

# Goal Blending for Responsive Shared Autonomy in a Navigating Vehicle

Yu-Sian Jiang,<sup>1</sup> Garrett Warnell,<sup>2</sup> Peter Stone<sup>1,3</sup>

<sup>1</sup> The University of Texas at Austin

<sup>2</sup> Army Research Laboratory

<sup>3</sup> Sony AI

sharonjiang@utexas.edu, garrett.a.warnell.civ@mail.mil, pstone@cs.utexas.edu

## Abstract

Human-robot shared autonomy techniques for vehicle navigation hold promise for reducing a human driver’s workload, ensuring safety, and improving navigation efficiency. However, because typical techniques achieve these improvements by effectively removing human control at critical moments, these approaches often exhibit poor responsiveness to human commands—especially in cluttered environments. In this paper, we propose a novel goal-blending shared autonomy (GBSA) system, which aims to improve responsiveness in shared autonomy systems by blending human and robot input during the selection of *local navigation goals* as opposed to low-level motor (servo-level) commands. We validate the proposed approach by performing a human study involving an intelligent wheelchair and compare GBSA to a representative servo-level shared control system that uses a policy-blending approach. The results of both quantitative performance analysis and a subjective survey show that GBSA exhibits significantly better system responsiveness and induces higher user satisfaction than the existing approach.

## 1 Introduction

Human-robot shared autonomy systems for a navigating vehicle promise to provide all the safety and efficiency of autonomous systems while retaining the customization and spontaneity of fine-grained individual control. Examples of these shared control systems include vehicle “guardian” systems that intervene during safety-critical situations (Maurer et al. 2018), intelligent wheelchairs which seek to assist users during operation (Simpson et al. 1998), and shared control teleoperation systems for search-and-rescue, exploration, and hazardous waste clean-up (Crandall and Goodrich 2002).

While existing techniques for shared autonomy have been shown to be successful in certain scenarios, in complex (e.g., cluttered) environments, many approaches lack the kind of *responsiveness* that would truly deliver on the promise of these systems. We consider responsiveness to be the quality of a shared autonomy system that is present when user-supplied commands induce desired changes in navigation behavior. In this paper, we operationalize responsiveness as the Hausdorff distance between a user’s intended path and

the path actually driven by the vehicle; the lower this distance, the greater the responsiveness. Unfortunately, neither of the two dominant paradigms for shared autonomy in the literature—*task-level shared control* and *servo-level shared control* (see, e.g., (Wang and Liu 2014))—seem to prioritize responsiveness. Task-level shared control techniques perform task inference, but otherwise give full control to either the human or the robot. Such systems lack responsiveness in that, if the task dictates that the robot has full control, the system ignores input from the human that might communicate desired changes in behavior. Servo-level shared control techniques, on the other hand, provide assistance by combining the user’s low-level motion commands with those output by the local motion planner of the robot. However, current implementations of servo-level systems employ restrictive safety policies that conservatively limit the responsiveness of the system in a cluttered environment.

In this work, we explore an alternative approach for shared autonomy that we refer to as *shared goal setting*, in which the human and robot input are blended at the level of determining a *local navigation goal* for the vehicle. Shared goal setting stands in stark contrast to the widely-studied shared control approach described above: while both represent approaches to shared autonomy, the former blends inputs to compute *goals* while the latter blends inputs to compute *controls*. In fact, in the existing literature, the terms “shared control” and “shared autonomy” are often used interchangeably, reflecting that the majority of existing shared autonomy approaches opt to blend human and robot inputs at the lowest level of control.

We hypothesize that safe and responsive shared autonomy navigation systems can be achieved through a novel shared goal setting method called *goal blending shared autonomy* (GBSA). GBSA interprets a user’s command as information that informs the selection of a new nearby goal for a local motion planner, resulting in re-planning, i.e., a response from the system. To test this hypothesis, we applied our method to a simulated robot wheelchair in human experiments to analyze the system performance and gathered user survey results. Our studies suggest that GBSA is more responsive than both servo-level and task-level shared control alternatives, while retaining most of their advantages in user workload, safety, and navigation efficiency.

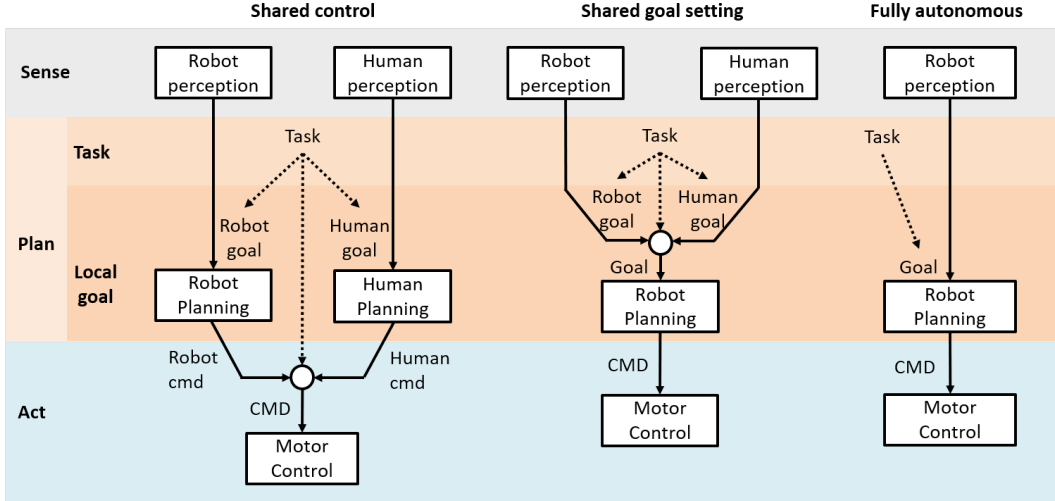


Figure 1: In this paper, we propose a new paradigm for shared autonomy, *shared goal setting*, and contrast it with the widely-studied shared control approaches. Shared goal setting blends human and robot input to compute *goals* instead of motion controls, thereby enabling safe and responsive navigation in complex environments.

