2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)
October 23-27, 2022, Kyoto, Japan

Collaborative Teleoperation with Haptic Feedback for Collision-Free Navigation of Ground Robots

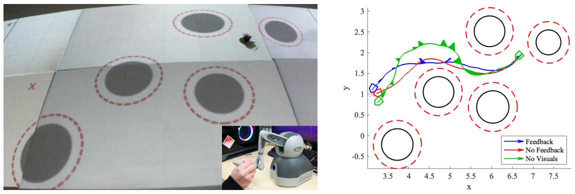
Mela Coffey and Alyssa Pierson

***Abstract*— We propose a collaborative teleoperation algorithm which utilizes haptic force feedback to guide users around oncoming obstacles while accounting for non-holonomic constraints. The proposed algorithm predicts the user's goal, plans a path using a modified RRT* algorithm to the predicted goal, and provides haptic guidance to the path and away from obstacles when the user is in an unsafe pose. We show that the vehicle cannot collide with obstacles under the proposed algorithm following the haptic commands. We assess the performance of our algorithm with a virtual pilot in simulations and hardware experiments, demonstrating its ability to prevent collisions while reaching the goal location. Additionally, we demonstrate human-in-the-loop navigation with a Geomagic Touch haptic device providing force feedback to the user. These simulations and experiments show that the proposed haptic guidance system is a useful and effective tool for co-navigation of non-holonomic vehicles via teleoperation.**

I. INTRODUCTION

Robots and automated technologies continue to integrate into our everyday life, from robotic vacuums to delivery drones. To date, most robots serve human needs by design, but we could accomplish so much more if we program them to work with us as teammates. Many robot control schemes do not truly collaborate with humans, nor do they take advantage of humans' sophisticated autonomy. We envision a collaborative personal mobility platform where a human user is responsible for high-level decision-making, while the mobility platform handles collision avoidance and route planning suggestions. Haptic feedback provides a natural communication mechanism with the human user, as haptic feedback gives interpretable cues about the autonomous system's intent while minimizing the additional cognitive load of the task. Such a mobility platform would be useful for personal transportation, such as autonomous wheelchairs, as well as service and transportation vehicles.

Our approach is also relevant to collaborative teleoperation and co-navigation tasks. Robotic teleoperation is a popular method for exploring remote environments, particularly when conditions are unsafe or impossible for human exploration. An inherent difficulty with teleoperation is the challenge of perception; classic teleoperation provides a narrow, two-dimensional field of view with no haptic feel of the environment [1], [2]. Shared autonomy can guarantee collision avoidance but limits the human's control of the robot [3]. These systems may augment known paths to reduce risk [4], predict the human intent for parallel autonomy [5], or give the user a choice between safe RRT*-generated paths [6].



(a) (b)

Fig. 1. (a) The Geomagic Touch, and experimental setup with the robot and projected environment. (b) Sample experimental results for a human operator, with arrows indicating force feedback magnitude and direction.

While shared autonomy reduces human control, a haptics-based collision avoidance system, known as *haptic shared autonomy* [7], [8], allows the human operator to maintain full control during teleoperation, while assisting them in collision avoidance. A collaborative haptic guidance system must assist the user in both collision avoidance and navigation to their goal. To minimize the cognitive load to the human user, the system should only apply haptic feedback where necessary and account for any constraints of the vehicle. In this paper, we propose such a haptic guidance system for non-holonomic ground robots. We propose a local RRT*-based path search algorithm to constantly generate a path from their current pose to a predicted goal pose. The haptic feedback combines obstacle avoidance and collision avoidance feedback. Obstacle avoidance guides the user along the safe RRT* path around the obstacle, and collision avoidance feedback helps steer the robot away from the obstacle. We evaluate the performance of our algorithm with a virtual pilot in simulation and experiments with a Sparkfun Jetbot, and demonstrate human-in-the-loop capabilities with a Geomagic Touch. The main contributions of this paper are:

- A haptic guidance algorithm for collaborative teleoperation of ground robots, which combines collision avoidance and obstacle avoidance feedback;
- A non-holonomic RRT*-based path search;
- Analysis of the proposed algorithm to guarantee collision avoidance with obstacles; and
- Simulations and hardware experiments with a virtual pilot and human using a Geomagic Touch haptic device.

Related Work

Our work is relevant to the field of haptic shared control for autonomous vehicles. Following the dichotomy of [9], this work relates to other haptic guidance systems, where the human operator receives feedback to improve the safe operation of the vehicle. Other approaches include: haptic shared autonomy, which improves agreement between the

Mela Coffey and Alyssa Pierson are with the Department of Mechanical Engineering at Boston University, Boston, MA 02215, USA. mcoffey@bu.edu, pierson@bu.edu

human operator and autonomous safety controls [7]; bilateral shared control with haptic feedback [10]; and persistent monitoring with human reshaping of trajectories [11].

In cluttered environments, haptic feedback is useful for collision avoidance, where feedback encodes a repulsive force from obstacles for drones [12], [10] and non-holonomic ground vehicles, such as wheelchairs [13], [14]. While these solutions inform the user to avoid the obstacle, they do not tell the user *how* to maneuver around the obstacle. Other studies use haptic feedback to help the operator steer the robot along a desired path. The authors of [15] proposed a haptic device to provide vibrotactile and force feedback based on the distance from a teleoperated surgical needle to a target. In [16], a holdable haptic device guides the user along a predefined 3D virtual path.

Few studies implement haptic feedback for both collision avoidance and guidance to a target. An assisted control scheme in [17] "haptically" guides the user through their environment while accounting for the kinematic constraints of a wheelchair, but must stop completely when encountering obstructions to switch control schemes. In [18], a haptic control scheme provides collision avoidance via a potential field approach, and obstacle avoidance wherein the robot follows tangential vectors of elliptical contours surrounding each obstacle. Virtual Fixture Based Dynamic Kinesthetic Boundary (VFDKB) planning for unmanned aerial vehicles (UAVs) provides force feedback for both collision avoidance and obstacle avoidance [19]. The virtual fixture (VF) provides obstacle avoidance by guiding the user along a safe path to a predicted user goal. The dynamic kinesthetic boundary (DKB) ensures collision avoidance based on the vehicle's distance to obstacles and slows the vehicle to a halt where necessary.

