SMT-Based Control and Feedback for Social Navigation

Thais Campos, Adam Pacheck, Guy Hoffman, and Hadas Kress-Gazit*

***Abstract*—This paper combines techniques from Formal Methods and Human-Robot Interaction (HRI) to address the challenge of a robot walking with a human while maintaining a socially acceptable distance and avoiding collisions. We formulate a set of constraints on the robot motion using Satisfiability Modulo Theories (SMT) formulas, and synthesize robot control that is guaranteed to be safe and correct. Due to its use of high-level formal specifications, the controller is able to provide feedback to the user in situations where human behavior causes the robot to fail. This feedback allows the human to adjust their behavior and recover joint navigation. We demonstrate the behavior of the robot in a variety of simulated scenarios and compare it to utility-based side-by-side navigation control.**

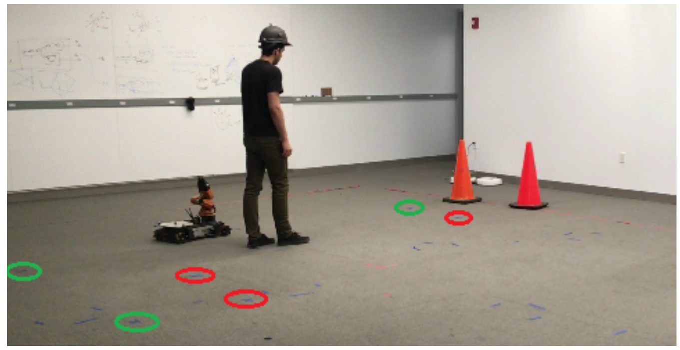

Fig. 1. A robot walking alongside a human following subgoals.

I. Introduction

Computational methods in Human Robot Interaction (HRI) have made significant advances in designing controllers for robots operating with and around humans (for a recent review, see [1]). The lack of formal representations and methods, however, has made it difficult for these systems to guarantee safety or correctness of operation. In this work, we combine techniques from Formal Methods and HRI to develop a human-appropriate robot controller with guaranteed behavior, and the ability to provide detailed feedback when safe or correct behavior cannot be guaranteed. We do so in the HRI domain of Social Navigation, specifically addressing the problem of navigating alongside a human.

Navigating in a safe, socially acceptable, communicative, and predictable way is a key skill for any robot moving in spaces shared with humans, especially when these humans are not trained experts. Example scenarios include delivery robots in offices, robots carrying tools and stock on factory floors, nursing support robots in healthcare applications, and robotic sales support in retail environments. These robots need to navigate around humans, approach them for collaboration, lead them from point to point, or walk with them while carrying a load related to the human's activity.

One aspect of Social Navigation is joint navigation between a robot and human, such as a robot walking side-by-side with a human [2], [3]. There are many cases where a robot should maintain a certain position and orientation with respect to a person. Consider a hospital patient walking down a hallway with an oxygen tank or an intravenous medicine bag, a robot supporting an older adult by carrying their groceries during shopping, or a robot carrying tools for a human laborer. In all cases, the robot would need to maintain a certain distance from the person without falling too much behind. To do so, the robot needs to infer something about the human's future movement, including where the person is heading and when the person intends to turn a corner. Equally importantly, the person would want assurances that the robot behavior is safe.

While there have been advances in developing robots that navigate side-by-side with people [2]–[5] , in these works there are no guarantees on the safety of the person or the success of the robot in maintaining a desired distance.

At the same time, the Formal Methods community has been developing techniques for synthesizing robot controls that provide guarantees of safety and task completion [6]. These formal approaches have also been used to provide feedback to the user regarding possible failures the robot may encounter, if no control is found [7]–[9]. While successfully used in various areas of autonomous robotics, these methods have rarely been applied to HRI scenarios, such as Social Navigation.

In this work, we combine insights and techniques from HRI and Formal Methods to synthesize control inputs for a robot traveling with a person while maintaining safety guarantees and automatically generating feedback.

A. Related Work

1) Social Navigation: A large body of work in HRI has been dedicated to the field of Social Navigation under the premise that humans are a "special kind of obstacle" necessitating a re-framing of the classic robot navigation problem [1], [10]–[12]. Within this area, several central models have emerged, including the Social Force Model (SFM) [13]–[15], and the Human Aware Motion Planner (HAMP) framework [16]. Some researchers have explored supervised learning to predict where a human will be to respect their personal space [17], [18].

*The authors are with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca NY 14853, USA {tcd58,akp84,hoffman,hadaskg}@cornell.edu

Most of these examples deal with approaching or avoiding humans. In addition, a separate body of work tries to build controllers for navigating side-by-side with a human. Some researchers have investigated how to design a wheelchair that would travel next to a caregiver or partner instead of in front of them [19], [20]. Others use a provably-safe motion planner to navigate a robot around a human [21]. They incorporate the uncertainty in human motion through predictive models and plan the robot's motion in real time. However, in [21], the robot does not provide feedback to the human and it is completely up to the robot to maintain a safe distance.

Another example of side-by-side navigation controllers uses utility-based algorithms to decide on the robot's next step, given a set of possible goals, with or without knowing which goal the human is currently traveling to [2], [3], [5]. In order to do this, the preferences of people for side-by-side travel, such as distance from each other and speed, were modeled. This was found to be more natural than simply using velocity to predict where the person will be next and moving to the point that will be next to the person.