## 2 Related Work

While many approaches have been proposed for shared autonomy, previous shared control methods for navigating vehicles mainly fall into two camps: task-level shared control and servo-level shared control (Wang and Liu 2014).

Servo-level shared control aims to generate a combination of human and autonomous control inputs at the motor control level so as to optimize some metric of performance (e.g., safety, smoothness, or obedience). One common servo-level approach is based on *policy blending* (Dragan and Srinivasa 2013), in which the robot command and the human command are combined using a weighted average, where the weights are assigned according to an arbitration function. Proposed arbitration functions might, e.g., assign a constant weight to each input, or modify the weights in real-time according to constraints such as safety, smoothness, and confidence (Poncela et al. 2009; Urdiales et al. 2011; Li, Chen, and Wang 2011; Anderson, Karumanchi, and Iagnemma 2012; Wang and Liu 2014). The arbitration weights may severely downweight the human command to ensure safety in cluttered environments. Another approach to servo-level shared control incorporates the human input into the objective function of a local motion planner such that the arbitration between the human command and the robot command is implicitly done in the optimization process of the motion planner (Erlien, Fujita, and Gerdes 2013; Inigo-Blasco et al. 2014). Again, the output motion commands of such a motion planner may deviate from the human inputs significantly when an obstacle avoidance objective takes effect. Yet another line of work for shared control at the servo-level either filters the human commands based on the robot commands (Erdogan and Argall 2017) or removes the human’s constraint-violating inputs altogether and completely replaces them with robot commands (Broad, Murphey, and Argall 2019). These approaches limit the responsiveness of

the shared control systems since they ignore user commands in certain scenarios.

Task-level shared control, on the other hand, employs fully-autonomous takeovers for certain predefined tasks, full human control for other tasks. When a condition is triggered (either a certain scenario is detected by the robot or an assistance function is activated manually), the robot takes over the full control authority from the user to perform the driving task. In one group of task-level approaches, the robot generates its behavior using a basic set of motion primitives such as obstacle avoidance, door passing, wall/corridor following, and docking for the driving task (Connell and Viola 1990; Simpson et al. 1998; Rao et al. 2002; Bruemmer et al. 2005; Xu et al. 2016). In another group of task-level shared control techniques, the robot is able to navigate toward a global navigation goal based on a global map, in which the global navigation goal may be selected from a UI or perhaps be inferred from the user’s inputs or trajectory (Demeester et al. 2008; Bonarini et al. 2012; Xu et al. 2016; Javdani et al. 2018). After the task is completed, the robot relinquishes the control back to the user. Therefore, at any time instant, the vehicle is controlled in only one of two modes: full manual control or full autonomy. That is, when the robot is in fully-autonomous mode, the human motion command is not considered. While some works combine servo-level and task-level shared control techniques in a single system (Argall 2015, 2016; Ghorbel et al. 2018; Rakita et al. 2018, 2019), thus far, they are still intrinsically limited in responsiveness similar to the servo-level shared control.

To overcome the limitations of both camps, we propose here a novel goal blending process, which blends control commands at the local goal level that yields responsiveness while allowing the robot to manage local motor control entirely for safety. In this manner, both safety and responsiveness are achieved at the same time. Since task-level shared

control approaches are not at all concerned with responsiveness, our study will focus on comparing the proposed approach with servo-level shared control.

### 3 Goal Blending Shared Autonomy

We propose here a novel shared goal setting paradigm called *Goal-Blending Shared Autonomy* (GBSA) that performs blending at the local goal level of autonomous navigation. GBSA can be used for vehicles navigating in complex (e.g., cluttered) environments in scenarios where: (1) the user’s intended destination is not given *a priori* and there are no pre-specified goal candidates, (2) the human policy is unknown, and (3) both safety and responsiveness are considered critical. Existing shared control approaches utilizing a goal prediction algorithm often assume that the robot has prior knowledge of the global goal candidates or the human policy given their goal (Argall 2015; Ghorbel et al. 2018). These methods limit the goal space of the problem, such that the solutions are restricted to known environment or targets. We avoid such restrictive assumptions in our work to make the solution ready for scenarios in the wild.

#### 3.1 Goal Inference and Blending

We now describe how GBSA uses human steering commands to compute a local goal for the motion planner. We divide this process into two stages. In the first stage, we transform a user’s steering command  $u \in U$  into a user goal  $\mathbf{G}_u$ . In the second stage, we blend  $\mathbf{G}_u$  with information from the autonomous system in order to compute a blended goal  $\mathbf{G}_b$ —a feasible local goal coordinate for motion planning. Fig. 1 depicts the proposed process for doing so.

**User goal inference** Referring to Fig. 2, the user goal  $\mathbf{G}_u$  is computed by first inferring a user goal angle  $\theta_u$  relative to the robot heading, and then setting the goal location based on the size of the local area under consideration.<sup>1</sup> In particular, given a user steering command  $u$ , the user goal angle at time  $t$  is computed according to

$$\begin{aligned} \theta_{u,t} &= \theta_{u,t-1} + \theta_{r,t-1}^{(g)} + K_u(u_t) \cdot \Delta t - \theta_{r,t}^{(g)} \\ &= \theta_{u,t-1} + K_u(u_t) \cdot \Delta t - \omega_v \Delta t, \end{aligned} \quad (1)$$

where  $\theta_{r,t}^{(g)}$  denotes the vehicle orientation at time  $t$  with respect to a global coordinate system,  $\omega_v$  denotes the angular velocity of the vehicle, and  $K_u(\cdot)$  represents a transform function from the user command signal to an angle, where  $K_u(\cdot)$  may represent fixed scalar multiplication in the simplest case or, e.g., a nonlinear function that adjusts sensitivity to the user commands.  $\Delta t$  is set to be the  $1/f$ , where  $f$  is the control loop frequency. Note that  $\theta_{u,t}$  is not simply the instantaneous command  $u_t$  (or  $K_u(u_t)$ ), but rather a transformed and filtered version of that command that depends on the previous user goal and the state of the platform. Using  $\theta_{u,t}$ , we determine the inferred user goal location  $\mathbf{G}_u$  as

$$\mathbf{G}_u = (G_{u,x}, G_{u,y}) = (\rho \cos \theta_u, \rho \sin \theta_u), \quad (2)$$

