Learning Perceptual Hallucination for Multi-Robot Navigation in Narrow Hallways

Jin-Soo Park[1], Xuesu Xiao[2], Garrett Warnell[3], Harel Yedidsion[2], and Peter Stone[2,4]

Abstract— **While current systems for autonomous robot navigation can produce safe and efficient motion plans in static environments, they usually generate suboptimal behaviors when multiple robots must navigate together in confined spaces. For example, when two robots meet each other in a narrow hallway, they may either turn around to find an alternative route or collide with each other. This paper presents a new approach to navigation that allows two robots to perform hallway passing without colliding, stopping, or waiting. Our approach, Perceptual Hallucination for Hallway Passing (PHHP), learns to synthetically generate virtual obstacles (i.e., *perceptual hallucination*) to facilitate narrow hallway passing for multiple robots that utilize otherwise standard autonomous navigation systems. Our experiments on physical robots in different hallways show improved performance compared to multiple baselines.**

I. INTRODUCTION

One of the grand goals of the robotics community is to safely and reliably deploy fully-autonomous mobile robots in common environments over extended periods of time. Indeed, many researchers have moved toward this vision and reported hundreds of hours of unsupervised, collision-free navigation of a single robot [1], [2].

However, long-term deployment of *multiple* autonomous robots in common spaces still remains a difficult task. One reason for this difficulty is that conventional navigation systems are good at handling static environments, but their performance deteriorates in the presence of dynamic obstacles, e.g., other moving robots. While some solutions to this problem have been explored in the community, they typically come with unrealistic requirements such as a perfectly-controlled space (e.g., a warehouse) or perfect sensing, and cannot guarantee safety in novel environments without time-consuming movement schemes such as one robot halting while another moves past. To the best of our knowledge, there are no reports that claim long-term deployment of *multiple* autonomous robots in uncontrolled spaces without human supervision.

Separately, recent work in the navigation community leveraging the concept of *perceptual hallucination* [3]–[5] has demonstrated impressive results in allowing robots to navigate highly constrained spaces successfully. Perceptual hallucination refers to the technique of introducing virtual obstacles to the robot's perception so that a motion plan executed with these obstacles exhibits certain desired behaviors and also acts as a blindfold for the sensor to conceal unnecessary (or even distracting) information. To date, however, perceptual hallucination has not been applied in the context of multiple robots or dynamic obstacles.

In this paper, we hypothesize that perceptual hallucination can be used to improve conventional navigation systems in multi-robot and confined settings. In particular, we posit that, by using hallucination to obscure the presence of moving objects that would otherwise result in suboptimal behavior (e.g., turning around), we can enable multi-robot navigation in confined spaces such as a narrow hallway. By using perceptual hallucination, we can still solve the multi-robot navigation problem using conventional navigation systems that have been thoroughly tested to be stable in static environments.

To investigate this hypothesis, we introduce and evaluate *Perceptual Hallucination for Hallway Passing* (PHHP), a hallucination-based approach to improve a given navigation policy in the setting of multi-robot navigation in narrow hallways. PHHP uses experience gathered in domain-randomized simulation episodes in order to learn the proper size and placement of virtual obstacles so as to enable successful navigation. We investigate the performance and robustness of using PHHP in a narrow hallway with both simulation and real-world experiments, and we find that it can achieve similar performance compared to a widely known method, Optimal Reciprocal Collision Avoidance (ORCA) [6], while avoiding its assumption of perfect sensing. We further show that, compared to a rule-based right-lane-following method, PHHP reduces the average delay by 58.54%. Finally, we show that PHHP is sufficiently robust against the sim-to-real gap, different speeds, detection ranges, hallway widths, and even different underlying navigation systems.

II. RELATED WORK

One of the ultimate goals in the autonomous robotic navigation community is the long-term deployment of mobile robots in shared spaces without human supervision. Khandelwal et al. [2] and Biswas et al. [1] reported that their robots successfully navigated hundreds of hours with conventional navigation systems [7], [8].

Despite the successes of the conventional navigation systems for single robots, their degraded performance in the presence of the other robots inhibits the safe deployment of multiple robots in confined spaces. One solution is to use a centralized control system. For example, Kiva Systems (now Amazon Robotics) shows a very large-scale deployment of warehouse robots [9] with centralized control and

[1]Department of Electrical and Computer Engineering, [2]Department of Computer Science, The University of Texas at Austin, Austin, TX 78712 [3]Computational and Information Sciences Directorate, Army Research Laboratory [4]Sony AI js.park@utexas.edu, garrett.a.warnell.civ@army.mil, {xiao, pstone}@cs.utexas.edu

fixed lanes. Similarly, Jiang et al. [10] introduced Iterative Inter-Dependent Planning to avoid potential conflict at the planner level. However, the centralized method will not apply anymore when the density of robots in the shared space increases or objects that do not follow centralized control, e.g., people, exists.