The aforementioned studies demonstrate the potential for a system which provides haptic assistance for both collision avoidance and obstacle avoidance. Prior studies have not fully explored haptic feedback for safe co-navigation of non-holonomic vehicles along a smooth path to a predicted goal. In contrast, our work provides an in-depth, mathematical performance analysis for both collision and obstacle avoidance feedback, with detailed simulations and human demonstrations for such work. Our work aims to meet the need for a collaborative haptic guidance system that assists the user in both collision avoidance and navigation to their goal. Unlike many proposed shared autonomy systems, our approach does not override the human, but rather helps the human maintain a safe state by avoiding collisions and suggesting safe paths to the predicted goal. And unlike many haptic shared autonomy systems, such as potential field based approaches, our proposed haptic feedback is not always active, and we only provide necessary feedback. The remainder of this paper is organized as follows: Section II introduces our problem formulation. Section III describes the path planning algorithm, and Section IV details how we generate the haptic feedback. Section V contains a performance analysis of the proposed algorithm. We present simulation and experimental results in Section VI, and state our conclusions in Section VII.

TABLE I

MAIN SYMBOLS AND NOTATION

$q_r = [x_r, y_r, \theta_r]^T$	robot pose
$\dot{q}_r = [\dot{x}_r, \dot{y}_r, \dot{\theta}_r]^T$	robot velocity
$\dot{q}_u = [v_u, \omega_u]^T$	user velocity (steering) command
v_r, ω_r	linear and angular velocities of the robot
$f = [f_v, f_\omega]^T$	total haptic force applied to the user
f_o, f_c	obstacle and collision avoidance forces
$q_g = [x_g, y_g, \theta_g]^T$	predicted user goal
P	suggested safe path for the user to follow
λ_p, λ_o	distance from q_r to P and to nearest obstacle edge
u_c, u_o	unit vector corresponding to f_c and f_o
$Q \in \mathscr{R}^2$	environment through which the robot navigates
k	scalar constant to modify the magnitude of f
d	obstacle detection radius
$\lambda_{\text{obs}}^{\text{sf}}$	safe distance from an obstacle
$v_{\max}, a_{\max}$	maximum velocity and acceleration of the robot
$t_{\text{ahead}} = \frac{v_{\max}}{a_{\max}}$	look-ahead time
$\Delta t = t_{i+1} - t_i$	time step
R	radius of the robot
λ_a	distance to an obstacle at which f_c is activated
C_{act}	maximum distance from P such that f_o is active
$C_{\min}$	minimum distance from P such that f_o is active
$C_{\max}$	minimum distance from P at which f_o is maximum

II. PROBLEM FORMULATION

In our collaborative system, a human user shares control with an autonomous mobile ground robot. We assume the human operator teleoperates the robot via a haptic device, and force is fed back to the human as assistance in maintaining safe, collision-free navigation towards a goal. Here, we consider the teleoperation of a ground robot maneuvering through a static two-dimensional environment with circular obstacles. Consider an environment Q, with points in Q denoted q. The robot at point q_r can be modeled as a unicycle robot, where x_r and y_r define the position of the center of the wheels' axis in the environment Q, and θ_r is the orientation (heading angle) of the robot in the global frame. We define the robot dynamics $\dot{q}_r$ as a function of θ_r and user input $\dot{q}_u$:

$$\dot{q}_r = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \dot{q}_u. \tag{1}$$

We define the total haptic force f comprising two components f_o and f_c as

$$f = k(f_o + f_c). \tag{2}$$

Both f_o and f_c have the same structure as f, with linear and angular components. One can think of f_o as an attractive force toward a safe path which guides the user around obstacles, and f_c as a repulsive force away from the nearest obstacle.

Our path planner and propositions utilize an egocentric polar coordinate system [20], [21] with state variables $[r, \delta, \phi]$ (see Figure 2). The robot dynamics in terms of the egocentric polar coordinates are defined by

$$\begin{gathered} \begin{bmatrix} \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -v_r \cos\delta \\ \frac{v_r}{r}\sin\delta \end{bmatrix}, \\ \dot{\delta} = \frac{v_r}{r}\sin\delta + \omega_r. \end{gathered} \tag{3}$$

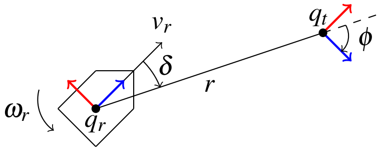


Fig. 2. Egocentric polar coordinate system. r is the radial distance from q_r to the target pose q_t, δ is the angle between the current heading and line of sight, and ϕ is the angle between the target heading and the line of sight.

III. PATH PLANNER

To generate a suggested local safe path around obstacles, and to ultimately determine the force f_o, we propose an RRT*-based path search algorithm for non-holonomic vehicles. Prior studies have modified and implemented RRT* as a real-time path-planning algorithm due to its computational efficiency and its guarantee to find a path. We therefore modify the non-holonomic RRT* algorithm of [20] with control law [21] to search for a safe path from the current robot pose q_r to the user goal q_g while accounting for any obstacles, the size of the robot, and the robot's non-holonomic constraints. For the purposes of our algorithm, and unlike prior work, we plan paths for short intervals, continuously searching for new paths as the human teleoperates the robot through Q and new obstacles are detected.

Note that since f_o is fed back to the user by the robot, the robot must calculate the safe RRT* path. This requires that the robot has knowledge of the surrounding obstacle sizes and locations. We therefore introduce a detection radius d and assume that once an obstacle enters the detection radius, its size and location is known. We call these obstacles *active*, and the local path planner computes a collision-free path around active obstacles only. Assumption 1 emphasizes the robot knowledge of the environment.

Assumption 1 (Robot Knowledge): The robot can detect the size and location of obstacles within a radius d and is responsible for planning a collision-free path using the proposed local RRT* planner.