<sup>1</sup>Unless otherwise specified, all quantities are represented in the robot’s local coordinate system.

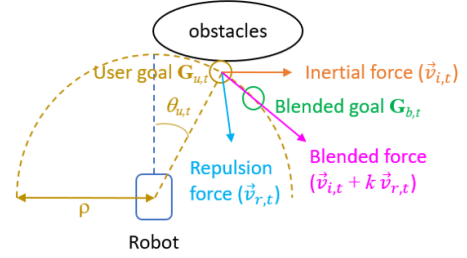


Figure 2: Depiction of relevant quantities for the proposed goal-blending shared autonomy (GBSA) system. The user’s current joystick angle is used to compute  $\theta_{u,t}$  (relative to the robot’s current heading, which is shown as a dotted blue line), which is used to infer the user’s goal  $\mathbf{G}_{u,t}$  (in dark yellow circle). The blended goal,  $\mathbf{G}_{b,t}$  (in green circle), is computed using  $\mathbf{G}_{u,t}$  and constraint forces  $\vec{v}_{r,t}$  (in light blue arrow) and  $\vec{v}_{i,t}$  (in orange arrow) known to the robot.

where  $\rho$  is a pre-specified parameter that denotes the distance at which local goals are to be placed (we use  $\rho = 2.0\text{m}$  in our experiments).

**Blended goal inference** Due to local obstacles,  $\mathbf{G}_u$  may not necessarily be a valid (i.e. collision-free) location for robot navigation, and so the second stage of the proposed goal inference and blending procedure is to use  $\mathbf{G}_u$  and constraint information from the robot to compute a *blended goal*  $\mathbf{G}_b$ . To perform this computation, we employ a physics-inspired method involving an *inertial force*, which represents a temporal smoothness constraint, and a *repulsion force*, which represents constraints induced by nearby obstacles.

The inertial force,  $\vec{v}_i$ , is calculated using the history of user goals, and is designed to be used in such a way that the blended goal is consistent with the recent history of inferred user goals. Let  $\xi_{u,t} = [\mathbf{G}_{u,t-T+1}, \dots, \mathbf{G}_{u,t}]$  denote the  $T$  most recent inferred user goal positions, all expressed with respect to the robot’s local coordinate system at time  $t$ , and let  $\bar{\mathbf{G}}_{u,t} = f(\xi_{u,t})$  represent a filtered goal computed over those positions. We compute the inertial force  $\vec{v}_i$  according to:

$$\vec{v}_{i,t} = \bar{\mathbf{G}}_{u,t} - \bar{\mathbf{G}}_{u,t-1}, \quad (3)$$

In our experiments, we select  $T$  such that  $\xi_{u,t}$  corresponds to a time window of one second, and we use an  $f(\cdot)$  corresponding to the sample mean. If the time window of the historical goal locations is set too short, the computed inertial force would be subject to larger noise due to numerical variance between consecutive goal positions. If it is set too long, the inertial force would not follow the changes of user goals quickly.

The repulsion force,  $\vec{v}_r$ , is calculated using information from the local costmap, and is designed to be used such that the blended goal will be pushed away from obstacles and toward collision-free positions. More specifically,  $\vec{v}_r$  is the spatial gradient of the local costmap computed at user goal location  $\mathbf{G}_u$ , i.e., it points in the direction of the steepest

---

**Algorithm 1** Overall Flow of GBSA.

---

- 1: Initialize a buffer for  $T$  most recent user goals,  $\mathbf{G}_u$
  - 2: Initialize a buffer for  $T$  most recent blended goals,  $\mathbf{G}_b$
  - 3: **for each**  $t$  **do**
  - 4:   Obtain a user command,  $u_t$
  - 5:   Compute a costmap,  $C$ , from obstacle detections
  - 6:    $\theta_{u,t} \leftarrow \theta_{u,t-1} + K_u(u_t) \cdot \Delta t - \omega_v \Delta t$     $\triangleright$  Eq. (1)
  - 7:    $\mathbf{G}_{u,t} \leftarrow (\rho \cos \theta_{u,t}, \rho \sin \theta_{u,t})$     $\triangleright$  Eq. (2)
  - 8:    $\overline{\mathbf{G}}_{u,t} \leftarrow f([\mathbf{G}_{u,t-T+1}, \dots, \mathbf{G}_{u,t}])$
  - 9:    $\vec{v}_{i,t} \leftarrow \overline{\mathbf{G}}_{u,t} - \overline{\mathbf{G}}_{u,t-1}$     $\triangleright$  Eq. (3)
  - 10:    $\vec{v}_{r,t} \leftarrow$  spatial gradient of  $C(\mathbf{G}_{u,t})$
  - 11:    $\mathbf{G}_{b,t} \leftarrow \arg \min_l C(\mathbf{G}_{u,t} + \frac{l(\vec{v}_{i,t} + k \vec{v}_{r,t})}{|\vec{v}_{i,t} + k \vec{v}_{r,t}|})$     $\triangleright$  Eq. (4)
  - 12:    $\mathbf{G}_{b,t}^* \leftarrow \text{OPTIMIZEGB}(\mathbf{G}_{u,t}, \mathbf{G}_{b,t})$     $\triangleright$  Section 4.3
  - 13:    $(v_r, \omega_r) \leftarrow \text{MOTIONPLANNER}(\mathbf{G}_{b,t}^*, C)$
  - 14: **end for**
- 

slope in the costmap. If  $\mathbf{G}_u$  lies inside an obstacle where there is no gradient in the costmap, we instead compute the gradient of the costmap at the edge of the obstacle where the gradient vector is non-zero.