Hence, there are attempts to replace conventional navigation systems with new motion planners focused on multi-robot applications. The family of velocity obstacle algorithms tries to find optimal steering commands by considering motion in the velocity space. Fiorini et al. [11] first introduce the concept of reciprocal collision avoidance and Van den Berg et al. [6] present the ORCA algorithm which finds an optimal reciprocal collision avoidance behavior. However, this family of velocity obstacle approaches heavily relies on perfect sensing and holonomic systems. There is some research that tries to relax these conditions [12], [13], but it is still sensitive to sensor noise and requires heavy parameter tuning for every new environment.

Inspired by the recent successes of machine learning, there exists a trend to solve the multi-robot navigation problem with neural networks. Long et al. [14] show that multilayer perceptrons can successfully map motion commands from noisy sensor readings to mimic the behavior of ORCA. Long et al. [15] further improve their method with reinforcement learning. Lin et al. [16] present how to learn swarm behavior with centralized learning and decentralized execution. Tan et al. [17] incorporate synchronized location map of other agents to further improve the performance. However, these neural-network-based policies are vulnerable to unseen environments and often suffer from a lack of safety guarantees, e.g., they do not prevent collisions in highly constrained narrow hallways.

There have been some attempts to combine the advantages of conventional navigation systems and machine learning. Fan et al. [18] develop a switching policy that uses PID control in simple scenarios, reinforcement policy in complex ones, and a safe policy in dangerous ones. They successfully eliminate unsafe behaviors of the reinforcement-based policy, but the greedy behavior makes it hard to deploy in complex indoor environments. Park et al. [19] developed a method that finds a rest spot on the fly right after a robot detects the presence of another robot. The halting robot will park at the rest spot until the other passes and then resume traveling to its goal. This halting behavior is very time-consuming and hard to extend to multiple robots.

On the other hand, the concept of hallucination has emerged to address navigation in highly-constrained spaces. Xiao et al. [3] introduced the concept of learning from hallucination (LfH). They record random maneuvers at an open space and convert them into agile maneuvers in the most cluttered environment by adding hallucinated obstacles. They then trained the policy to learn agile maneuvers in cluttered environments with that data. Xiao et al. [4] and Wang et al. [5] further improved this idea by finding a minimal hallucination set and learning to generate obstacle configurations from a given robot's trajectory. Despite these successes, the idea of hallucination has not previously been applied to dynamic scenarios, including multiple robots.

III. APPROACH

In this section, we first formulate the multi-robot hallway passing problem. We then describe our solution, Perceptual Hallucination for Hallway Passing (PHHP).

A. Problem Formulation

We consider here the specific scenario in which two robots moving in opposite directions must pass each other in a hallway that is narrow, but also wide enough to allow the two robots to simultaneously pass each other. In this scenario, let $\vec{p}^1$ and $\vec{p}^2$ denote the two-dimensional positions of the first and second robot, and let $\vec{c}$ denote the position of the center of the hallway, respectively. Additionally, we assume that each robot is equipped with a two-dimensional LiDAR scanner, and the LiDAR measurements obtained by each robot at time t are denoted as $\mathbf{l}_t^1$ and $\mathbf{l}_t^2$. Finally, we assume that both robots are using an existing autonomous navigation system (e.g., ROS `move_base` [20]).

In this paper, we seek to investigate whether perceptual hallucination can improve (i.e., reduce collisions and increase the speed and smoothness of navigation) the existing navigation system in the scenario described above. Mathematically, we use h to denote the *hallucination function*, i.e., the sensor reading $\mathbf{l}_{\mathcal{H}} = h(\mathbf{l}, \mathcal{H})$ is modified by transforming a LiDAR scan $\mathbf{l}$ such that it appears as if virtual obstacles specified an *obstacle field* $\mathcal{H}$ were added to the current environment. Importantly, $\mathbf{l}_{\mathcal{H}}$ only contains *additional* obstacles, i.e., to compute the depth value at any particular bearing, k, the *minimum* value between the real scan, l^k, and a virtual scan corresponding to only obstacles in $\mathcal{H}$, $v_{\mathcal{H}}^k$, is chosen. Additionally, we assume that positional information about each obstacle in $\mathcal{H}$ is specified relative to the pose of the particular robot that is hallucinating.

