c according to $\hat{\rho}$: $x_f = (1-\hat{\rho}) * x_u + \hat{\rho} * x_c$. In short, when $\hat{\rho}$ is high, x_f is close to x_c, and when $\hat{\rho}$ is low, x_f is close to x_u. The final command x_f is delivered to an IK solver considering collision avoidance to synthesize a joint configuration; we use RCIK [5] as the IK solver.

III. CONCLUSION

In this paper, we introduced a user command correction method for safe remote manipulation in dynamic environments. We presented two networks, the risk prediction network for dynamic obstacles and the command correction network. Based on the predicted risk, our method decides the final command whether to follow a user's command or a corrected command. In the future, we would like to advance the proposed methods and prove their robustness through various experiments.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. 2021R1A4A1032582).

REFERENCES

[1] Shinji Kawatsuma, Mineo Fukushima, and Takashi Okada, "Emergency response by robots to fukushima-daiichi accident: summary and lessons learned", *Industrial Robot: An International Journal*, 2012.

[2] Cai Meng, Tianmiao Wang, Wusheng Chou, Sheng Luan, Yuru Zhang, and Zengmin Tian, "Remote surgery case: robot-assisted teleneurosurgery", in *ICRA*. IEEE, 2004, vol. 1, pp. 819–823.

[3] Jakob Thumm and Matthias Althoff, "Provably safe deep reinforcement learning for robotic manipulation in human environments", in *ICRA*. IEEE, 2022.

[4] Daniel Rakita, Haochen Shi, Bilge Mutlu, and Michael Gleicher, "Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance", in *ICRA*. IEEE, 2021.

[5] Mincheul Kang, Yoonki Cho, and Sung-Eui Yoon, "RCIK: Real-time collision-free inverse kinematics using a collision-cost prediction network", *RA-L*, vol. 7, no. 1, pp. 610–617, 2021.