Using the specified inertial and repulsion forces, the blended goal  $\mathbf{G}_{b,t}$  is computed as:

$$\mathbf{G}_{b,t} = \arg \min_l C(\mathbf{G}_{u,t} + \frac{l(\vec{v}_{i,t} + k \vec{v}_{r,t})}{|\vec{v}_{i,t} + k \vec{v}_{r,t}|}) \quad (4)$$

$s.t. 0 \leq l < L,$

where both  $\vec{v}_{i,t}$  and  $\vec{v}_{r,t}$  are normalized to unit vectors for the computation above,  $k$  is a user-specified scalar that determines the relative strength between the inertial force and the repulsion force, and  $C(\mathbf{x})$  is the local costmap value at position  $\mathbf{x}$ . We set  $k = 1.1$  based on our desire for blended goals to be located in front of obstacles, which will happen if the repulsion force from the obstacle is weighted larger than the user-goal inertial force. The projection length,  $l$ , is a scalar between 0 and  $L$ , where  $L$  is the maximum distance that the blended goal can deviate from the user goal.

### 3.2 Overall System

Having defined how the local motion planner goals are computed from the human steering commands, we can now describe the overall GBSA system. The system includes four main functional blocks: the user interface, a perception component, the goal inference and blending block, and a motion planner. The user interface samples the driver steering commands periodically to form a sequence of driving commands. The perception block builds a local costmap based on the robot's sensor data. The goal inference and blending block, as illustrated in Section 3.1, infers a user goal from the sequence of driving commands and blends it with the robot's obstacle avoidance objective to determine a local goal. The motion planner takes the local goal and the costmap to compute a motion command to drive the vehicle. This may include a local path planning algorithm and a motion controller that computes linear and angular velocity command  $(v_r, \omega_r)$  to control the robot motors in a typi-

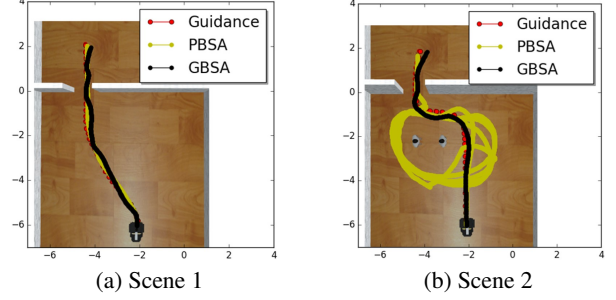


Figure 3: Representative wheelchair trajectories from users using both the PBSA system and our proposed GBSA system in (a) Scene 1 and (b) Scene 2, respectively. In Scene 2, the PBSA user failed to drive through the doorway and had to retry many times, whereas, in the GBSA experiment, the same user succeeded in driving through the doorway in her first attempt.

cal mobile robot implementation. Algorithm 1 illustrates the overall flow of GBSA.

## 4 Human Experiments

We hypothesize that, by performing blending of human and robot input at the level of local motion goals, GBSA will be more responsive than other shared autonomy systems without sacrificing safety. To test this hypothesis, we conducted a human study in a simulated wheelchair where the human and the robot attempt to collaboratively drive the wheelchair to accomplish a set of driving tasks. We compared GBSA to a state-of-the-art shared control system (Urdiales et al. 2011; Poncela et al. 2009) for settings in which (a) the user's intended destination is not given *a priori* and there are no pre-specified goal candidates, (b) the human policy is unknown, and (c) in the absence of human input, the robot drives autonomously by attempting to drive straight while avoiding collisions. We refer to the latter system as policy blending shared autonomy (PBSA). To the best of our knowledge, PBSA is the most appropriate baseline that can work in scenarios adhering to (a)-(c). We did not compare against task-level shared control systems here because we are interested in techniques that continually incorporate human input—task-level techniques continually monitor the state and prescribe *either* full human control or full robot control.

The users operated and compared the GBSA and PBSA systems in two scenarios: (1) *Scene 1*: driving through a doorway without obstacles (Fig. 3a), and (2) *Scene 2*: driving through a doorway in a cluttered environment (e.g., two static pedestrians near the door, as shown in Fig. 3b). As an experimental mechanism, we requested that participants follow as closely as possible a guidance path that we drew on the ground but was unknown to the autonomous system, and we assumed that participants complied with this request. Providing a fixed guidance path to each participant allows us to fairly compare results across different participants. In this setting, we quantify responsiveness of the shared autonomy system by measuring the deviation of the vehicle trajectory from the guidance path. The user workload, efficiency, and

safety are characterized by the intervention time, travel time, total number of backups, and total number of collisions, respectively (refer to Section 4.2 for definitions). All of the metrics were compared between the PBSA and GBSA systems in which the robots in both systems had no knowledge about the guidance path.