In order to use perceptual hallucination for our hallway passing problem, we must determine what each robot should use for its hallucinated obstacle field $\mathcal{H}$ (i.e., the shape and location of hallucinated obstacles) to enable smooth interaction. In general, each $\mathcal{H}$ could consist of an arbitrary number of obstacles, each with arbitrary shape. However, in order to make this problem tractable, we consider here only obstacle fields that contain a single object, where that object is a rectangle with rounded sides. We denote such obstacle fields as $\mathcal{H}_{\boldsymbol{\theta}}$, where the parameter $\boldsymbol{\theta} = (r, l, dx, dy)$ has components representing the radius, length, and the x-y coordinate of the center of the rounded rectangle relative to the hallway center $\vec{c}$, respectively. We found that a rectangle with rounded corners provides two advantages for the hallway passing problem considered here: *(1)* the rounded sides prevent the motion planner from generating sharp turning trajectories near the boundaries of the obstacle, and *(2)* the flat sides lead to stable lane-keeping-like behavior. Our method requires that each robot use the *same* $\mathcal{H}_{\boldsymbol{\theta}}$. However, note that, since each robot has a different starting pose (in particular, facing each other), each robot will still hallucinate

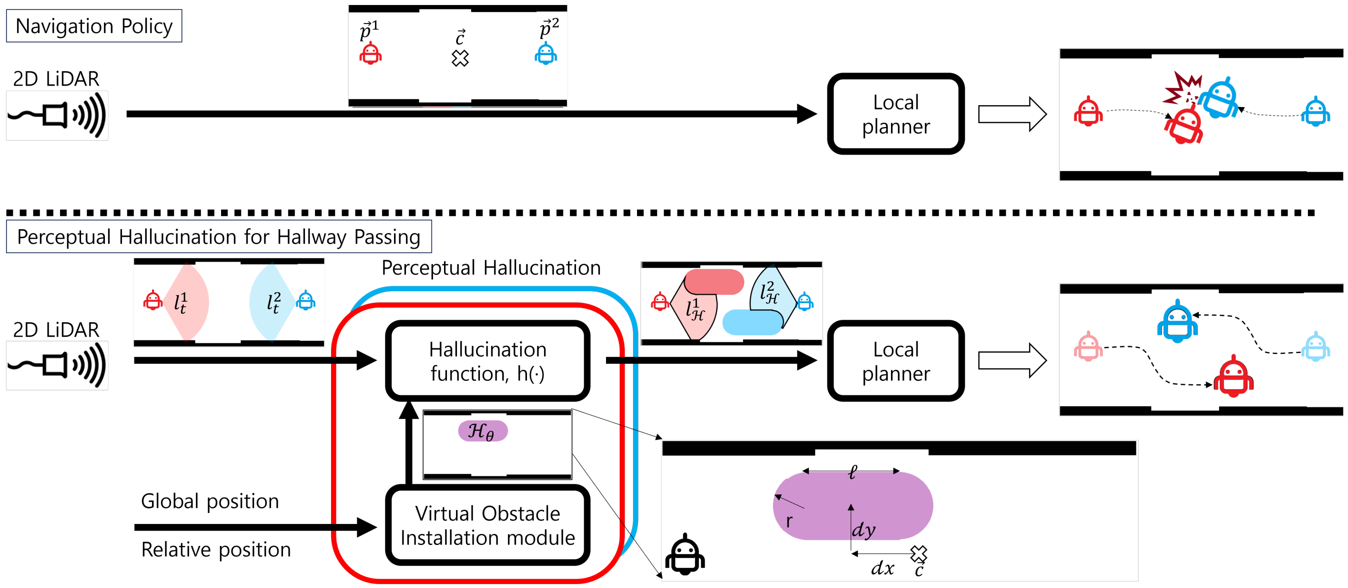

Fig. 1: Overview of Perceptual Hallucination for Hallway Passing: (top) multi-robot hallway passing scenario with an existing navigation system, (bottom) how PHHP improves the navigation system with hallucinated sensor readings. The color represents the ID of each robot. When one robot detects the other, the virtual obstacle installation module generates a virtual obstacle, $\mathcal{H}_{\boldsymbol{\theta}}$, in the global coordinate system defined with respect to the the center of the hallway and orientation of the robot. How $\boldsymbol{\theta}$ specifies the obstacle shape here is shown in the bottom right. The hallucination function, $h(\cdot)$ computes the depth value of hallucinated readings by taking the *minimum* value between the real scan and the virtual scan. The robots use their existing navigation system with hallucinated sensor readings $\mathbf{l}_{\mathcal{H}}$ to pass each other in the narrow hallway.

its own unique obstacle placed at a different location in the environment. An overview of the hallway passing scenario and how perceptual hallucination is applied is illustrated in Figure 1.