We also note that the human is not given any information about the safe path generated by RRT* algorithm, as this is only used by the robot to generate f_o. When operating a personal mobility platform such as a powered wheelchair, the human operator must focus on their surroundings to ensure safety while making high-level decisions. We therefore do not wish to cause any distractions by showing them a visual RRT* path to follow; rather, we provide force feedback to steer the robot in a safe direction. Assumption 2 formalizes that only the robot has knowledge of the RRT* path.

Assumption 2 (Human Knowledge): The human operator has no knowledge of the safe path generated by RRT*.

We now proceed with our implementation of a non-holonomic RRT* algorithm. To predict the user goal, $\dot{q}_u$ is assumed to remain constant for time t_{ahead} [12]. The user goal q_g is defined as

$$q_g = \dot{q}_r t_{\text{ahead}} + q_r. \tag{4}$$

The non-holonomic RRT* algorithm is based on the work of Park et al. [20], [21] and generates a safe path from q_r to q_g while accounting for the set of circular active obstacles, as well as the size and kinematic constraints of the robot. Furthermore, the RRT* algorithm [20] is implemented in egocentric polar coordinates with a non-holonomic distance [21] and cost function

$$\text{Dist}(q_r, q_t) = \sqrt{r^2 + k_\phi^2 \phi^2} + k_\delta |\delta - \delta^*|, \tag{5}$$

where q_t is a target node in the tree, k_ϕ and k_δ are scalar constants set to 6.0 and 3.0, respectively, and δ^* is the steering function defined as $\delta^* = \arctan(-k_\phi \phi)$.

We employ RRT* as a local path planner; however, as the robot approaches an obstacle, the generated user goal q_g may end up inside an obstacle. Therefore, to search for a path around the obstacle, we extend the user goal by incrementally increasing t_{ahead} by Δ until q_g no longer lies inside the obstacle. Additionally, we need to establish the appropriate area in which the RRT* algorithm searches. We therefore define the search area based off of the current pose and predicted goal pose, expanding the xy search area by a scalar ε, and the θ search area by scalar ψ. Both of these methods are further described in Algorithm 1. Finally, in order to account for the size of the robot, the RRT* algorithm expands the size of the circular active obstacles by a distance $\lambda_{\text{obs}}^{\text{sf}}$ such that the path generated will be at least $\lambda_{\text{obs}}^{\text{sf}}$ away from any obstacle. Note, however, that $\lambda_{\text{obs}}^{\text{sf}}$ cannot be chosen arbitrarily; in order to prevent collision, $\lambda_{\text{obs}}^{\text{sf}} > v_{\max}\Delta t + R$.

Algorithm 1 summarizes our use of the RRT* algorithm with respect to the proposed force feedback algorithm. Algorithm 1 runs as a single process on the robot, separately from the Force Feedback Algorithm (Algorithm 2). It receives the current q_r, q_g, and $\dot{q}_r$ as inputs. It then extends the goal pose, if necessary, and defines the local search area. Finally, it uses this information to generate a new path P_{new} using the non-holonomic RRT* algorithm as in [20] and publishes the safe path for use in Algorithm 2. If RRT* is unable to find a safe path from q_r to q_g, even after q_g is extended, then P_{new} does not exist; we discuss this case further in the Section V.

Algorithm 1 Local Path Planner: Non-Holonomic RRT*

```
Retrieve $q_r$, $q_g$, $\dot{q}_r$, and the set of active obstacles
while any active obstacle contains $q_g$ do
    $t_{\text{ahead}} += \Delta$
    Recalculate $q_g$ using (4)
end while
Compute the possible limits of the local search area
    $x_{\text{range}} = [x_r - \varepsilon, x_r + \varepsilon, x_g - \varepsilon, x_g + \varepsilon]$
    $y_{\text{range}} = [y_r - \varepsilon, y_r + \varepsilon, y_g - \varepsilon, y_g + \varepsilon]$
Set the local search area for RRT* algorithm
    $x_{\text{search}} = [\min(x_{\text{range}}), \max(x_{\text{range}})]$
    $y_{\text{search}} = [\min(y_{\text{range}}), \max(y_{\text{range}})]$
    $\theta_{\text{search}} = [\theta_r - \psi, \theta_r + \psi]$
Input $q_r$, $q_g$, $\dot{q}_r$, $x_{\text{search}}$, $y_{\text{search}}$, $\theta_{\text{search}}$, and active obstacles into non-holonomic RRT* algorithm from [20]
Retrieve and publish safe path $P_{\text{new}}$
```

Algorithm 1 describes our use of the non-holonomic RRT* algorithm [20] as a local path planner to the predicted user goal q_g. Given our implementation of RRT*, we can introduce Remark 1 as follows.

Remark 1 (Safe RRT Path):* If a safe path P_{new} exists from q_r to q_g, then P_{new} does not collide with obstacles and maintains a distance of at least $\lambda_{\text{obs}}^{\text{sf}}$ away from obstacles.

IV. Haptic Feedback

For our collaborative teleoperation, a human operator co-navigates with the autonomous robot. In complex environments, the robot relies on the human operator for high-level decision making. In turn, the robot generates a suggested safe path for the human operator given its knowledge of obstacles. We propose a force feedback algorithm to assist the human operator in safely navigating the robot through the environment Q. The overall haptic feedback algorithm is summarized in Algorithm 2, which loops until the process is terminated. The force feedback implemented in this paper pushes the human operator's hand in a direction such that the human will steer the robot in a safe direction.

The algorithm first retrieves the most recent safe path P_{new} published by Algorithm 1. Since Algorithm 1 runs continuously, it is possible that the current path P is a better path than P_{new}; specifically, P may be shorter than P_{new}. We therefore introduce a check (lines 2-4) to ensure that the optimal path is used to compute f_o. In addition, we need to ensure that P is still relevant in leading the user to q_g. We introduce this additional check in line 2, comparing q_g and the end point of P. Given the human user's autonomy outperforms the robot's, P is a suggested safe path for the user to follow, and the user may choose to overcome the force feedback and follow a different path (for example, if the human encounters a situation the robot does not understand).

Once P is defined, the algorithm receives $\dot{q}_u$, q_r, and $\dot{q}_r$. It then determines the current active obstacles, computes the Euclidean distances to those obstacles and to P, and predicts q_g as in (4). This information is used in Algorithm 1. Then it computes f_o and f_c and sends the total force feedback f to the haptic device. This process is repeated until terminated.

