Follow The Robot: Modeling Coupled Human-Robot Dyads During Navigation*

Amal Nanavati[1], Xiang Zhi Tan[1], Joe Connolly[2], Aaron Steinfeld[1]

***Abstract*— Many robot applications being explored involve robots leading humans during navigation. Developing effective robots for this task requires a way for robots to understand and model a human's following behavior. In this paper, we present results from a user study of how humans follow a guide robot in the halls of an office building. We then present a data-driven Markovian model of this following behavior, and demonstrate its generalizability across time interval and trajectory length. Finally, we integrate the model into a global planner and run a simulation experiment to investigate the benefits of coupled human-robot planning. Our results suggest that the proposed model effectively predicts how humans follow a robot, and that the coupled planner, while taking longer, leads the human significantly closer to the target position.**

I. INTRODUCTION

With more robots being deployed in public, scenarios will arise where robots guide humans through complex indoor spaces (e.g., navigational aides for people who are blind, tour guides, and security escorts). In order to effectively plan for such tasks, robots must be aware of where the following human is – to drop them off in a pose convenient for reaching their final destination, to avoid running them into obstacles, and to ensure a smooth navigational experience. For example, it is possible to walk a person who is blind into a door frame or corner if the robotic guide does not know where the person is positioned relative to it. Therefore, such robots must understand how humans follow robots and incorporate that information into their planning system.

In this paper, we present a predictive model of how a human follows a robot for scenarios where the human holds onto the robot. This model was developed, optimized, and validated on data from a user study in which users held onto a mobile robot as it guided them inside an office building. In order to integrate into traditional global planners, this model is Markovian and sensor-free. We demonstrate the model's robustness and effectiveness at predicting following behavior. We also present results from a simulation experiment that demonstrates the benefits of integrating the model into a global planner for coupled human-robot planning. To the best of our knowledge, this is the first predictive model of how a human follows a robot while holding onto it.

*This work was funded by the National Science Foundation (CBET-1317989 & IIS-1734361) and the National Institute on Disability, Independent Living, and Rehabilitation Research (90DPGE0003).

[1]Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15289, USA {arnanava@alumni., zhi.tan@ri., steinfeld@}cmu.edu

[2]Department of Computer Science, Yale University, New Haven, CT 06520, USA joe.connolly@yale.edu

II. RELATED WORK

Many robots have been developed to guide humans through spaces. They have acted as navigational aides [1], [2], tour guides [3], [4], and shopping assistants [5], [6]. A recent survey [7] covers advancements in this area. However, many such navigation systems either do not account for the fact that a human is following or naively increase the robot's bounding box to prevent the human from colliding with obstacles [8]. One work investigates how humans follow a mobile robot when not holding on to it and reports that humans' following behavior depends on the curvature of the robot's trajectory and the robot's acceleration [9]. However, this work does not present a way to incorporate these insights into planning. Other works present methods to predict human trajectories [10], [11] and incorporate those predictions into a local planner [12]. However, these methods require sensors that update human position in-real-time, making them inadequate for applications in global planning. Further, they focus on how human trajectories evolve without external influences, and do not explicitly model how robot motion influences human motion. Finally, other works that focus on robot navigation in human spaces have studied how robots should follow or approach humans [13], [14], walk side-by-side with humans [15], or navigate around humans [16], [17].

Within global planning, there has been research into planning for tractor-trailer robots, where a coupled body passively follows the actively moving robot [18], [19]. These efforts use mathematical models to calculate how the trailer moves as a function of the tractor's movement. Such models are insufficient for our purpose because humans are agents that do not passively follow the robot with rigid form and motion. However, there is still merit in this analogy and we draw inspiration for our coupling model from these efforts.

III. DATA COLLECTION USER STUDY DESIGN

To gather data on how humans follow a robot, we ran an IRB-approved data collection user study ($n = 10$), with participants recruited through an online participation pool and by word-of-mouth in the Pittsburgh metropolitan area. Participants held onto the handle of a mobile robot as it guided them along trajectories through the hallways of an office building. Our mobile robot (Fig. 1 Left) was a Pioneer P3DX base with a SICK LMS 111-10100 laser scanner and a compliant adjustable handle (ranging from 0.96 m, setting 1, to 1.11 m, setting 6, off the ground) developed from past research [20]. The robot contained a static map of the office building and used the Adaptive Monte-Carlo Localization (AMCL) algorithm provided by ROS navigation to localize.

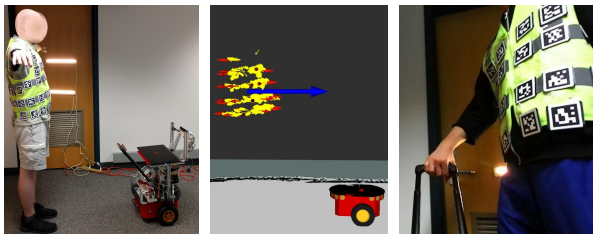

Fig. 1: **Left**: The robot circled the user during calibration, storing images of the vest. **Middle**: Processed calibration data. Yellow arrows are detected AprilTag poses over time, the blue arrow is the calculated torso pose, and red arrows are average poses per tag (to calculate static transforms). **Right**: A view from the robot's camera, showing how markerless methods would have difficulty detecting torso pose.