In order to understand which $\mathcal{H}_{\boldsymbol{\theta}}$ is best for hallway passing, we define a hallway-passing cost function. For a given hallway passing scenario, we define this cost to encourage both fast and safe passing, i.e.,

$$C(\mathcal{H}_{\boldsymbol{\theta}}) = \frac{\mathrm{TTD}_1(\mathcal{H}_\theta) + \mathrm{TTD}_2(\mathcal{H}_\theta)}{2} + c_{coll} * \mathcal{I}_{coll}(\mathcal{H}_\theta) \ , \quad (1)$$

where $\mathrm{TTD}_i(\mathcal{H}_\theta)$ denotes the amount of time (seconds) it takes for robot i to reach their goal with virtual obstaccle $\mathcal{H}_\theta$, $\mathcal{I}_{coll}(\mathcal{H}_\theta)$ is an indicator function that is 1 if a collision occurred (and 0 otherwise), and we set the collision penalty c_{coll} to 100. With this setup, the problem of finding the best obstacle to hallucinate for the hallway passing problem becomes one of finding the $\boldsymbol{\theta}$ that minimizes this cost, i.e.,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} C(\mathcal{H}_{\boldsymbol{\theta}}) \ . \quad (2)$$

B. Optimal Hallucination

We solve Equation (2) and find $\boldsymbol{\theta}^*$ using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [21] algorithm, a population-based, black-box optimization that selects and evaluates successive generations of samples. In each generation, the mean of the next distribution is calculated by the weighted sum of samples ($\boldsymbol{\Theta}$) where the sample with the lower result has a larger weight. The new samples are collected from the normal distribution with measured mean and covariance and tested until the search distribution become small enough. At this point, the minimum-cost sample across all generations is returned as $\boldsymbol{\theta}^*$.

For each sample $\boldsymbol{\theta}$, we measure the value of $C(\mathcal{H}_{\boldsymbol{\theta}})$ by executing an hallway passing episode with perceptual hallucination in simulation. More specifically, the episode is initialized such that the two robots face one another, and each is given a navigation goal corresponding to the starting position of the other robot. Then each robot begins navigating using its existing navigation system (here, the ROS navigation stack). The robots detect one another when they are within a certain range and, when this happens, they employ hallucination (i.e., they begins using $\mathbf{l}_{\mathcal{H}_{\boldsymbol{\theta}},t}$ as the LiDAR readings supplied to the navigation system). The episode ends when both robots get sufficiently close to their respective goal locations. The amount of time it takes each robot is recorded as TTD_i. Collision is defined as any contact between robots or robot and walls.

In order to ensure $\boldsymbol{\theta}^*$ is robust to differences between conditions in simulation and those in the real world, we further employ *domain randomization* [22], [23]. That is, we compute the CMA-ES objective for each sample by averaging costs obtained over several simulation episodes, each with randomized starting delay t_i and detection range D_i.

The pseudocode of the *perceptual hallucination for hallway passing* (PHHP) is given in Algorithm 1.

IV. EXPERIMENTS

We now seek to characterize the efficacy of Perceptual Hallucination for Hallway Passing (PHHP). In particular, we

Algorithm 1 Find optimal Hallucination with CMA-ES

Require: r_0, l_0, x_0, y_0
  CMAES.initialize(r_0, l_0, x_0, y_0)
  min_cost $\leftarrow \infty$
  $\boldsymbol{\theta}^* \leftarrow$ None
  while $\sigma \geq threshold$ **do**
    $\boldsymbol{\Theta} \leftarrow$ CMAES.generate_samples()
    for k $\leftarrow 1$ to N **do**
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\Theta}[k]$
      $t_1, t_2 \leftarrow \mathcal{U}_{[0,t_{max}]}, \mathcal{U}_{[0,t_{max}]}$
      $D_1, D_2 \leftarrow \mathcal{U}_{[d_{min},d_{max}]}, \mathcal{U}_{[d_{min},d_{max}]}$
      TTD$_1$, TTD$_2$, $\mathcal{I}_{coll} \leftarrow$ episode($\boldsymbol{\theta}$, t_1, D_1, t_2, D_2)
      cost[k] $\leftarrow \frac{\text{TTD}_1+\text{TTD}_2}{2} + 100 \cdot \mathcal{I}_{coll}$
      if min_cost $\geq$ min(cost) **then**
        min_cost $\leftarrow$ min(cost)
        $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
      end if
    end for
    CMAES.optimize(samples, costs)
    $\sigma \leftarrow$ CMAES.evaluate()
  end while
  return $\boldsymbol{\theta}^*$