#### 4.1 Experimental Setup

Apart from the modules for human and robot goal blending, the rest of PBSA and GBSA systems were identical, including modules for perception, localization, user interface, user command sampling, path planning, and the robot’s motor controller. The simulated wheelchair is equipped with a simulated SICK LMS111 laser scanner and basic odometry sensors for robot perception and localization. The local path planning module uses the elastic band local path planner (EBand) (Quinlan and Khatib 1993) operating at a rate of approximately 20 Hz. For GBSA, the local goal locations are provided by the goal-blending algorithm described in Section 3.1, and the motion command computed from the planner directly goes to the robot’s motor controller. For PBSA, the local goal locations are set 2 meters ahead of the robot (Urdiales et al. 2011), and the motion command computed from the planner is the robot command that is then arbitrated with the human command to determine a *blended* motion command for the robot’s motor controller. In our system, the robot’s motor controller is a differential-drive velocity controller.

The user interface is a standard gamepad-like controller that allows the human participants to provide control signals: steering angle, start, brake, and backup. The participants can move the joystick forward to start the robot, and then the wheelchair would drive forward autonomously (in shared autonomy mode) even if the joystick is released. The participants can communicate the steering angle by moving the 2D joystick to any angle between [90, -90] degrees. The steering angle signals are sampled at a fixed rate of 20Hz to create the user steering command  $u_t$  for the shared autonomy algorithms. The participants can stop the wheelchair by pulling the joystick backward (i.e., giving a brake command). They can back up the wheelchair manually by pressing a button to set the reverse gear and then pulling the joystick backward. To resume forward motion, they can press another button and the wheelchair would drive in shared autonomy mode again.

We asked our human participants to follow, as closely as possible, the guidance paths marked in red dots on the ground in two scenes. In Scene 1, the marked path guides the wheelchair user to turn the wheelchair well before the doorway (Fig. 3a), whereas in Scene 2 the marked path guides the wheelchair user to turn the wheelchair just in front of the doorway (Fig. 3b). The guidance path is collision-free, and it is regarded as the user’s ground truth intention in the experiments. When the participants find it difficult to move the robot along the guidance path, they may deviate from the guidance path temporarily or back the robot up to correct its path. Before the experiment begins, the participants have about one minute of practice for each system. They can move the wheelchair around the environment freely.

During this phase they can (and often do) test whether the wheelchair collides with an obstacle if they deliberately try to make it do so. We recorded the user commands and the wheelchair trajectories in the experiment in order to compute the performance metrics defined in Section 4.2 for analyzing both systems. After the experiments were done, the participants were given a questionnaire to rate their experiences with both systems.<sup>2</sup>

The participants comprised 15 male and 5 female students of ages ranging from 20 to 35 years old. None of the participants had prior experience with the system. The participants were told which system they were going to use before each driving task. In order to avoid sequential bias, half of the participants were asked to pilot the PBSA system first, and the other half were asked to pilot the GBSA system first.

#### 4.2 Performance Evaluation Metrics

The driving data collected during these experiments allow us to quantitatively measure the responsiveness of the shared autonomy systems. The metrics include: (1) the Hausdorff distance between the user’s actual path (i.e., wheelchair trajectory) and the guidance path, (2) overall travel time, (3) intervention time, (4) total number of backups, and (5) total number of collisions.

**Hausdorff distance** measures how far two subsets of data are from each other. In our experiment, we calculate the Hausdorff distance between the user’s actual path (i.e., wheelchair trajectory) and the guidance path. Formally, the Hausdorff distance is defined as

$$h(A, B) = \max_{a \in A} \{ \min_{b \in B} \{ d(a, b) \} \} \quad (5)$$

where  $a$  and  $b$  are points on the user’s actual trajectory  $A$  and the guidance path  $B$ , respectively, and  $d(a, b)$  is the Euclidean distance between points  $a$  and  $b$ . Ideally, the Hausdorff distance between the guidance path and the user’s actual trajectory is low.

**Travel time** is the total time required to complete the driving task.

**Intervention time** is defined as in Cooperstock *et al.* (Cooperstock et al. 2007). Specifically, it is calculated as the percentage of non-zero user steering commands issued during the driving task. The lower this metric, the more we can say that the user finds the system’s behavior acceptable.

**Total number of backups** is recorded as the total number of times that the user had to shift into reverse in order to move away from obstacles. Ideally, this metric is as low as possible, indicating that the system did not get stuck.

Finally, participants were also asked to provide their degree of agreement with several subjective statements about each system. The statements include: (a) “The wheelchair goes to my intention correctly,” (b) “The wheelchair was responsive to my commands,” (c) “I felt confident that the vehicle would not collide with obstacles,” and (d) “I prefer this system of driving over the other system.” Ratings of each statement were collected on a Likert scale of five responses: strongly disagree (1), disagree (2), neither agree

<sup>2</sup>The experimental paradigm was reviewed and approved by our university’s institutional review board.

nor disagree (3), agree (4), and strongly agree (5). The average answers of all the participants are calculated for each survey question.

### 4.3 Implementation Details

We implemented GBSA in ROS for our experiments. Our implementation leverages many standard modules available in the ROS repositories, with custom modules for computing the costmap and, of course, the local navigation goals that the robot should use.

**Costmap computation** The costmap computed for the perception block is a local 2D map cetered at the robot, where a lethal cost is placed in cell locations that the Lidar indicates to be an obstacle.

