# Learning User Preference Constraints from Demonstration

Che-Sang Park 2020-25752

## I. Proposal

USAGE of modern robots are expanding from structured industries to more various types of environments. Common robots before are mainly designed to be used in industrial field, isolated with the humans. However, robots these days are developed to share space with the humans and 'Cobot' and 'Domestic robots' are one of them. Cobots are robots working with the labors in the field, and Domestic robots are robots that does the housework instead. Such like these robots, the robots are evloving to share the domain with the human being.

As the domain of robot gets more complicated, we should consider various types of constraints and the first one is the occupancy space. Occupancy space is a space where the robot can exist. In complex environments, there are risks of colliding with obstacles and humans, so setting this space is important. Also, this space should not only consider currently existing obstacles but also obstacles that could appear in the future, which cannot be detected by the sensor. Similar work were done to notify the user about danger zone based on current state of the robot and its task[1]. However, this work has limitations since it is the user that has to avoid, which makes user passive.

The next constraint should be considered is privacy. If the robots work inside domestic environments, it might sometimes watch places where the user think as private. Therefore we should restrict the spaces where the robot can see. This problem has been discussed in mobile robot [2] [3] and it needs to be handled in domestic robots too. However, privacy is highly subjective property so robot could not detemine it alone. Therefore the user should set private spaces by him or herself. In this literature, we will call spaces that robot could see as 'Privacy-free space'.

Explicitly entering values to set these constraints could be the most direct method, however it is not user friendly way. Using dedicated software like UR robot sofware or ROS to set constraints requires deep understanding of the software and robotics, which is a high burden for the users. Also, every time the environment changes, the user should change the constraints all over again which is very time consuming. Therefore we need a way to program this constraints easily and intuitively.

We propose programming by demonstration framework to enable for users to set user preference constraints. Programming by demonstration[4] is a framework which user programming the robot by demonstration. This framework has advantage of easy programming by only moving the robot. However this frame work only focused on learning the motion frome the demonstration and not the constraints. Our work will focus on learning occupancy space and privacy-free space from the demonstration.

The occupancy space and privacy-free space are defined by space where the robot and vision cone sweep by the demonstration. After determining these two spaces, the robot should always satisfy following two conditions during the control. First, the robot should always be inside the occupancy space, and second, the vision cone should always be inside the privacy-free space.

To express these two space, we use Zonotope. Zonotope is defined as a Minkowski sum of set of lines called 'Generators' [5]. It is advantageous for expressing space sweeped by specific polytope, and detemining collision between two zonotopes. However, directly applying zonotopes has several difficulties. The zonotope could only express convex spaces, making it hard to express occupancy and privacy-free space which could be non-convex. Also, our purpose is to determine the inclusive relationship between occupancy, privacy-free space and robot, vision cone respectively and this is not a collision detecting problem. Therefore we cannot apply normal collision determining algorithm of zonotope.

To solve this difficulties, we change the problem into expressing complement of the space with union of zonotopes. Zonotope alone could only express convex space, but with union of zonotopes, non-convex space could be expressed. Also, zonotope completely inclusive to bigger zonotope means it doesnot collide with the complement of the bigger zonotope. Therefore if we could apply collision detection algorithm on the complement space of big zonotope to determine whether small zonotope is completely inclusive to the bigger one.

## References

[1] Antti Hietanen, Roel Pieters, Minna Lanz, Jyrki Latokartano, and Joni-Kristian Kämäräinen. Ar-based interaction for human-robot collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing*, 63:101891, 2020.

[2] Myeung Un Kim, Harim Lee, Hyun Jong Yang, and Michael S. Ryoo. Privacy-preserving robot vision with anonymized faces by extreme low resolution. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 462–467, 2019.

[3] Ying Zhang, Cui-Hua Zhang, and Xuyang Shao. User preference-aware navigation for mobile robot in domestic via defined virtual area. *Journal of Network and Computer Applications*, 173:102885, 2021.

[4] Eric M. Orendt, Myriel Fichtner, and Dominik Henrich. Robot programming by non-experts: Intuitiveness and robustness of one-shot robot programming. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 192–199, 2016.

[5] Trevor J. Bird and Neera Jain. Unions and complements of hybrid zonotopes. *IEEE Control Systems Letters*, 6:1778–1783, 2022.