evaluate: *(1)* the amount of delay PHHP incurs during the multi-robot hallway passing scenario compared to a single robot in the equivalent episode, *(2)* the extent to which PHHP suffers from collisions or planning failures, and (3) whether PHHP is robust enough to generalize to various environments in simulation and real-world deployment.

We compare PHHP to three alternative methods: a rule-based, right-lane-following baseline; ORCA [6]; and our prior approach, which we refer to here as the *halting method* [19]. [1] The explanation of each method is in Sec. IV-C.

We evaluate the performance of each method using the following metrics:

- Δt: The amount of delay compared to a single robot traversing the same hallway.
- $P_{collision}$: The probability of collision.
- $P_{failure}$: The probability that the navigation system fails to generate a plan, which typically manifests as the robot turning around.

Δt, or delay, is measured by the amount of TTD incurred in the episode compared to the time it takes for a single robot to travel between exactly the same start and end points without the presence of the other robot.

A. Platform

We evaluate PHHP using BWIBots [24], a custom differential-drive robot atop a Segway base. A single BWIBot is 65cm wide, and has a maximum linear velocity of 1.0 m/s. The BWIBot is equipped with a front-facing Hokuyo LiDAR sensor with a 170-degree field of view and a maximum range of 20m. For the embedded navigation system, the BWIBot uses the E-Band planner [7] as the local planner, which continually generates a sequence of motion commands over a 4m horizon.

[1] The method was originally called the “adaptive method” in the paper.

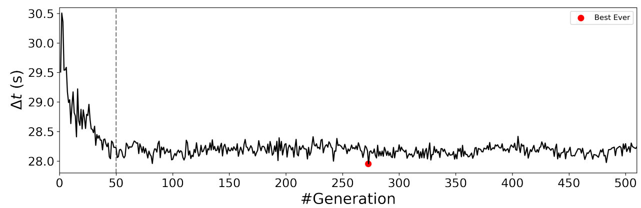

Fig. 2: The optimization curve of CMA-ES in simulation. The system finds approximated solutions after only 50 generations. The best configuration is marked as a red dot.

B. Training

We train PHHP using the widely-used Gazebo [25] simulator since it provides safe and fast ways to collect realistic data. The simulated training hallway is 1.6m wide, where the two BWIBots can barely pass each other. Training episodes proceed as described in Section III-B, where the robots spawn at either end of the hallway, 17m apart from one another. For domain randomization, we sample starting delays uniformly over the interval $[0, 5]$s, and detection ranges uniformly over the interval $[6, 10]$m for each episode.

We use PHHP to find the optimal virtual obstacle for four different hallways ranging from 1.6m to 2.8m in 0.4m intervals. To accelerate the CMA-ES search, we used $(r, l, dx, dy) = (1.0, 2.0, 1.0, 1.0)$ as an initial hypothesis, which, intuitively, represents a virtual obstacle that entirely blocks the left half of the hallway from the robot's perspective. For each run of CMA-ES, a total of approximately 500 generations occur before the standard deviation of all samples in a generation becomes less than our selected threshold of 0.001. A single generation contains 8 sample configurations, and each configuration is evaluated by the average cost in Eq. 1 averaged over 16 domain-randomized episodes. The learning curve of CMA-ES in the narrowest hallway (1.6m wide) is shown in Figure 2, where it can be seen that an approximate solution is found after 50 generations. The optimized configurations of virtual obstacle in all four hallways are presented in Table I.