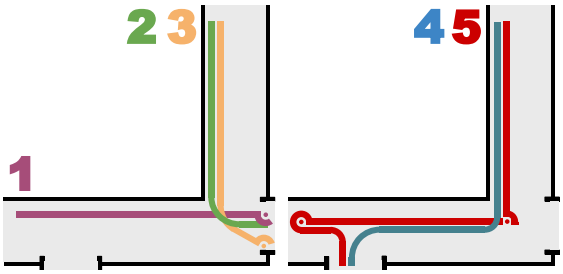

Fig. 2: The five pre-recorded trajectories every participant went on. Trajectory 6 (not shown) was over five times longer and was teleoperated by the experimenter.

The robot used a backwards-facing ZED stereo camera to track participants as they followed it. We opted for an on-board sensor strategy instead of mounting sensors in the environment in order to track users over longer trajectories. Finally, participants wore a vest with AprilTags [21] (48-52 tags, depending on the vest size) to enable robust person-tracking in twisted postures (e.g., Fig. 1 Right). The ZED camera and AprilTags were only used for data-collection, and are not required to utilize the model in planning.

The robot guided the participant along six trajectories in the hallways of an office building, each ending at a door. The first five trajectories (Fig. 2) were pre-recorded and played back during the study, and were inspired by a prior user study [22]. These trajectories took short paths (7 - 16m) and incorporated a variety of movements including gradual and in-place turns. The order of these trajectories for each participant was balanced with a Latin square to mitigate ordering effects. We then ran the sixth trajectory, which took a longer (80 - 90m) path that involved multiple rotations and varying robot speeds. Due to the complexity, this path was teleoperated by an experimenter from behind the participant.

After consenting to the research, participants selected their preferred vest size and were led to the hallway. We started with a calibration phase to ensure proper torso perception (Fig. 1 Left). We proceeded to a familiarization phase, where the participant experienced robot navigation under different handle height settings (1 - 6, with 1 being the shortest) and hand-holding configurations (right, left, or both) to find the most comfortable fit. The experimenter teleoperated the robot during this phase. Participants were asked to maintain their selected configuration for the remainder of the study. After familiarization, participants followed the robot along all six trajectories. For all trajectories, the experimenter walked behind and wirelessly issued small trajectory corrections when needed. If a trajectory did not complete (due to technical difficulties, an AprilTag falling off, passing pedestrians that impacted following behavior, etc.) the experimenter repeated it after the sixth trajectory. Afterwards, participants completed a semi-structured interview about the experience.

IV. DATA PROCESSING

After data collection, we processed the raw data into sequences of paired human and robot poses. To extract this information, we used the ArUco marker detection library [23] to convert raw image data per frame into a list of detected marker poses (x, y, z, roll, pitch, yaw) in $\mathbb{R}^3$. With the calibration data (Fig. 1 Middle), we calculated the participant's torso pose by averaging the position of all detected AprilTags and the orientation of a pre-selected group of tags at the front-center of the chest. We then calculated static transforms between each AprilTag's average pose and the participant's torso pose. With the trajectory data, we calculated human torso pose per frame by applying the static transforms from calibration to the detected AprilTag poses, using a rolling window of size 40 frames (0.67 secs) to prevent noise. This form of torso pose prediction was necessary because lighting conditions or particular robot turns sometimes caused only one tag to be visible in a frame.

With both calibration and trajectory data, we removed outliers that were $\geq d$ units away from $\geq p$ proportion of the points. We applied this procedure at multiple stages of data processing, separately for position and orientation. A manual analysis was conducted after data processing to ensure the calculated torso poses visually matched participant poses from recorded user study videos.

A. Creating the Dataset

Finally, we combined all trajectories into a unified dataset. First, for each trajectory we paired the human and robot poses that were closest in time and reduced the poses to three dimensions of interest: x, y, and θ (yaw). We then subsampled every trajectory so that the time interval between consecutive human-robot pose pairs was roughly $\Delta t := 0.2$ seconds. Then, we split the trajectories to a fixed number of seconds, $T := 15$, each. We took every possible T length sub-trajectory – the one from 0.0 seconds to 15.0 seconds, the one from 0.2 seconds to 15.2 seconds, 0.4 to 15.4, etc. Later in the paper, to investigate the robustness of the model to time intervals and trajectory length, we vary Δt and T.

V. DATA ANALYSIS

We had 7 female and 3 male participants, with average age 24.8 ($SD = 7.9$). 6 participants chose to hold the robot with their right hand, 3 with both hands, and 1 with their

left hand. 5 participants picked handle setting 1 (the lowest), 2 picked handle setting 2, 2 picked handle setting 5, and 1 picked handle setting 6. In the semi-structured interview, some participants explained that their handle height choices were based on day-to-day activities that were similar to the robot navigational experience, such as pushing a stroller or a shopping cart. Some participants liked holding onto the robot because it freed their attention and allowed them to look at their surroundings. Others would have preferred to follow the robot visually instead of holding on, due to jerkiness and a lack of predictability in the robot's motion.