**Blended goal refinement** Computing the blended goal  $\mathbf{G}_b$  via Eq. 4 provides a reasonable blended local goal for the robot in most cases, but the goal position may still exhibit instability over time due to the complexity of the environment. To provide a smoother trajectory of blended goals, we further refine the blended goal positions using a sample-based technique that employs the method described by Jiang *et al.* (Jiang, Warnell, and Stone 2018). At each time-step, we create a set of  $N$  samples as shifted user goals,  $\mathbf{G}'_{u,t} \in \mathcal{G}$ , each of which is spatially offset from the nominal user goal  $\mathbf{G}_{u,t}$  by a small amount. We then transform each of these samples into a corresponding blended goal  $\mathbf{G}'_{b,t}$  based on Eq. (4). Next, we find a refined blended goal  $\mathbf{G}^*_{b,t}$  by trying to 1) minimize the distance between the nominal user goal  $\mathbf{G}_{u,t}$  and an optimized user goal  $\mathbf{G}^*_{u,t}$  for tracking the nominal user goal, and 2) minimize the deviation between the optimized blended goal  $\mathbf{G}^*_{b,t}$  and previous blended goal  $\mathbf{G}^*_{b,t-1}$  for maintaining temporal consistency over consecutive blended goals. The optimized user goal  $\mathbf{G}^*_{u,t}$  is computed as:

$$\begin{aligned} \mathbf{G}^*_{u,t} &= \arg \min_{\mathbf{G}'_{u,t} \in \mathcal{G}} \left( \sum w_1 |\Delta_x \mathbf{G}_u|^2 + \sum w_2 |\Delta_t \mathbf{G}_b|^2 \right) \\ &= \arg \min_{\mathbf{G}'_{u,t} \in \mathcal{G}} \left( \sum_{i=0}^T w_1 |\mathbf{G}'_{u,t-i} - \mathbf{G}_{u,t-i}|^2 \right. \\ &\quad \left. + \sum_{i=0}^{T-1} w_2 |\mathbf{G}'_{b,t-i} - \mathbf{G}'_{b,t-i-1}|^2 \right) \end{aligned} \quad (6)$$

where  $w_1$  and  $w_2$  are specified coefficients for the two deviation terms (empirically chosen as  $w_1 = 1$  and  $w_2 = 10$  in our experiments). The blended goal corresponding to the optimized user goal  $\mathbf{G}^*_{u,t}$  is used as the local goal position in the subsequent motion planning step. Algorithm 2 implements this blended goal optimization procedure.

### 4.4 Experimental Results

The results of our study validated our hypothesis, i.e., GBSA is a more responsive shared autonomy system than the PBSA system. By using the proposed shared goal setting concept, the shared autonomy navigation system can control

---

#### Algorithm 2 Blended Goal Optimization.

---

```

procedure OPTIMIZEGB( $\mathbf{G}_{u,t}, \mathbf{G}_{b,t}$ )
2:    $E_{min} \leftarrow \inf$ 
   Create a set of  $\mathbf{G}'_{u,t} \in \mathcal{G}$  from  $\mathbf{G}_{u,t}$ 
4:   for each  $\mathbf{G}'_{u,t} \in \mathcal{G}$  do
      $\vec{v}'_{r,t} \leftarrow \text{gradient of } C(\mathbf{G}'_{u,t})$ 
6:      $\mathbf{G}'_{b,t} \leftarrow \arg \min_l C(\mathbf{G}'_{u,t} + \frac{l(\vec{v}_{i,t} + k \vec{v}'_{r,t})}{|\vec{v}_{i,t} + k \vec{v}'_{r,t}|})$ 
      $E \leftarrow \sum w_1 |\Delta_x \mathbf{G}_u|^2 + \sum w_2 |\Delta_t \mathbf{G}_b|^2 \triangleright \text{Eq. (6)}$ 
8:     if  $E < E_{min}$  then
        $E_{min} \leftarrow E$ 
10:       $\mathbf{G}^*_{b,t} \leftarrow \mathbf{G}'_{b,t}$ 
     end if
12:   end for
   return  $\mathbf{G}^*_{b,t}$ 
14: end procedure

```

---

the vehicle to perform desired navigation behaviors in response to user-supplied control commands in complex environments (e.g., Fig. 3b). The results showed GBSA outperformed PBSA in responsiveness, efficiency, and workload reduction (Fig. 4), while maintaining collision-free operation. In Scene 1, both systems perform well for users following the guidance path, and all three metrics are comparable (Figs. 3a and 4). However, in Scene 2—which represents a more cluttered environment—the PBSA system failed to allow users to follow the guidance path whereas the GBSA system led to trajectories closer to the guidance path. The performance metrics for each system are shown in Fig. 4.

We additionally compared the two systems with an Oracle system that represents entirely autonomous control with access to the guidance path as the goals for local path planning in Fig. 4. In the ideal case, both the Oracle and a human following the guidance path will exhibit the exact same behavior. The Oracle system uses the same perception module and motion planner as those in GBSA and PBSA systems, except that the goals given to the motion planner are waypoints of the guidance path that are presented at about 2m ahead of the moving wheelchair. Note that even the Oracle system’s trajectory deviated from the guidance paths (i.e., the Hausdorff distance is not zero) because the local motion planner is not perfect for tracking curved paths.

Finally, in the post-experiment questionnaire, the participants selected one of the five Likert scale ratings for each system. On average, the users gave better ratings in all the questions for the GBSA system than they gave for the PBSA system (Fig. 5).

## 5 Discussion and Future Work

The proposed goal-blending scheme ensures that the shared autonomy system is responsive to users because user commands always result in goal changes, whereas user commands in typical servo-level shared control schemes (e.g., PBSA) often have to be ignored due to safety constraints.