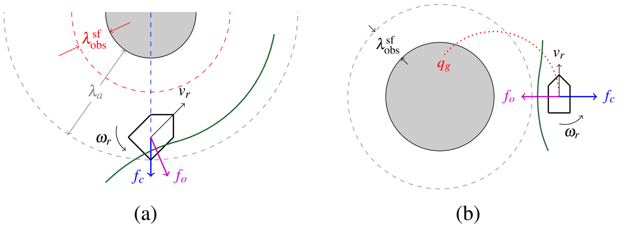
Algorithm 2 Force Feedback

```
1: Retrieve most recent path $P_{\text{new}}$ from Algorithm 1
2: **if** length($P_{\text{new}}$) $<$ length($P$) **or** $P$ irrelevant **then**
3:     $P = P_{\text{new}}$
4: **end if**
5: Retrieve $q_r$, $\dot{q}_r$, $\dot{q}_u$, and set of active obstacles
6: Calculate $\lambda_o$, $\lambda_p$, and $q_g$
7: Calculate $f_c(\lambda_o)$ and $f_o(\lambda_p)$
8: Send total force feedback $f = k\,(f_o(\lambda_p) + f_c(\lambda_o))$
```

Algorithm 2 describes the process of determining the forces to feed back to the user. The total force feedback is a function of the obstacle avoidance force $f_o(\lambda_p)$ and the collision avoidance force $f_c(\lambda_o)$. In the following subsections, we provide further details about these two components.

A. Collision Avoidance Force

Similar to [22] and [19], our collision avoidance algorithm implements a DKB-based approach to provide a repulsive force feedback away from obstacles. The DKB-based approach results in robust haptic feedback with reduced congitive load and is preferred by users compared to potential field based approaches [23]. For repulsive feedback, the primary force feedback direction u_c of f_c is defined as the normalized radial vector pointing from the nearest obstacle to the robot (see Figure 3(a)). This vector is then scaled depending on $\lambda_o(t)$ such that the force feedback f_c is



$\lambda_{\text{obs}}^{\text{sf}}$
λ_a
v_r
ω_r
f_c
f_o
$\lambda_{\text{obs}}^{\text{sf}}$
q_g
v_r
f_o
f_c
ω_r

(a) (b)

Fig. 3. (a) Example of both f_c and f_o contributing to the total force vector f to avoid the gray circular obstacle. f_c is normal to the obstacle and serves to push the robot away from the obstacle, while f_o is normal to the path and pulls the robot toward the safe path (green curve). (b) Example of a scenario in which f_c and f_o cancel out. The red dotted line is the predicted robot trajectory, leading to the predicted goal q_g which must lie inside the obstacle for f to be active. For the forces to cancel out, the safe path must lie between the robot and the obstacle as shown, as f_o must be perpendicular to the path and drawing the robot toward the path.

$$f_c = \begin{cases} u_c(t)(e^{\lambda_a - \lambda_o(t)} - 1), & \lambda_o(t) \leq \lambda_a \\ 0, & \lambda_o(t) > \lambda_a \end{cases}. \tag{6}$$

The collision avoidance f_c is constructed such that the magnitude of force feedback increases as the robot nears an obstacle. One can show that our proposed formulation for f_c is bounded above by $e^{\lambda_a} - 1$ and below by 0 (see Figure 4).

B. Obstacle Avoidance Force

The obstacle avoidance force was inspired by [19], [24] and aims to restrict motion along the safe RRT* path, generated by Algorithm 1, while providing the user with information about the deviation from said path. Recall that the RRT* path only serves as a suggested safe path for the user, although the user does not have knowledge of the safe path (Assumptions 1 and 2). Therefore, we wish to determine the force required to "nudge" the user onto the safe path.

We first determine the nearest node on the safe path. This is the node corresponding to the minimum Euclidean distance to the robot. We then define the normalized vector u_o perpendicular to the path and pointing in the direction of the robot (see Figure 3(a)). This serves as the primary force feedback direction for the force component f_o. We then scale u_o based on the distance $\lambda_p(t)$ to the safe path such that the force feedback f_o is

$$f_o = \begin{cases} u_o(t)\left(e^{\alpha(\lambda_p(t) - C_{\min})} - 1\right), & C_{\min} \leq \lambda_p(t) \leq C_{\max} \\ u_o(t)\left(e^{\alpha(C_{\max} - C_{\min})} - 1\right), & C_{\max} < \lambda_p(t) \leq C_{\text{act}} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\alpha = \frac{\lambda_a}{C_{\max} - C_{\min}}$. In this way, if the robot is already near the safe path (i.e. within $C_{\min}$) or too far from the path (i.e. farther than C_{act}), f_o will not be applied. As the user steers

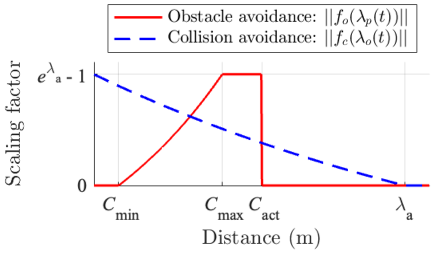


Fig. 4. Force scaling for obstacle avoidance (red solid line) and collision avoidance (blue dashed line). Note that both forces share the same bounds.

the robot farther from the safe path, the force magnitude will increase exponentially but will be bounded when $\lambda_p(t)$ reaches C_{max}. Furthermore, the force feedback f_o is designed in such a way that as the user moves farther from the path, the magnitude of the force feedback will increase, and vice versa. If Algorithm 1 cannot find a path to the predicted path, and the most recent path Algorithm 2 receives is outdated, then C_{act} serves to ensure f_o does not push the robot to an irrelevant path. One can show that the obstacle avoidance force f_o shares the same bounds as f_c.