TABLE I: Optimal configuration of virtual obstacle in various hallways.

width	radius	length	dx	dy
1.6	0.437	1.739	0.8595	0.491
2.0	1.650	0.041	0.254	1.998
2.4	4.804	0.378	0.323	5.181
2.8	5.317	1.257	0.4475	5.819

C. Alternative Approaches

We compare PHHP with three alternative methods; a right-lane-following baseline, ORCA, and the halting method. The right-lane-following baseline, or simply baseline, is inspired by the US traffic standard. It is a rule-based algorithm that, upon detection of the other robot, moves the robot

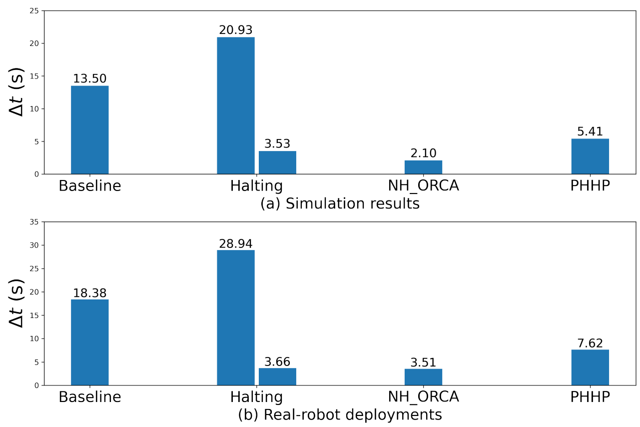

Fig. 3: The average delay of each policy at a 1.6m-wide hallway in (a) simulation and (b) actual deployment. The left and right bar of the Halting method represents the performance of the halting / non-halting robot, respectively. Note that NH_ORCA results in the smallest delay, but it also records 71.4% of the collision rate in the real-world deployment.

into a human-annotated right lane and proceeds in that lane there until the two robots pass one another. ORCA uses the velocity-obstacle field to indicate whether the goal-directed velocity is safe to execute. If not, it finds maximum possible collision avoidable velocity assuming that the opponent will do the same. While ORCA provides excellent performance (we consider it to be an upper bound), it also requires complete knowledge of surroundings—including the precise position and velocity of the other robot–which limits the situations in which it can be applied. In the simulation, this information is easily accessible, but robots have to use communication to share their location and velocity in the real-world experiment. Hence, whenever the channel becomes noisy, ORCA have a risk of collision. This contrasts with PHHP, which only needs to observe the presence of the other robot once. Finally, the halting method is a system designed for hallway passing in which, when a halting robot detects a potential collision, that halting robot immediately moves to the nearest, safe parking spot until the non-halting robot completely passes then resumes. It is a general approach but, due to the halting behavior, the average delay is high.

D. Comparison vs. Alternative Approaches

First, We evaluate the performance of PHHP by comparing it with the baseline, ORCA, and halting methods. The experiments are conducted in simulations and the real world. The result is shown in Fig. 3.

Fig. 3 shows the performance of each method in the most constrained hallway (1.6m wide). Notice first that the baseline requires a fairly sizeable average delay of 13.5 / 18.4 seconds, which arises because the E-Band local planner used in this experiment attempts to stay away from walls, causing the robot to use two slow, right-angle turns to enter the lane. In contrast, PHHP uses virtual obstacle with a learned shape and placement that more gradually narrows the hallway, which allows the E-Band planner to find smoother trajectories, and ultimately reduces the average delay by 59.93 / 58.54 % relative to the baseline.

The two bars corresponding to the halting method represent the average Δt for the halting robot (left) and the non-halting robot (right). Notice that, while the halting method results in relatively little delay for the non-halting robot, it results in a large delay time for the halting robot. PHHP, on the other hand, does not require either robot to halt, resulting in an overall delay of only about 2-4 seconds more than the non-halting robot.

Finally, the performance and safety of PHHP are compared to the optimal policy ORCA in the hallway passing scenario. While ORCA represents a performance upper bound, it also assumes that it has access to each robot's true position and velocity, which is a difficult assumption to satisfy in a real-world setting. Therefore, unlike the simulation, we observed 5 collisions among 7 experiments in the real robot deployment. We did not perform further experiments with ORCA in real robot deployment due to hard crashes. The major reason for this catastrophic failure is because of the noisy communication channel, not the algorithm itself, but still, ORCA is not applicable to real-robot deployment. PHHP, on the other hand, provides a solution with 3.3s to 4.1s additional delay while preserving perfect safety. One reason for the stability of PHHP is because it only needs to observe the presence of the other robot once when they are at a reasonable distance. (6-10m).