We observed three common participant behaviors. First, there was a notable lag between robot motion and human motion. When the robot started moving or began a turn, participants stayed in-place briefly, extending their arm before following the robot. This lag was most evident on turns, where participants first rotated just their arm before moving their whole body to follow the robot. Second, while the robot was in motion, participants actively compensated by moving towards a more comfortable holding position. For example, participants would move to a position directly behind the robot even after the robot stopped. Third, we noticed that during turns, participants often did not face the direction of the robot but rather the tangent direction of the turn (Fig. 3 Bottom). These factors became guiding principles as we developed models of human following behavior.

We conducted a preliminary analysis to understand underlying trends, using the Dynamic Time Warp algorithm [24] to align trajectories across participants. Table I shows the average and standard deviation of human displacement from the robot, for every pair of corresponding human-robot poses. As can be seen, there was a large variability in participant following behavior. The bias to situate to the left of the robot could be due to hand-holding preference (the 6 right-holding participants had a greater average perpendicular displacement of 0.14 m) or because the robot moved on the right side of the hallways to align with American norms. Furthermore, the participants' 0.89 m average parallel distance from the robot could be related to handle height, as the 5 participants who set the handle setting to 1 had a smaller average parallel displacement of 0.85 m. Fig. 3 Top illustrates this variability. Note that not all this variability is due to individual factors, as robot trajectories varied slightly across participants.

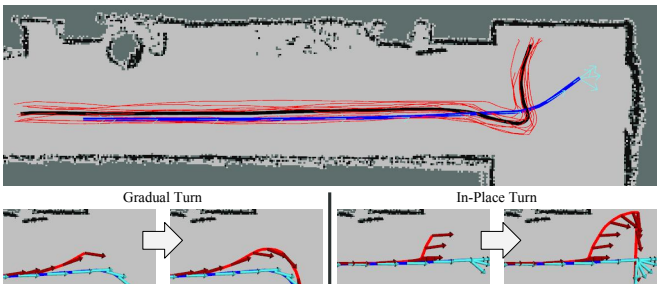

Fig. 3: **Top**: The average robot (dark blue) and human (black) paths for Trajectory 3, and each participant's path (red). **Bottom**: Participants swung out less for gradual turns (**Left**, Trajectory 4) than for in-place turns (**Right**, Trajectory 5). In both, the participant's torso angle (red arrow) turned somewhat towards the tangent direction of the turn (light blue).

TABLE I: Average Human-to-Robot Displacement*

Parallel (m)	Perpendicular (m)	Orientation (rad)
0.891 (0.083)	0.085 (0.091)	-0.05 (0.218)

*Values are: Avg (StdDev). Away from the robot, to the left of the robot, and counterclockwise are positive for the three columns respectively.

VI. MODEL FITTING

We compared variants of a baseline coupling model as well a geometric coupling model. In this section, we detail the model requirements, the models we trained, and the process of training and evaluating the models.

A. Model Requirements

The ultimate goal of the coupling model is to take a stream of robot poses – generated by a global planner – and output a corresponding sequence of predicted human poses. Since this coupling model has to integrate into existing planners, it must follow the Markov property – that a human predicted pose at time t only depends on the robot pose and human predicted pose from time $t-1$. Given these requirements, the model must have two capabilities. *(1) Initial Pose Prediction* uses a single robot pose as input and produces a *predicted* human pose as output. *(2) Next Pose Prediction* uses an old robot pose, an old human *predicted* pose, and a new robot pose as input, and produces a new human *predicted* pose as output. Since global planners do not allow for intermediate sensing of the human until execution, our model predicts human poses solely from robot poses.

B. Models

The baseline model, `Baseline` (`Base`) predicts that the human is always directly behind the robot, at a distance ℓ away, facing the robot. Since `Base` does not account for the aforementioned tendency for participants to turn towards the tangent direction of a turn, we developed `BaselineAngleOfMotion` (`BaseAOM`). `BaseAOM` predicts that the human faces a weighted average of the angle of motion (the angle between the old and new predicted human poses) and the angle to the robot (the angle between the new predicted human pose and the robot), with weight α.

Neither baseline model accounts for the aforementioned lag in following behavior, nor the range of angular and linear configurations the human could make with the robot due to the human arm's compliance. Therefore, we developed a more sophisticated geometric model, `Geometric` (`Geo`), which constructs a region within which the human can comfortably hold onto the robot and assumes the human will move the minimal distance to remain in that region. This region (Fig. 4) is defined by *centerOffset*, *regionOffset*, *regionSize*, and *regionAngle*. When predicting pose at time t, the model constructs the new region based on the new robot pose. It then predicts the new human position to be the

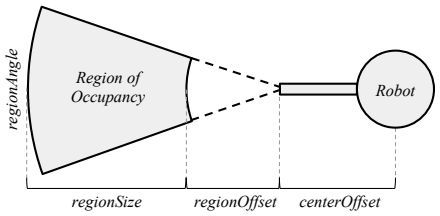

Fig. 4: The geometric model's position prediction, which creates a region of occupancy around the robot and assumes that the human moves the minimal distance such that they still remain within this region.

closest point in the new region to the old human position. We tested three variants of this model, differing in orientation prediction. `Geo` predicts that the human always faces the center of the region's arcs. `GeometricAngleOfMotion` (`GeoAOM`) predicts that the human faces a weighted average between the angle of motion and the angle to the center of the arcs, with weight α. However, neither model accounts for the aforementioned observation that while the robot is pulling the human, the human is actively compensating by moving towards a more comfortable location. Therefore, we developed `GeometricCompensation` (`GeoC`), which predicts the human will move to a weighted average (weight β) of the "pull pose" (predicted by `GeoAOM`) and the "compensation pose" (directly in the center of the region, facing the robot). All models predict initial pose by assuming the human is a fixed distance from the robot (in the middle of the region for geometric models), facing the robot.