V. PERFORMANCE WITH A VIRTUAL PILOT

It is important to note that, in reality, it is possible that the human operator may ignore the haptic commands and choose to override the force feedback. For our analyses and some experiments, we represent the human user as a PD controller to generate the user's steering commands. However, in practice with a human operator, the PD controller plays no part in the teleoperating task – the steering commands are generated solely from user input, and the start and goal poses are unknown to the robot. We will assume that the virtual pilot is a "reasonable human" (Assumption 3). In other words, we assume that the user follows any and all force feedback commands, hence the total steering command (9). Assumption 3 allows us to explore the behavior of the robot as a response to the force feedback provided.

Assumption 3 (Reasonable Human): The human follows the force feedback commands provided by Algorithm 2 such that the total velocity of the robot is defined by (9).

To evaluate the performance of our algorithm, we represent the user as a simple PD controller as in [12], [25], [26]. This virtual pilot has no information about the obstacles, and its only ability is to steer the robot from a predefined start pose to a predefined goal pose. Given $\dot{q}_u(t_i)$ from the virtual pilot and the force feedback $f(t_i)$, the new q_r at time t_{i+1} is

$$q_r(t_{i+1}) = \begin{bmatrix} \cos\theta(t_i) & 0 \\ \sin\theta(t_i) & 0 \\ 0 & 1 \end{bmatrix} \dot{q}_{\text{tot}}(t_i)\Delta t + q_r(t_i), \tag{8}$$

where

$$\dot{q}_{\text{tot}}(t_i) = \dot{q}_u(t_i) + f(t_i). \tag{9}$$

We use this model to evaluate the performance of our proposed approach. In the following analyses, we consider the virtual pilot (PD controller) steering the robot through a locally smooth environment with bounded curvature and continuously differentiable obstacles.

We first evaluate the boundedness of the force feedback. We noted in Section IV that f_c and f_o are bounded, and thus the total force feedback f is bounded when these components are active alone; i.e. when one of the forces is 0 and the other is not. In Lemma 1 we prove that the total force feedback is bounded when both forces are active.

Lemma 1: Consider the force feedback in which both f_c and f_o are active. Thus, $\lambda_o \in [0, \lambda_a]$ and $\lambda_p \in [C_{\text{min}}, C_{\text{act}}]$. Then the total force feedback f is bounded.

Proof: The total force with the given inputs is

$$f = k\left(u_c(e^{\lambda_a - \lambda_o(t)} - 1) + u_o(e^{\alpha(\lambda_p(t) - C_{\text{min}})} - 1)\right). \tag{10}$$

With the given inputs, the upper and lower bounds occur when the two forces are parallel. Thus, f is bounded from above when the two forces are in the same direction by

$$k\left((e^{\lambda_a} - 1) + (e^{\alpha(C_{\text{max}} - C_{\text{min}})} - 1)\right) = 2k(e^{\lambda_a} - 1), \tag{11}$$

and from below when the two forces are opposite by 0. ■

Next, we consider the unlikely scenario in which the two force feedback components f_c and f_o cancel out. Such an instance could happen if the vectors u_c and u_o are pointed in opposite directions, and if f_c and f_o have the same magnitude. Proposition 1 claims that if this happens, the robot cannot collide with the obstacle.

Proposition 1: In the case where $f_c = -f_o$, the robot will not collide with an obstacle.

Proof: The proof follows the schematic of Figure 3(b). Assume $f_c = -f_o$ at time t_i, then f_c and f_o have the same magnitude and act opposite one another. Note that, since f_c only acts normal to and outward from the obstacle, then f_o can only be opposite when it acts normal to and toward the obstacle. Thus the safe RRT* path from Algorithm 1 must lie between the robot and the obstacle (Figure 3(b)).

Given Remark 1, and since f_o is active, a safe path exists around the obstacle which is at least a distance of $\lambda_{\text{obs}}^{\text{sf}}$ away from the obstacle edge. Since the safe path is between the robot and obstacle, then $\lambda_o(t_i) \geq \lambda_{\text{obs}}^{\text{sf}}$.

We can also assume that, since Algorithm 2 calculates f_o based on the most recently generated local path, and given Assumption 3, then the current pose, particularly the heading angle, is approximately the same as that generated by the safe path. Therefore, the robot velocity vector must be roughly perpendicular to f_c and f_o, as in Figure 3(b).

Suppose the robot moves toward the obstacle at time t_i when the forces cancel, following a path similar to the red dotted line in Figure 3(b). Then the robot will move closer to the safe path, decreasing f_o, and closer to the obstacle, increasing f_c, at time t_{i+1}. Then, $f_c > f_o$ at time t_{i+1}, which will prevent the collision with the obstacle. ■

Proposition 1 claims that a cancellation in f_c and f_o cannot cause collisions with obstacles. To evaluate the ability of Algorithm 2 to follow a safe RRT* path, we consider the case where only the obstacle avoidance component f_o is active, and $f_c = 0$. In Proposition 2 we evaluate the path following performance by evaluating the tracking error $\lambda_p(t)$.

Proposition 2: Assume no collision avoidance force f_c and a safe RRT* path P exists to the predicted goal (Remark 1). Assume also that $\lambda_p(t) \in [C_{\text{min}}, C_{\text{act}}]$. Then the force feedback $f = kf_o$ decreases the tracking error in reference to the safe RRT* path.

Proof: Recall the egocentric dynamics (3). Let q_t be the nearest node on the RRT* path to the robot, and define the tracking error as the distance r from the robot to the path, as in Figure 2. The tracking error r decreases if and only if $\dot{r} < 0$. From (3),

$$\dot{r} = -v_r \cos\delta. \tag{12}$$

We will observe $\dot{r}$ in two possible cases based on the orientation of the robot with respect to the safe path.

Case 1: $\delta(t) \in (-\frac{\pi}{2}, \frac{\pi}{2})$. In this case, the robot is oriented toward the path, and we can assume that the vector f is oriented in approximately the same direction as r (i.e. $f \approx r$). Therefore, the total velocity from (9) is

$$v_r = v_u + k\|f_o\|, \tag{13}$$

where $\|f_o\|$ is the linear component of f_o. Given Assumption 3, $v_u = \|\dot{q}_u\| = 1$. Plugging (13) into (12), we obtain $\dot{r} = -(1+k\|f_o\|)\cos\delta$. Since $1+kf_o > 0$ and $\cos\delta > 0$, then $\dot{r}(t) < 0\ \forall t$ such that $f = kf_o$ and $\delta \in (-\frac{\pi}{2}, \frac{\pi}{2})$.