GBSA performs almost exactly the same as PBSA in free space or simple scenarios such as Scene 1. In these scenarios, PBSA allocates high weights to the user commands be-

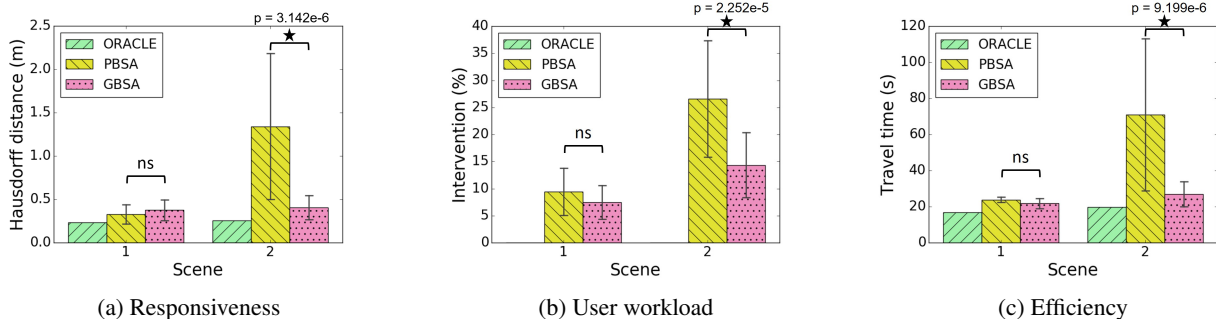


Figure 4: Performance comparison for the Oracle, PBSA, and GBSA systems for (a) Hausdorff distance for characterizing responsiveness, (b) intervention time for characterizing user workload, and (c) travel time for characterizing efficiency. The Oracle system is a fully autonomous system where the robot has access to the ground-truth guidance paths. The t-test results show that the GBSA system outperforms the PBSA system significantly in the cluttered environment ( $p \leq 0.05$ ).

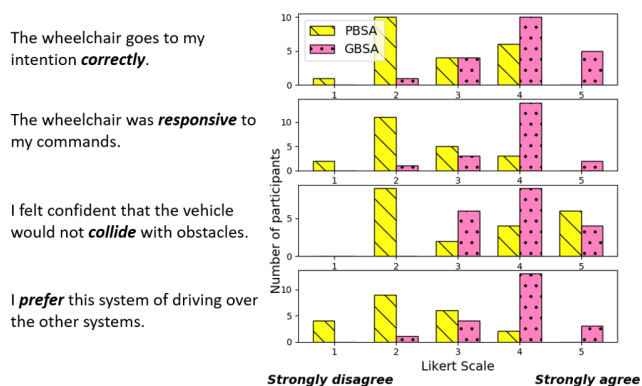


Figure 5: Survey response results for the questionnaire given to participants. On average, GBSA received better ratings (Agree) for all questions than PBSA did (Disagree for responsiveness and preference, and Neutral for correctness and safety).

cause the safety constraints are not a concern. The performance of GBSA is similar in these scenarios as long as the local path planner generates an optimized trajectory that resembles manual driving behaviors in free space. In cluttered scenarios such as Scene 2, however, PBSA may discard the user’s commands due to safety concerns. In this scenario, the user gives a command to turn right in front of the doorway, but the human command is effectively discarded because, according to the safety metrics defined in Urdiales *et al.* (Urdiales *et al.* 2011), the command is unsafe. Thus, the user effectively loses their control of the wheelchair (i.e., the system is not responsive), and the user cannot coerce the vehicle to closely follow the specified guidance path.

One interesting phenomenon that we observed was that almost all of the participants operating the PBSA system failed to drive through the doorway in their initial attempt in Scene 2, and they often needed to backup the wheelchair in order to complete the driving task, while only a few users operating the GBSA system needed to back up the wheelchair in the same scene. The average number of backups for the PBSA

system in Scene 2 is 2.29, versus 0.12 for the GBSA system. All the participants would have been able to pass the doorway without backing up the wheelchair with the forward-looking blending algorithm of GBSA—for those that performed backups, all of them are in the group that used the PBSA system first and had experienced a failure in Scene 2 before using the GBSA system. They all gave feedback, saying that they intervened early with GBSA in order to avoid a predicted failure.

One limitation of the proposed GBSA method is that it still requires a fairly high user workload. That is, given the dynamics of the environment, the user will likely have to provide commands fairly frequently. The user workload might be reduced further by trying to infer navigation goals at longer ranges (i.e., larger  $\rho$ ), but we conjecture that may subsequently reduce the responsiveness of the system.

To apply GBSA to applications with more environmental constraints, such as assistive driving for automotives, additional development will be necessary. For example, the user goal inference method needs to take into account how humans interact with different user interfaces (e.g., the throttle, the brake, or a turn signal switch), and the computation of blended goals needs to be subject to traffic rules and cultural factors. This paper opens up these topics as interesting and potentially fruitful avenues for further research towards making autonomous systems better able to work with people across a range of domains.

## 6 Conclusion

This paper introduced a novel goal-blending concept for shared autonomy navigation systems, called goal-blending shared autonomy (GBSA) to enable responsive control in cluttered environments. GBSA blends the user and robot input at the level of local navigation goals rather than at the level of motor commands. We showed experimentally that this new concept can lead to greater responsiveness to human commands in a cluttered environment compared to a state-of-the-art servo-level shared control paradigm, while maintaining similar performance for those scenarios that the existing paradigm already handled well.

## Acknowledgment

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CPS-1739964, IIS-1724157, NRI-1925082), the Office of Naval Research (N00014-18-2243), Future of Life Institute (RFP2-000), Army Research Office (W911NF-19-2-0333), DARPA, Lockheed Martin, General Motors, and Bosch. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