In summary, our proposed model, `GeoC`, has 6 parameters: *centerOffset*, *regionOffset*, *regionSize*, *regionAngle*, α, and β. Its update function takes in a new robot position and an old predicted human position. It first transforms the region of occupancy to be centered around the new robot pose, given the fixed *centerOffset*, *regionOffset*, *regionSize*, *regionAngle* (Fig. 4). It then finds the point in the region that is closest to the old human pose, and sets that as the position of the "pull pose." Next, it gets the angle from the old human position to the new human position and the angle from the human to the robot, and sets the orientation of the "pull pose" to a weighted average of them (weight α). It then calculates the "compensation pose," which is directly in the middle of the region of occupancy, facing the robot. Finally, it sets the final human pose to a weighted average of the "pull pose" and "compensation pose" (weight β). Note that all the formerly-mentioned models are degenerate cases of `GeoC`. For example, `GeoC` becomes `Base` with the hardcoded parameters of $\alpha := 1.0$, $\beta := 0.0$, *centerOffset* + *regionOffset* $:= \ell$, and *regionAngle* $:= 0$. Although this guarantees that the latterly-mentioned models will outperform the formerly-mentioned ones, we test them across various metrics to understand each addition's contribution to predictive capabilities.

C. Model Training

All our models were trained using the DIRECT optimization algorithm [25], a deterministic algorithm which iteratively trisects parts of the parameter space until it reaches the termination criteria. The DIRECT algorithm is guaranteed to eventually find a global optima on any continuous real-valued function with bounded first-derivatives. To optimize the model, we calculate two errors. The orientation error e_θ calculates the average root mean squared error between predicted and actual human orientation, and the position error e_{xy} calculates average root mean squared error between predicted and actual human position.

$$e_\theta := \frac{1}{\mid D \mid} \sum_{\tau \in D} \sqrt{\frac{\sum\limits_{t \in \tau} \texttt{angleDist}(\hat{h}^t_\theta(r^{t-1}, \hat{h}^{t-1}, r^t), h^t_\theta)^2}{\mid \tau \mid}}$$

$$e_{xy} := \frac{1}{\mid D \mid} \sum_{\tau \in D} \sqrt{\frac{\sum\limits_{t \in \tau} \texttt{dist}(\hat{h}^t_{xy}(r^{t-1}, \hat{h}^{t-1}, r^t), h^t_{xy})^2}{\mid \tau \mid}}$$

In the above equations, D is the dataset composed of trajectories τ, `angleDist` calculates the minimum distance between two angles, and `dist` calculates the euclidean distance between two (x, y) points. r^t and h^t refer to the robot and human's actual poses at time t. h^t_{xy} and h^t_θ refer to the (x, y) position and yaw angle of the human at time t. $\hat{h}^t$ refers to the model's prediction function, which returns a human pose at time t. We combine these error functions into a joint error:

$$e := C_1 \cdot e_{xy} + \frac{C_2}{1 + C_3 \cdot e_{xy}} \cdot e_\theta$$

where C_1, C_2, and C_3 are positive constants. We chose this error function because, in coupled motion, human orientation is constrained by human and robot position. The human will generally face towards the robot as they follow the robot, with some deviations for situations like turns. Therefore, we weight e_θ by the inverse of e_{xy}, to prioritize minimizing position error before minimizing orientation error. For this study, we use $C_1, C_2, C_2 := 1.0$. We run DIRECT optimization for $200 \cdot P$ function evaluations, where P is the number of parameters in the model being trained.

D. Model Evaluation

We evaluated our model using k-fold cross validation, in which the complete dataset is partitioned into k equal-sized sets and we separately train on every subset of $k-1$ partitions and test on the held-out partition. Importantly, we partition the overall trajectories and *then* divide them into T length sub-trajectories; this ensures that sub-segments of the same overall trajectory do not appear in both the training and test sets. All models were trained and tested on the same random partitions. We then computed average performance statistics across the k iterations. Since [9] found that humans' following behavior depends on the curvature of the robot's trajectories, we separated trajectories in the dataset into "straight" or "curved" trajectories. We then ensured that each random partition had a proportional distribution of each trajectory type. A trajectory was categorized as "curved" if

its curvature was above a fixed $\kappa_{threshhold}$. The curvature of a trajectory was calculated by averaging the curvature at every point along the trajectory, given by

$$\kappa(x, y) = \frac{\mid x'y'' - y'x'' \mid}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

where derivatives are taken with respect to time. For this study, we use $\kappa_{thresh} := 0.40$.