Case 2: $\delta(t) \in (-\pi, -\frac{\pi}{2}]$ or $\delta(t) \in [\frac{\pi}{2}, \pi)$. Assuming $f \approx r$, then f results in a negative linear velocity with respect to v_r. Thus $v_r = v_u - k\|f_o\|$. Then (12) becomes $\dot{r} = -(1 - k\|f_o\|)\cos\delta$. Since $\cos\delta < 0$, we observe that $\dot{r} < 0$ when $1 - k\left(e^{\alpha(r - C_{\min})} - 1\right) < 0$. This happens when

$$r > \frac{1}{\alpha}\ln\left(\frac{1+k}{k}\right) + C_{\min}. \tag{14}$$

Thus, when the robot is oriented away from the path, the tracking error will begin decreasing when the distance reaches that defined in (14). Furthermore, when $f = kf_o$ and given Assumption 3, the maximum distance the robot can move from the path is given by (14). ■

Proposition 2 shows that the obstacle avoidance force f_o will guide the user toward the safe path. We now consider the case where Algorithm 1 cannot find a safe RRT* path to the user goal and thus the path P does not exist, e.g. the predicted user goal lies on the other side of a wall. In such a case, $f_o = 0$ and Algorithm 2 will rely solely on the collision avoidance force f_c. Proposition 3 shows that although the goal is unattainable, the proposed algorithm will prevent collisions with the obstacle.

Proposition 3: Assume an initial condition of $\lambda_o(t) \in (\lambda^{\mathrm{sf}}_{\mathrm{obs}}, \lambda_a]$ such that $f = kf_c$. Then for all $t > 0$,

$$\lambda_o(t) \geq \lambda_a - \ln\left(\frac{1+k}{k}\right), \tag{15}$$

where

$$k > \frac{1}{e^{\lambda_a} - 1}. \tag{16}$$

Proof: Recall the egocentric polar coordinate system shown in Figure 2 and described by (3). Let the target pose q_t be the nearest point on the edge of the nearest obstacle. Then r is the distance between the robot and obstacle edge. We will consider two cases based on the orientation of the robot with respect to the obstacle and determine the distance r at which the robot no longer moves toward the obstacle (i.e. when $\dot{r}(t) = 0$).

Case 1: $\delta(t) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. In this case, the linear velocity v_r of the robot is oriented toward the obstacle, while that of the force feedback f is oriented opposite the robot velocity, since f points normal from the obstacle. Therefore, the total velocity of the robot can be described by

$$v_r = v_u - k\|f_c\|. \tag{17}$$

Given Assumption 3, $v_u = \|\dot{q}_u\| = 1$. Then plugging (17) into (12), we obtain $\dot{r} = -(1 - k\|f_c\|)\cos\delta$. We observe that $\dot{r} = 0$ either when $\delta = \pm\frac{\pi}{2}$, or when $1 - k(e^{\lambda_a - r} - 1) = 0$. By solving for r, we can conclude that $\dot{r} = 0$ when

$$r = \lambda_a - \ln\left(\frac{1+k}{k}\right). \tag{18}$$

Case 2: $\delta(t) \in (-\pi, -\frac{\pi}{2})$ or $\delta(t) \in (\frac{\pi}{2}, \pi)$. In this case, both v_r and f are oriented away from the obstacle. Thus, the total velocity becomes

$$v_r = v_u + k\|f_c\|. \tag{19}$$

Given the same assumptions as in (i), then $\dot{r} = -\left(1 + k(e^{\lambda_a - r} - 1)\right)\cos\delta$. Solving for r, we observe that $\dot{r} = 0$ only when $r = \lambda_a - \ln\left(\frac{k-1}{k}\right)$. Note that this distance is greater than the minimum distance calculated in Case 1. Therefore, the minimum distance between the robot and obstacle is given by (18) when $f(t) = kf_c(t)\ \forall t$.

Although it is possible for $r < 0$ in (18), the positive constant k cannot be chosen arbitrarily. Rather, k should be selected such that $r > 0\ \forall t$ while accounting for the size of the robot. We can set $r > 0$ in (18) and solve for k to obtain (16). Thus, although (18) can be negative, by (16) it will always be positive, resulting in a collision-free configuration $\forall t > 0$, proving the claim of the proposition. ■

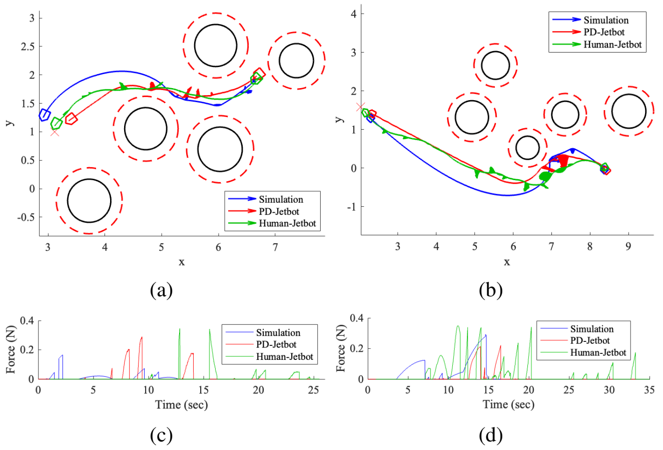
Propositions 1-3 showed that given a collision-free initial condition, our approach will prevent any collisions with obstacles in a locally smooth environment with continuously differentiable obstacles and appropriately selected constants.

VI. EXPERIMENTS

We conducted a series of virtual pilot simulations and hardware experiments, and performed human-in-the-loop demonstrations with the haptic device, to demonstrate collision-free navigation to the goal under the haptic feedback. We implemented our algorithm using Robot Operating System (ROS), with a desktop computer (8-core, 32GB RAM, Ubuntu 20.04) running Algorithms 1 and 2 in Python. Algorithm 1 generated RRT* paths every 300 to 600 ms, and Algorithm 2 updated the force feedback at an average rate of 300-400 Hz. Assumptions 1-3 hold for the virtual pilot experiments in Subsections VI-A and VI-B, and Assumptions 1 and 2 hold for the human experiments in Subsection VI-C. Experiments and animated simulations with suggested paths can be found in the supplemental video for the environment shown in Figures 5(a) and 6(a).