## References

- Anderson, S. J.; Karumanchi, S. B.; and Iagnemma, K. 2012. Constraint-based planning and control for safe, semi-autonomous operation of vehicles. In *2012 IEEE intelligent vehicles symposium*, 383–388. IEEE.
- Argall, B. D. 2015. Turning assistive machines into assistive robots. In *Quantum Sensing and Nanophotonic Devices XII*, volume 9370, 93701Y. International Society for Optics and Photonics.
- Argall, B. D. 2016. Modular and adaptive wheelchair automation. In *Experimental robotics*, 835–848. Springer.
- Bonarini, A.; Ceriani, S.; Fontana, G.; and Matteucci, M. 2012. Introducing LURCH: a shared autonomy robotic wheelchair with multimodal interfaces. In *Proceedings of IROS 2012 workshop on progress, challenges and future perspectives in navigation and manipulation assistance for robotic wheelchairs*.
- Broad, A.; Murphey, T.; and Argall, B. 2019. Highly parallelized data-driven MPC for minimal intervention shared control. *arXiv preprint arXiv:1906.02318*.
- Bruemmer, D. J.; Few, D. A.; Boring, R. L.; Marble, J. L.; Walton, M. C.; and Nielsen, C. W. 2005. Shared understanding for collaborative control. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 35(4): 494–504.
- Connell, J.; and Viola, P. 1990. Cooperative control of a semi-autonomous mobile robot. In *Proceedings., IEEE International Conference on Robotics and Automation*, 1118–1121. IEEE.
- Cooperstock, J.; Pineau, J.; Precup, D.; Atrash, A.; Jaulmes, R.; Kaplow, R.; Lin, N.; Prahacs, C.; Villemure, J.; and Yamani, H. 2007. Smartwheeler: A robotic wheelchair test-bed for investigating new models of human-robot interaction. In *Proc. of the IEEE Conference on Intell. Robots and Systems (IROS), San Diego, USA*.
- Crandall, J. W.; and Goodrich, M. A. 2002. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *IEEE/RSJ international conference on intelligent robots and systems*, volume 2, 1290–1295. IEEE.
- Demeester, E.; Hüntemann, A.; Vanhooydonck, D.; Vanacker, G.; Van Brussel, H.; and Nuttin, M. 2008. User-adapted plan recognition and user-adapted shared control: A Bayesian approach to semi-autonomous wheelchair driving. *Autonomous Robots* 24(2): 193–211.
- Dragan, A. D.; and Srinivasa, S. S. 2013. A policy-blending formalism for shared control. *The International Journal of Robotics Research* 32(7): 790–805.
- Erdogan, A.; and Argall, B. D. 2017. The effect of robotic wheelchair control paradigm and interface on user performance, effort and preference: an experimental assessment. *Robotics and Autonomous Systems* 94: 282–297.
- Erlie, S. M.; Fujita, S.; and Gerdes, J. C. 2013. Safe driving envelopes for shared control of ground vehicles. *IFAC Proceedings Volumes* 46(21): 831–836.
- Ghorbel, M.; Pineau, J.; Gourdeau, R.; Javdani, S.; and Srinivasa, S. 2018. A decision-theoretic approach for the collaborative control of a smart wheelchair. *International Journal of Social Robotics* 10(1): 131–145.
- Inigo-Blasco, P.; Diaz-del Rio, F.; Diaz, S. V.; and Muniz, D. C. 2014. The shared control dynamic window approach for non-holonomic semi-autonomous robots. In *ISR/Robotik 2014; 41st International Symposium on Robotics*, 1–6. VDE.
- Javdani, S.; Admoni, H.; Pellegrinelli, S.; Srinivasa, S. S.; and Bagnell, J. A. 2018. Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research* 37(7): 717–742.
- Jiang, Y.-S.; Warnell, G.; and Stone, P. 2018. DIPD: Gaze-based intention inference in dynamic environments. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.
- Li, Q.; Chen, W.; and Wang, J. 2011. Dynamic shared control for human-wheelchair cooperation. In *2011 IEEE International Conference on Robotics and Automation*, 4278–4283. IEEE.
- Maurer, S.; Erbach, R.; Kraiem, I.; Kuhnert, S.; Grimm, P.; and Rukzio, E. 2018. Designing a guardian angel: Giving an automated vehicle the possibility to override its driver. In *Proceedings of the 10th international conference on automotive user interfaces and interactive vehicular applications*, 341–350.
- Poncela, A.; Urdiales, C.; Pérez, E. J.; and Sandoval, F. 2009. A new efficiency-weighted strategy for continuous human/robot cooperation in navigation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 39(3): 486–500.
- Quinlan, S.; and Khatib, O. 1993. Elastic bands: Connecting path planning and control. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 802–807. IEEE.
- Rakita, D.; Mutlu, B.; Gleicher, M.; and Hiatt, L. M. 2018. Shared dynamic curves: A shared-control telemanipulation method for motor task training. In *Proceedings of the 2018*



*ACM/IEEE International Conference on Human-Robot Interaction*, 23–31.

Rakita, D.; Mutlu, B.; Gleicher, M.; and Hiatt, L. M. 2019. Shared control-based bimanual robot manipulation. *Science Robotics* 4(30).

Rao, R.; Conn, K.; Jung, S.-H.; Katupitiya, J.; Kientz, T.; Kumar, V.; Ostrowski, J.; Patel, S.; and Taylor, C. J. 2002. Human robot interaction: application to smart wheelchairs. In *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, volume 4, 3583–3588. IEEE.

Simpson, R. C.; Levine, S. P.; Bell, D. A.; Jaros, L. A.; Koren, Y.; and Borenstein, J. 1998. NavChair: an assistive wheelchair navigation system with automatic adaptation. In *Assistive technology and artificial intelligence*, 235–255. Springer.

Urdiales, C.; Peula, J. M.; Fdez-Carmona, M.; Barrué, C.; Pérez, E. J.; Sánchez-Tato, I.; Del Toro, J.; Galluppi, F.; Cortés, U.; Annichiarico, R.; et al. 2011. A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation. *Autonomous Robots* 30(2): 179–197.

Wang, H.; and Liu, X. P. 2014. Adaptive shared control for a novel mobile assistive robot. *IEEE/ASME Transactions on Mechatronics* 19(6): 1725–1736.

Xu, R.; Hartshorn, R.; Huard, R.; Irwin, J.; Johnson, K.; Nelson, G.; Campbell, J.; Ay, S. A.; and Taylor, M. E. 2016. Towards a semi-autonomous wheelchair for users with ALS. In *Proc. IJCAI Workshop on Autonomous Mobile Service Robots, New York City, USA*.