For a single train-test cycle, we computed the joint error e on the test set, as well as the position error e_{xy} and orientation error e_θ on: (1) the whole test set; (2) just the "straight" subset of the test set; and (3) just the "curved" subset of the test set. Across the k train-test cycles within one round of cross validation, we report the average and standard deviation of these values. For this study, we used $k := 10$ partitions. To eliminate randomness, we repeated the cross validation procedure 5 times and averaged the aforementioned statistics across the repeats. We also conducted a repeated measure one-way ANOVA to determine whether the error differences across model pairs were significant[1].

VII. RESULTS

Table II shows the results for the aforementioned models. The move from baseline to geometric models improved position prediction, most of which was on curved trajectories. This validates our intuition that the geometric models would be better at modeling the lag in following behaviors, which was most evident during turns. Further, the move from baseline to geometric models also greatly improved orientation prediction. This illustrates the aforementioned dependency between position error and orientation error; since the human and robot are coupled, accurate orientation prediction is contingent upon accurate position prediction.

Moving from regular orientation prediction to angle of motion prediction increased orientation prediction, mostly for curved trajectories. This also validates our intuition that angle of motion prediction models the participants' tendency to orient towards the tangent direction of the turn. Further, angle of motion prediction improved `Base` much more than `Geo`. This is likely because much of the improvement in orientation prediction had already been obtained in the change from baseline to geometric, so angle of motion prediction had less room for additional improvements.

Finally, `GeoC` outperformed every model on every metric, and significantly outperformed every model on every metric except for `GeoAOM` on orientation error. This validates our intuition that `GeoC` models the fact that humans actively compensate as the robot is moving, a tendency that primarily affects position but also orientation. Further, not only is `GeoC` the best, it is also able to predict human position to within 0.156 m and human orientation to within 0.24 rad, an accuracy that should be sufficient for most planning and obstacle avoidance tasks.

In the subsequent sections, we probe `GeoC`'s applicability to existing planners by modifying factors that are known to vary across planners. Specifically, we vary Δt, the time interval, and T, the trajectory length, in the train and test data to understand how `GeoC` would function in planners that use a different simulation time or plan for different amounts of time than the model was trained on.

[1]The repeated factor was the train-test set and the fixed factor was the model. Pairwise comparison was done with Tukey HSD.

A. Generalizability Across Time Intervals

Due to `GeoC`'s compensation per timestep, the model implicitly has a dependency on time. Specifically, the more frequently it is called, the closer the predicted human pose will move to the "compensation pose". Therefore, the performance of the model may change if incorporated into a planner with a different step size than it was trained on. To investigate this dependency, we created two additional datasets, with time interval $\Delta t := 1, 2$ seconds respectively. Table III and IV shows the position error and orientation error of `GeoC` trained and tested on each of the three datasets (maintaining the same partitions across all train-test pairs). Although every pairwise difference across test time intervals (for a fixed train time interval) was significant, the errors across a single train time interval were small, within 0.015 m and 0.031 rad. Therefore, while `GeoC`'s training procedure works for situations where data is captured at different time intervals, the fact that there are significant differences and that each model performing best on the same time interval demonstrates a learned dependency on time. In future work it may be useful to develop a model that explicitly incorporates time; however, such a model may be harder to integrate into existing planners that do not simulate movement time.

B. Generalizability Across Trajectory Length

As different planners may plan a different number of steps into the future, our model must be able to generalize to situations where it is applied on a trajectory of different length than it was trained on. To investigate how well a trained instance of `GeoC` generalizes to previously unseen trajectory lengths, we trained and tested `GeoC` on trajectory lengths $T := 2.5, 7.5, 15.0$. Table V and VI shows the position and rotation error of `GeoC` trained and tested on each of the three datasets (maintaining the same partitions across all train-test pairs). For position error, there was only a significant difference in the $T := 2.5$ training length, where the $T := 2.5$ test length was significantly better. For rotation error, the $T := 15$ test length was significantly worse than all others across all train lengths. Despite these significant differences, the maximum difference across error averages within a single train length was 0.01 m for position error and 0.005 rad for rotation errors, which is nearly imperceptible by humans. These results demonstrate that the model is robust to trajectory length, which enables it to effectively integrate into a variety of global planners.

VIII. PLANNING WITH THE COUPLING MODEL

Given that `GeoC` is our best and most generalizable coupling model, we created a simulation experiment to investigate the value of integrating it into a global planner. Specifically, we compared two global planners:

TABLE II: Model Comparison On The Complete Dataset*

Model Name	Joint Error, e	Position Error, e_xy (m)			Orientation Error, e_θ (rad)		
		All	Straight	Curved	All	Straight	Curved
Base	0.435† (0.080)	0.176† (0.041)	0.162† (0.042)	0.186† (0.041)	0.305† (0.076)	0.231† (0.070)	0.360† (0.075)
BaseAOM	0.392† (0.076)	0.176† (0.041)	0.162† (0.042)	0.186† (0.041)	0.254† (0.071)	0.215† (0.074)	0.283† (0.065)
Geo	0.368† (0.063)	0.159† (0.035)	0.157† (0.041)	0.161† (0.032)	0.242† (0.065)	0.209† (0.073)	0.269† (0.059)
GeoAOM	0.365† (0.067)	0.162† (0.035)	0.158† (0.041)	0.164† (0.032)	0.235† (0.067)	0.206† (0.075)	0.260† (0.061)
GeoC	**0.357 (0.072)**	**0.156 (0.035)**	**0.156 (0.040)**	**0.156 (0.034)**	**0.233 (0.072)**	**0.204 (0.011)**	**0.255 (0.068)**