A. Virtual Pilot Simulations

We implemented the virtual pilot controller from Section V as a proxy for a human user. The virtual pilot has no information about any obstacles, and its only purpose is to



(a) (b)

(c) (d)

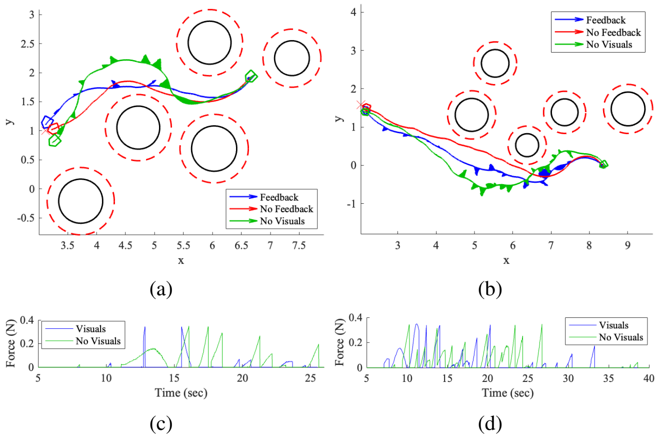
Fig. 5. Simulation (blue) compared with the PD-controlled robot (red) and human-controlled robot (green) experiments for two selected environment. (a) and (b) show the trajectories of the robot navigating through the two environments under each condition from the blue · to the red ×, with the proposed algorithm providing force feedback, represented by arrows, to avoid obstacles. (c) and (d) show the respective force magnitude applied over time.

steer the robot from the start pose to the goal pose. We performed 100 simulations and recorded (i) the minimum distance to any obstacle and (ii) the minimum distance from the goal for any trial. For each trial, we randomly generated a start and a goal pose and five randomly positioned and randomly sized obstacles. Start and goal poses were restricted to be at least a distance of λ_a away from any obstacle to ensure that the robot reached its goal. Additionally, the five obstacles were generated such that they did not overlap one another. Across all 100 simulations, the minimum distance to an obstacle was 0.19, and the maximum distance from the goal was 0.25, which was the distance at which the PD controller terminated. We conclude from these simulations that for all 100 trials, the robot was able to reach its goal. Our algorithm also satisfied (15) for $\lambda_a = 0.4$ and $k = 3.0$, and thus $\lambda_o(t) > R \ \forall t$, confirming that the robot never collided with an obstacle.

We then selected two randomly generated environments which we believed to be challenging. In Figure 5, we present simulations with the virtual pilot through the two environments. The trajectory of the robot under the haptic feedback is shown in blue in 5(a) and 5(b), and the corresponding force magnitude applied to the virtual pilot is shown in 5(c) and 5(d). Results are shown alongside those from hardware-in-the-loop and human-in-the-loop experiments for comparison.

B. Virtual Pilot + Jetbot

To ensure our algorithm could be implemented in hardware, we performed experiments with the virtual pilot on the SparkFun Jetbot v2.0[1], powered by the NVIDIA Jetson Nano[2]. The desktop computer sent steering commands (9) to the robot via WiFi. An Optitrack Motion Capture (MOCAP) system[3] provided position and orientation data. Results are shown in Figure 5 in red, where we observe similar results to the simulations.



(a) (b)

(c) (d)

Fig. 6. Human teleoperation experiments in three cases: visual feedback + haptic feedback, visual feedback without haptic feedback, and haptic feedback without visual feedback. (a) and (b) show the trajectory of the robot navigating from the blue · to the red ×, with the proposed algorithm providing force feedback, represented by arrows, to avoid obstacles. (c) and (d) show the respective force magnitude over time.

C. Human Pilot Demonstration

An experimenter tested the proposed algorithm with a haptic device, the Geomagic Touch[4], to observe the forces applied to a human pilot. We navigated the Jetbot through the MOCAP space using the Geomagic Touch while standing outside the environment, such that we had an imperfect third-person view of the robot and projected environment shown in Figure 1(a). The position of the end effector on the Geomagic Touch was mapped to linear and angular velocity commands sent to the robot as f, with forward (back) positions mapping to positive (negative) linear velocity v_u, and left (right) positions mapping to positive (negative) angular velocity ω_u. The computer sent haptic feedback (2) to the Geomagic Touch using the same mappings, such that the operator's hand was pushed in a direction that steered the robot in a safe direction, resulting in collaborative navigation.

We performed experiments in the same two environments as the previous subsections under three conditions: (i) visual feedback only, (ii) visual feedback + haptic feedback, and (iii) haptic feedback without visual feedback. The results for these experiments are shown in Figure 6, and case (ii) is shown alongside the virtual pilot experiments in Figure 5. We observed results similar to the virtual pilot experiments, with force feedback applied to the human more frequently than the virtual pilot, a result of the human taking longer to react to the force feedback compared to the virtual pilot.

From Figures 6(c) and 6(d), we observe little high-frequency, noise-like forces applied by the haptic device to the user during operation. Such forces are known as buzzing and can result in perceived instability [27]. However, because the purpose of our algorithm is co-navigation, rather than haptic perception, we consider minimal buzzing acceptable

[1] www.sparkfun.com
[2] https://developer.nvidia.com/embedded/jetson-nano-developer-kit
[3] https://optitrack.com/
[4] https://www.3dsystems.com/haptics-devices/touch

for our applications, and we can furthermore conclude that an update rate of 300-400 Hz is sufficient.