E. Robustness Analysis

We investigate the robustness of PHHP in terms of sim-to-real transfer, different environments, and different characteristics of the robot (i.e., different detection ranges, velocities, or even different base navigation systems). The test setup is as follows. In simulation, 1,000 experiments each are conducted in simulated hallways ranging in width from the most constrained (1.6m-wide) hallway to a relatively wide (2.8m-wide) hallway in 0.4m intervals. In the real world, we define several particular conditions, each with specific environment and robot parameters, and we run 30 continuous experiments per condition. Note that the speed of the robot in real-world deployments is limited to 0.75m/s for safety reasons unless specified. The robots report their locations to each other through wireless communication once they are within detection range. Also, there is about 50cm of uncertainty in the reported location due to communication delay and localization error. The detail of each real-robot deployment condition is given in Table II, and the result of each experiment is shown in Figure 4. ***Importantly, no collision or turnaround was observed during the entire set of simulation and real experiments***. The results indicate that PHHP is robust to overcome sim-to-real gap since we directly deploy a policy optimized in simulation without any changes.

Simulation results and `Wide_w` experiment in actual deployment indicate that PHHP trained on the most constrained

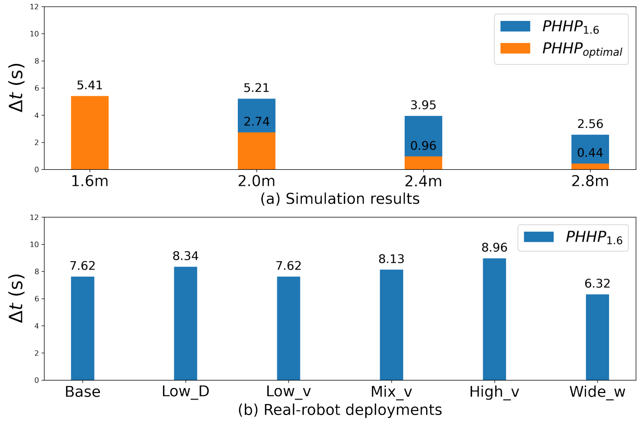

Fig. 4: The average delay of PHHP in (a) simulation and (b) real-world deployment. PHHP is tested in different widths of the hallway in simulation and the various conditions in the real-world experiment. The detailed condition of each experiment in (b) is provided in Table. II.

1.6m-wide hallway is remarkably robust to hallways of any wider widths. The robot with most constrained PHHP (Orange) successfully pass each other with only 2.1s to 3.0s additional delay in unseen environments compared to the optimal PHHP (Blue) trained on that environment. The optimal configuration of virtual obstacles in each simulated hallway is described in Table. I.

Additionally, the real-world results show that PHHP trained in simulation is robust to other factors such as detection range (D) and speed (v_i). Like the simulation results, ***no collision or turnaround was observed during real-world deployment***. We find it interesting that the average delay of PHHP increases as the velocity increases or the detection range decreases. If robots use faster linear velocity or a shorter detection range, the virtual obstacle created by PHHP appears at a closer distance. As a result, it incurs a sharp turning trajectory with more rotational movement and increases the average delay. Lastly, robustness to different control systems is studied. The experiment was conducted 100 times in simulation with two robots using E-band [7] and DWA [8] planners, respectively, in the 1.6m-wide hallway. The average delay of the robot using the E-Band planner is 5.91 seconds. This delay is similar to that of the homogeneous control system experiment. Within 100 episodes, no collisions or turnaround is observed. Therefore, PHHP is robust to the changes in detection range and velocity, and even base control systems show that PHHP can be applied to the robots with heterogenous control systems without any finetuning.

In this section, we reported a total of 7,100 simulation episodes along with 180 real robot episodes. The results confirmed that PHHP could operate robustly regardless of the sim-to-real problem, the hallway's width, different detection ranges, velocity, or even with a different base navigation system.

TABLE II: The configuration used in real world experiments.

name	Base	Low_D	Low_v	Mix_v	High_v	Wide_w
D	8.0	**6.5**	8.0	8.0	8.0	8.0
V_1	0.75	0.75	0.75	0.75	**1.2**	0.75
V_2	0.75	0.75	0.75	**1.2**	**1.2**	0.75
w	1.6	1.6	1.6	1.6	1.6	**1.8**

Fig. 5: The two hallways in which we performed the real-world PHHP experiments. The hallway shown in (left) is 1.6m in width, while the hallway shown in (right) is 1.8m wide.