*5 repeats of 10-fold cross validation. Values are: Avg (StdDev). † represents a significant difference with *GeoC* (controlled and $p < 0.05$)

TABLE III: Position Error of GeoC Across Time Intervals

		Test Time Interval (sec)		
		0.2	1.0	2.0
Train Time Interval (sec)	0.2	0.156 (0.037)	0.164 (0.034)	0.171 (0.033)
	1.0	0.160 (0.042)	0.158 (0.038)	0.161 (0.037)
	2.0	0.162 (0.042)	0.158 (0.039)	0.160 (0.039)

TABLE IV: Rotation Error of GeoC Across Time Intervals

		Test Time Interval (sec)		
		0.2	1.0	2.0
Train Time Interval (sec)	0.2	0.232 (0.075)	0.238 (0.067)	0.249 (0.065)
	1.0	0.259 (0.080)	0.232 (0.072)	0.241 (0.06)
	2.0	0.267 (0.085)	0.236 (0.077)	0.242 (0.072)

TABLE V: Position Error of GeoC Across Length

		Test Traj Length (sec)		
		2.5	7.5	15.0
Train Traj Length (sec)	2.5	0.159 (0.029)	0.160 (0.031)	0.160 (0.035)
	7.5	0.157 (0.030)	0.158 (0.030)	0.158 (0.030)
	15.0	0.158 (0.030)	0.159 (0.031)	0.159 (0.035)

TABLE VI: Rotation Error of GeoC Across Length

		Test Traj Length (sec)		
		2.5	7.5	15.0
Train Traj Length (sec)	2.5	0.231 (0.057)	0.229 (0.061)	0.234 (0.069)
	7.5	0.230 (0.057)	0.228 (0.062)	0.233 (0.070)
	15.0	0.231 (0.057)	0.229 (0.061)	0.234 (0.070)

RobotOnlyPlanner plans in robot space and only uses the coupling model to compute the robot's goal pose, while CoupledPlanner plans in human and robot space by using GeoC to calculate where the human likely moves at each step of robot motion. We then simulated the robot guiding a human using each planner and tested multiple criteria related to human pose and distance to obstacles to compare the planners. For this experiment, we trained GeoC on all the data for $\Delta t := 0.2$ and $T := 15$ secs. This resulted in the optimized parameters *centerOffset* = 0.241 m, *regionOffset* = 0.421 m, *regionSize* = 0.269 m, *regionAngle* = 0.696 rad, $\alpha = 0.643$, and $\beta = 0.802$.

A. Experiment Setup

1) Planners: Both planners were given human goal poses. RobotOnlyPlanner used the coupling model to reverse-engineer the robot's goals from the human goals and then planned solely in the robot's space. CoupledPlanner, on the other hand, incorporated the coupling model throughout planning and planned in both the human and robot's space to get the human to the target *human* goal. Both planners used A* graph search, where the nodes consisted of robot pose (r_x, r_y, r_θ) and velocity (v, ω), and the edges simulated robot trajectories under feasible velocities given pre-defined acceleration limits. CoupledPlanner additionally included human pose (h_x, h_y, h_θ) in the state, and the edges first simulated feasible robot trajectories, and then transformed them into human trajectories using the coupling model. In both planners, edge cost was determined with the same formula: a weighted linear combination of the distance traveled, the distance between the positions projected forward by 1 sec, orientation change, and closeness to obstacles along that edge. However, RobotOnlyPlanner calculated that cost using robot poses, while CoupledPlanner calculated that cost using human poses. Finally, all state space elements were discretized with a finer granularity closer to the goal, to allow more nuanced motions as the robot approached the target. Both planners ran until timeout, at $20,000$ cycles, or until they reached a node within a threshold distance and orientation from the goal. Upon timeout, the planner returned the best path found so far based on cost and heuristic, or failed if even the best path was too far from the goal. Both global planners outputted a robot path, which was passed on to the local planner. These planners were inspired by planners for self-driving cars and tractor-trailer vehicles [18], [26].

Our local planner was built on top of ROS navigation stack's base_local_planner. It used the trajectory rollout algorithm [27] to determine feasible trajectories, and evaluated them using a weighted linear combination of: (1) distance to the target robot path; (2) distance to the local goal along the target path; and (3) distance to obstacles. The robot trajectories from both RobotOnlyPlanner and CoupledPlanner were passed to the same local planner, which planned solely in robot space.

2) Environment and Procedure: We simulated the Willow Garage office using an open source map on the Gazebo platform. We chose human start and goal poses based on interesting features and likely places humans would go within the office area. Fig. 5 shows the 29 selected start and goal positions. To ensure repeatability, we localized the robot once per start position using AMCL, and then reset the robot to that pre-localized state per planner. We then sent a set of human goal poses to the planner, which RobotOnlyPlanner used to reverse-engineer corresponding robot goal poses (by calculating where the robot would be if the human were in

Fig. 5: The 29 trajectories we evaluated. The green arrows next to goal poses show desired ending orientations. Green trajectories were completed by both planners, red trajectories were only completed by `RobotOnlyPlanner`, blue trajectories were only completed by `CoupledPlanner`, and grey trajectories were completed by neither.

the middle of the region of occupancy). After each planner generated a robot path, the local planner followed that path. Due to sensor and robot motion stochasticity, each trajectory was repeated 5 times per planner.