VII. Conclusions

In this paper we present a novel haptic guidance system to help human users co-navigate a ground robot through a remote environment. Our proposed algorithm provides force feedback for both collision avoidance and obstacle avoidance. The collision avoidance force applies force feedback away from an obstacle, while the obstacle avoidance force applies feedback toward a safe path generated by a local non-holonomic RRT*-based algorithm. We present performance analyses for our algorithm and demonstrate its performance through a series of simulations, hardware experiments, and human pilot demonstrations. Unlike prior work in the literature, we present thorough mathematical analyses of our performance with a virtual pilot, and demonstrate that the user need not have information about obstructions in order to safely reach their goal. Our algorithm also grants the human full control of the robot. Our results show that the proposed algorithm could be a useful and collaborative teleoperation system for ground robots.

Future work will include further evaluation of the proposed algorithm through user studies, particularly studying the ability to co-navigate a robot through complex environments, as well as the effect of buzzing on user experience. Future work can also implement more sophisticated methods for predicting the user goal, such as Bayesian [28] or machine learning [29] methods. Increasing the accuracy of the user goal will improve the user experience and theoretically prevent disagreements between the human and force feedback algorithm.

References

[1] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, and J.-K. Kämäräinen, "Ar-based interaction for human-robot collaborative manufacturing," *Robot Comput Integr Manuf*, vol. 63, p. 101891, 2020.

[2] A. Vysocký, P. Oščádal, M. Vocetka, P. Novák, and Z. Bobovský, "Improved mutual understanding for human-robot collaboration: Combining human-aware motion planning with haptic feedback devices for communicating planned trajectory," *Sensors*, vol. 21, no. 11, p. 3673, 2021.

[3] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 57–68, 2012.

[4] S. G. McGill, G. Rosman, T. Ort, A. Pierson, I. Gilitschenski, B. Araki, L. Fletcher, S. Karaman, D. Rus, and J. J. Leonard, "Probabilistic risk metrics for navigating occluded intersections," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4322–4329, Oct 2019.

[5] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2994–3008, 2018.

[6] D. Schitz, S. Bao, D. Rieth, and H. Aschemann, "Shared autonomy for teleoperated driving: A real-time interactive path planning approach," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 999–1004.

[7] D. Zhang, R. Tron, and R. P. Khurshid, "Haptic feedback improves human-robot agreement and user satisfaction in shared-autonomy teleoperation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3306–3312.

[8] "Toward human-vehicle collaboration: Review and perspectives on human-centered collaborative automated driving," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103199, 2021.

[9] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 475–491, 2020.

[10] C. Masone, M. Mohammadi, P. R. Giordano, and A. Franchi, "Shared planning and control for mobile robots with integral haptic feedback," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1395–1420, 2018.

[11] M. Cognetti, M. Aggravi, C. Pacchierotti, P. Salaris, and P. R. Giordano, "Perception-aware human-assisted navigation of mobile robots on persistent trajectories," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4711–4718, 2020.

[12] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. van Paassen, "Artificial force field for haptic feedback in uav teleoperation," vol. 39, no. 6, pp. 1316–1330, 2009.

[13] "Haptic control for powered wheelchair driving assistance," *IRBM*, vol. 36, no. 5, pp. 293–304, 2015.

[14] G. Bourhis and M. Sahnoun, "Assisted control mode for a smart wheelchair," in *2007 IEEE 10th Int. Conf. Rehabil. Robot*, 2007, pp. 158–163.

[15] M. Aggravi, D. A. L. Estima, A. Krupa, S. Misra, and C. Pacchierotti, "Haptic teleoperation of flexible needles combining 3d ultrasound guidance and needle tip force feedback," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4859–4866, 2021.

[16] J. M. Walker and A. M. Okamura, "Continuous closed-loop 4-degree-of-freedom holdable haptic guidance," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6853–6860, 2020.

[17] E. Vander Poorten, E. Demeester, E. Reekmans, J. Philips, A. Hüntemann, and J. De Schutter, "Powered wheelchair navigation assistance through kinematically correct environmental haptic feedback," in *2012 IEEE Int. Conf. Robot. Autom.*, 2012, pp. 3706–3712.

[18] I. Mantegh, J. Liao, and T. He, "Integrated collision avoidance for unmanned aircraft systems with harmonic potential field and haptic input," in *2020 IEEE/SICE Int. Symp. Syst. Integr.*, 2020, pp. 927–932.

[19] X. Hou, X. Wang, and R. Mahony, "Haptics-aided path planning and virtual fixture based dynamic kinesthetic boundary for bilateral teleoperation of vtol aerial robots," in *2016 35th Chin. Control Conf. CCC*, 2016, pp. 4705–4710.

[20] J. J. Park and B. Kuipers, "Feedback motion planning via non-holonomic rrt* for mobile robots," in *2015 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 4035–4040.

[21] J. J. Park and B. Kuipers, "A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4896–4902.

[22] X. Hou, P. Fang, and Y. Qu, "Dynamic kinesthetic boundary for haptic teleoperation of unicycle type ground mobile robots," in *2017 36th Chin. Control Conf. CCC*, 2017, pp. 6246–6251.

[23] X. Hou and R. Mahony, "Dynamic kinesthetic boundary for haptic teleoperation of vtol aerial robots in complex environments," vol. 46, no. 5, pp. 694–705, 2016.

[24] J. Abbott and A. Okamura, "Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures," in *2006 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2006, pp. 169–175.

[25] T. M. Lam, M. Mulder, and M. M. van Paassen, "Haptic interface in uav tele-operation using force-stiffness feedback," in *2009 IEEE Conf. Syst. Man Cybern.*, 2009, pp. 835–840.

[26] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Switching-based mapping and control for haptic teleoperation of aerial robots," in *2012 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 2629–2634.

[27] S. Choi and H. Z. Tan, "Perceived Instability of Virtual Haptic Texture: III. Effect of Update Rate," *Presence: Teleoperators and Virtual Environments*, vol. 16, no. 3, pp. 263–278, 06 2007.

[28] E. Demeester, A. Huntemann, D. Vanhooydonck, G. Vanacker, A. Degeest, H. V. Brussel, and M. Nuttin, "Bayesian estimation of wheelchair driver intents: Modeling intents as geometric paths tracked by the driver," in *2006 IEEE Int. Conf. Intell. Robots Syst.*, 2006, pp. 5775–5780.

[29] V. K. Narayanan, A. Spalanzani, and M. Babel, "A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance," in *2016 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4700–4707.