V. CONCLUSIONS

In this paper, we presented Perceptual Hallucination for Hallway Passing (PHHP), a new method that enables multi-robot navigation in constrained spaces. We showed how to find the best obstacle for PHHP to hallucinate for a given environment and navigation policy using CMA-ES. The simulation and real-world deployment results indicate that PHHP achieves comparable performance against ORCA, while removing the assumption that the robot has continuous access to the other robots' position and velocity. Moreover, PHHP outperforms both a right-lane-following baseline and our prior work, the halting method, in terms of delay. Additionally, real-world deployment results experimentally confirm that PHHP, which is trained in simulation, can successfully be deployed in a wide variety of real-world settings, including those in which the size of the hallway has changed, the robots move with different velocities, their perception system exhibits a different detection range, or their underlying navigation system has changed. Despite the successes we presented, PHHP has only been developed and evaluated here in a two-robot, straight-hallway setting, and we have only explored using a single static obstacle in the hallucinated obstacle field. Therefore, an important direction for future work is to investigate how to expand PHHP to work with multiple, arbitrarily shaped obstacles in a wider variety of settings.

References

[1] J. Biswas and M. Veloso, “The 1,000-km challenge: Insights and quantitative and qualitative results,” *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 86–96, 2016.

[2] P. Khandelwal, S. Zhang, J. Sinapov, M. Leonetti, J. Thomason, F. Yang, I. Gori, M. Svetlik, P. Khante, V. Lifschitz, *et al.*, “Bwibots: A platform for bridging the gap between ai and human–robot interaction research,” *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 635–659, 2017.

[3] X. Xiao, B. Liu, G. Warnell, and P. Stone, “Toward agile maneuvers in highly constrained spaces: Learning from hallucination,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1503–1510, 2021.

[4] X. Xiao, B. Liu, and P. Stone, “Agile robot navigation through hallucinated learning and sober deployment,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[5] Z. Wang, X. Xiao, A. J. Nettekoven, K. Umasankar, A. Singh, S. Bommakanti, U. Topcu, and P. Stone, “From agile ground to aerial navigation: Learning from learned hallucination,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[6] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in *Robotics research*. Springer, 2011, pp. 3–19.

[7] S. Quinlan and O. Khatib, “Elastic bands: Connecting path planning and control,” in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 802–807.

[8] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[9] P. R. Wurman, R. D’Andrea, and M. Mountz, “Coordinating hundreds of cooperative, autonomous vehicles in warehouses,” *AI magazine*, vol. 29, no. 1, pp. 9–9, 2008.

[10] Y. Jiang, H. Yedidsion, S. Zhang, G. Sharon, and P. Stone, “Multi-robot planning with conflicts and synergies,” *Autonomous Robots*, March 2019.

[11] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[12] D. Hennes, D. Claes, W. Meeussen, and K. Tuyls, “Multi-robot collision avoidance with localization uncertainty.” in *AAMAS*, 2012, pp. 147–154.

[13] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, “Optimal reciprocal collision avoidance for multiple non-holonomic robots,” in *Distributed autonomous robotic systems*. Springer, 2013, pp. 203–216.

[14] P. Long, W. Liu, and J. Pan, “Deep-learned collision avoidance policy for distributed multiagent navigation,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 656–663, 2017.

[15] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, “Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6252–6259.

[16] J. Lin, X. Yang, P. Zheng, and H. Cheng, “End-to-end decentralized multi-robot navigation in unknown complex environments via deep reinforcement learning,” in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2019, pp. 2493–2500.

[17] Q. Tan, T. Fan, J. Pan, and D. Manocha, “Deepmnavigate: Deep reinforced multi-robot navigation unifying local & global collision avoidance,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6952–6959.

[18] T. Fan, P. Long, W. Liu, and J. Pan, “Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios,” *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[19] J.-S. Park, B. Tsang, H. Yedidsion, G. Warnell, D. Kyoung, and P. Stone, “Learning to improve multi-robot hallway navigation,” November 2020.

[20] OSRF, “Ros wiki move_base,” http://wiki.ros.org/move_base, 2018.

[21] N. Hansen, S. Muller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES).” *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[22] N. Jakobi, “Evolutionary robotics and the radical envelope-of-noise hypothesis,” *Adaptive Behavior*, vol. 6, no. 2, pp. 325–368, 1997.

[23] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[24] P. Khandelwal, S. Zhang, J. Sinapov, M. Leonetti, J. Thomason, F. Yang, I. Gori, M. Svetlik, P. Khante, V. Lifschitz, J. K. Aggarwal, R. Mooney, and P. Stone, “Bwibots: A platform for bridging the gap between ai and human–robot interaction research,” *The International Journal of Robotics Research*, 2017. [Online]. Available: http://www.cs.utexas.edu/users/ai-lab?khandelwal:ijrr17

[25] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.