As the local planner was executing, we ran the coupling model in-real-time (making calls every 0.2 seconds) to simulate where the human would be. We kept track of the average, max, and min distance between the human and the closest obstacle per timestep. After the local planner finished execution, we calculated (x, y, θ) differences between the final human pose and the closest of the human goal poses.

B. Results

As mentioned, we evaluated 29 paths on both planners. Not all the paths were completed by either planner. Both planners successfully executed 22 of those paths, where success was determined by at least one attempt succeeding. `RobotOnlyPlanner` successfully executed 3 paths that `CoupledPlanner` did not, and `CoupledPlanner` successfully executed 2 that `RobotOnlyPlanner` did not. 2 paths were completed by neither planner. For all the paths that did not complete, the global planners still found valid trajectories but the local planner failed to evaluate them due to: the robot's handle moving too close to walls, the local planner reaching local optima, or the local planner being unable to maneuver narrow hallways. Of the 22 paths that both planner completed, `CoupledPlanner` timed out 11 times and `RobotOnlyPlanner` timed out once.

Table VII shows the results of the simulation. The differences for all measures except Max Obs Dist. between planner were significant[2], $p < 0.001$. The results show that while both planners were similarly effective at obstacle avoidance, `CoupledPlanner` dropped the user off much closer to their target position. The large standard deviation

[2]Calculated through a multi-level linear model using REML with trajectory as a random factor.

TABLE VII: Robot Only Planner versus Coupled Planner*

	`RobotOnlyPlanner`	`CoupledPlanner`
Avg. Obs. Dist. (m)*	1.16 (1.22)	1.11 (1.16)
Max Obs. Dist. (m)	1.57 (1.48)	1.61 (1.42)
Min Obs. Dist. (m)*	0.60 (0.82)	0.51 (0.80)
Goal Dist. (m)*	0.31 (0.05)	0.17 (0.09)
θ Goal Dist. (rad)*	0.1 (0.2)	1.0 (1.1)

*Values are: Avg (StdDev) across 22 paths (only successful attempts) completed by both planners. * represents a significant difference.

in obstacle distance is likely due to the varied map areas we ran the simulation in. Finally, although `CoupledPlanner` performs much worse than `RobotOnlyPlanner` on θ goal distance, most of this error comes from cases where the local planner terminated while trying to exit a local minima with low position distance but high orientation difference. Such difference could be minimized with more search cycles or a better tuned cost parameters.

Although `CoupledPlanner` took longer (7778 compare to 4440 average cycles) and timed out more often, we believe this difference is acceptable given that it delivers users closer to their target position. In a navigation system that interleaves planning and execution, `CoupledPlanner` only needs to be used near the end of the plan, thereby impacting only a short portion of overall navigation. Finally, these execution times could be improved with global planners that incorporate parallelization, optimization, and/or stochasticity.

IX. CONCLUSION

In this paper, we presented a predictive model of how a human follows a robot while holding on. We collected data on human following behavior and trained multiple baseline and geometric models. The best model generates a region within which the human can comfortably hold onto the robot, assumes the human moves the minimum amount to stay in that region, and assumes the human actively tries to move to a more comfortable position while being pulled by the robot. We also demonstrated the model's generalizeability across time intervals and trajectory lengths, with small acceptable errors. Finally, we incorporated a trained instance of this model into a global planner to compare a planner that only plans in the robot space to one that jointly plans in the human and robot spaces. We found that the planner that jointly plans in the human and robot spaces, although slower, delivers the human much closer to their target position.

As with most novel efforts, there are limitations. First, our sample of human following behavior was small ($n = 10$), which may have introduced unwanted bias. Second, our robot never reversed. Although we felt this was reasonable since backing up while a human is holding on is usually unsafe, there may be cases where the robot must reverse. Third, design factors such as robot height, mass, behavior, and volume play a role in human following behavior [28], so it is possible that following behavior would change with different robots. However, given that the principles that motivated the creation of our model – the lag between human and robot motion, humans actively repositioning themselves, and

humans turning towards the tangent direction of a turn – are due to the anatomy of the human arm, we believe that the core aspects of GeoC will generalize to different robot designs. Fourth, our model was developed from data of following behavior in an office building. Although how a human follows a guide robot may be different in open areas or areas with many obstacles, our setting allowed us to explore map features that are crucial to leading people in many buildings. Fifth, how humans follow robots may change as robot navigation becomes more commonplace. However, we did not observe novelty effects in our user study, possibly due to the initial familiarization phase. Sixth, our model was trained on real-world robot motion data, whereas the data sent to it through a global planner will be simulated robot motion data. However, since the goal of global planners is to generate paths that are feasible for robots to navigate, we believe the paths will be similar enough for the coupling model to generalize (a belief supported by the coupling model's applicability in the simulation experiment). Finally, some participants wanted to visually follow the robot instead of holding on, and it is unclear how our model may generalize to that scenario.

This work reveals exciting new avenues for future research. On the modeling front, a model that accounts for contextual factors such as walls or pedestrians could be more applicable. One idea is to develop a model that, rather than predicting one human pose at a time, predicts a probability distribution over human poses that can be transformed based on environmental factors. In addition, a model that accounts for coupling factors such as interaction force, torque, or guiding velocity would likely describe following behavior more accurately, although it may be harder to integrate into existing planners. Further, a model that accounts for individual preferences, such as hand-holding and handle height, may prove more effective at predicting following behavior. Our initial investigation into this revealed that hand-holding preference may influence a perpendicular offset in participants' position relative to the robot. On the planning front, future work includes developing a local planner that incorporates human pose and evaluating the end-to-end, human-aware navigational system in a real-world experiment. One option for this local planner is using our coupling model, whose Markovian nature lends itself well to incorporation in state-of-the-art local planners. An alternative is using sensors to measure human position as the robot is moving. Finally, on the design front, future work should explore how robot design – notably, the design of the coupling link between the human and the robot – influences human following behavior.

ACKNOWLEDGEMENTS

We would like to thank Cecilia Morales and our user-testers for their invaluable feedback.

REFERENCES

[1] S. Azenkot, C. Feng, and M. Cakmak, "Enabling building service robots to guide blind people a participatory design approach," in *Proc. HRI*, March 2016, pp. 3–10.

[2] V. A. Kulyukin and C. Gharpure, "User intent communication in robot-assisted shopping for the blind," in *Advances in Human-Robot Interaction*. InTech, 2009.

[3] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: a second-generation museum tour-guide robot," in *Proc. ICRA*, vol. 3, May 1999, pp. 1999–2005 vol.3.

[4] S. Wang and H. I. Christensen, "Tritonbot: First lessons learned from deployment of a long-term autonomy tour guide robot," in *Proc. RO-MAN*, Aug 2018, pp. 158–165.

[5] H. . Gross, H. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten, and A. Bley, "Toomas: Interactive shopping guide robots in everyday use - final implementation and experiences from long-term field trials," in *Proc. HRI*, Oct 2009, pp. 2005–2012.

[6] Y. Chen, F. Wu, W. Shuai, N. Wang, R. Chen, and X. Chen, "Kejia robot–an attractive shopping mall guider," in *International Conference on Social Robotics*. Springer, 2015, pp. 145–154.

[7] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[8] M. Shomin, "Navigation and physical interaction with balancing robots," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, October 2016.

[9] J. Reinhardt, J. Schmidtler, M. Körber, and K. Bengler, "Follow me! wie roboter menschen führen sollen," *Zeitschrift für Arbeitswissenschaft*, vol. 70, no. 4, pp. 203–210, 2016.

[10] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proc. IEEE CVPR*, 2016, pp. 961–971.

[11] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proc. ICRA)*, May 2018, pp. 1–7.

[12] A. Garrell and A. Sanfeliu, "Cooperative social robots to accompany groups of people," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1675–1701, 2012.

[13] R. Gockley, J. Forlizzi, and R. Simmons, "Natural person-following behavior for social robots," in *Proc. HRI*. ACM, 2007, pp. 17–24.

[14] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "How to approach humans?-strategies for social robots to initiate interaction," in *Proc. HRI*, March 2009, pp. 109–116.

[15] Y. Morales, T. Kanda, and N. Hagita, "Walking together: Side-by-side walking model for an interacting robot," *Journal of Human-Robot Interaction*, vol. 3, no. 2, pp. 50–73, 2014.

[16] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Field and service robotics*. Springer, 2016, pp. 607–622.

[17] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *Proc. ICRA*, May 2013, pp. 2153–2160.

[18] R. Stahn, T. Stark, and A. Stopp, "Laser scanner-based navigation and motion planning for truck-trailer combinations," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, Sept 2007, pp. 1–6.

[19] O. Ljungqvist, N. Evestedt, M. Cirillo, D. Axehill, and O. Holmer, "Lattice-based motion planning for a general 2-trailer system," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 819–824.

[20] A. Kulkarni, A. Wang, L. Urbina, A. Steinfeld, and B. Dias, "Robotic assistance in indoor navigation for people who are blind," in *Companion of HRI*, March 2016, pp. 461–462.

[21] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proc. ICRA*, May 2011, pp. 3400–3407.

[22] A. Nanavati, X. Z. Tan, and A. Steinfeld, "Coupled indoor navigation for people who are blind," in *Companion of HRI*, 2018, pp. 201–202.

[23] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, 2018.

[24] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[25] D. R. Jones, "Direct global optimization algorithm direct global optimization algorithm," in *Encyclopedia of optimization*. Springer, 2001, pp. 431–440.

[26] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments: Part i," in *Proc. IROS*, Sept 2008, pp. 1063–1069.

[27] B. P. Gerkey and K. Konolige, "Planning and control in unstructured terrain," in *ICRA Workshop on Path Planning on Costmaps*, 2008.

[28] I. Rae, L. Takayama, and B. Mutlu, "The influence of height in robot-mediated communication," in *Proc. HRI*. IEEE Press, 2013, pp. 1–8.