2) Synthesis through Satisfiability Modulo Theory (SMT): In this paper we utilize an SMT solver to synthesize control actions for the robot. Given a set of Boolean and continuous variables, and constraints over them, an SMT solver will return a satisfying assignment to the variables or will state that the constraints are unsatisfiable. In the context of robot control, SMT has been used for composing motion primitives, for example, [22] use a SMT solver to synthesize motion plans for a multi-robot system. They impose constraints on the formation of quadrotors in a static environment and compute the composition offline. Similarly, [23] uses SMT solvers to compose safe motion primitives in the context of task and motion planning problems.

In [24], the authors generate motion plans for a robot performing a mobile manipulation task and use an SMT solver to find a plan that satisfies a set of logical constraints. If the motion plan does not comply with all of the constraints, a new motion plan is generated and verified. In [24], the SMT solver is used as the verification step in an inductive synthesis process rather than the primary synthesis tool, as in our work.

3) Automated feedback for infeasible tasks: When a specification cannot be synthesized, researchers have leveraged SAT solvers and counter-strategies to either pinpoint which parts of the specifications are inconsistent [8], [9], [25], [26], or what an adversarial environment can do to make the robot fail [7]–[9], [25]–[28]. This feedback is typically provided offline, during synthesis time, using natural or structured language [9], [26], [28]. In this paper we provide online feedback regarding constraint violation.

II. Approach

In this section we describe the process of encoding both the social and safety requirements of the joint navigation problem as an SMT formula, and the use of an SMT solver to synthesize guaranteed controls for the robot. If no controls exist that satisfy the requirements, the structure of the SMT formulation allows the robot to provide feedback regarding infeasible requirements in human-understandable terms. This enables the human to take corrective actions and the robot to resume correct navigation.

A. Satisfiability Modulo Theory (SMT) Solvers

Satisfiability Modulo Theory (SMT) solvers are automated reasoning engines capable of solving problems involving Boolean combinations of predicates (typically inequalities) from different *theories*. These theories include Linear Integer Arithmetic (with constraints such as $x_0 \cdot C_0 + \ldots + x_i \cdot C_i \geq C_{i+1}$) as well as non-linear Integer and Real Arithmetic (with constraints such as $\sin(x) + \cos(y) > 0.5$). Roughly speaking, SMT solvers alternate solving a Boolean satisfiability problem where truth values are assigned to the predicates, and theory solving, which attempts to find solutions to the mathematical inequalities that correspond to those truth values. Today, there are several popular SMT solvers that share a common standardized interface [29]. The two most popular for formulas involving large numbers of different theories are CVC4 [30] and Z3 [31], but there are also several solvers specialized in particular classes of formulas, for example, dReal [32] can solve problems involving nonlinear real functions.

B. Control Synthesis

To generate the control for the robot, we encode an SMT formula over the following continuous variables and constants:

- The current and next pose of the robot and the human in a 2D workspace $x_\alpha, y_\alpha, \theta_\alpha$, where $\alpha = \{RC, RN, HC, HN\}$, RC = Robot current, RN = Robot next, HC = Human current, HN = Human predicted next
- The next robot control v_R, ω_R corresponding to forward and angular velocities, respectively
- The current human speed v_H
- The time step dt
- Constants $d_{\theta RH}$ and d_{vRH} to encode the maximum difference in robot and human velocities and orientation, respectively
- Constants d_{min}, d_{max} and $d_{minWall}$ to define the minimum and maximum distance between human and robot, and the minimum distance between robot and walls to avoid collisions, respectively
- Constants v_{Rmax} and w_{Rmax} to limit the robot's forward and angular velocity, respectively
- The map (coordinates of the walls)

Robot dynamics: We model the dynamics of the robot as a differential drive robot and encode it as:

$$\varphi_{dymanics} = \left(\begin{bmatrix} x_{RN} \\ y_{RN} \\ \theta_{RN} \end{bmatrix} = \begin{bmatrix} x_{RC} + v_R dt \cos\theta_{RC} \\ y_{RC} + v_R dt \sin\theta_{RC} \\ \theta_{RC} + \omega_R dt \end{bmatrix} \right) \quad (1)$$

Human motion prediction: The generated control for the robot depends on the predicted pose of the human, which is encoded as part of the SMT formula:

$$\varphi_{human} = \left(\begin{bmatrix} x_{HN} \\ y_{HN} \\ \theta_{HN} \end{bmatrix} = \begin{bmatrix} x_{HC} + v_H dt \cos\theta_{HC} \\ y_{HC} + v_H dt \sin\theta_{HC} \\ \theta_{HC} \end{bmatrix} \right) \tag{2}$$

Desired behavior: We encode the desired robot behavior in additional predicates which include integer variables $b_i \in \{0, 1\}$, emulating Boolean variables indicating whether a constraint is satisfied or not. If the value of b_i is 1, the constraint is satisfied, while if it is 0, the constraint may or may not be satisfied for a valid solution of the SMT formula. If the SMT solver finds a solution where all b_i are 1, then all the constraints are satisfied. If not, the b_i are used to generate feedback, as described in Section II-C.

It is important to note that the solution provided by the SMT solver satisfies all the constraints, but it is not optimized w.r.t an objective function. There may exist multiple satisfying assignments to the variables, but only one solution is returned.

1) Correct Distance to Human: The robot should maintain a minimum distance from the human to avoid collision (d_{min}), but also not move too far away (d_{max}) to maintain socially acceptable behavior. The distance between the human and the robot (d_{RH}) is calculated according to Equation 3.

$$d_{RH} = \sqrt{(x_{RN} - x_{HN})^2 + (y_{RN} - y_{HN})^2} \tag{3}$$

We encode the distance constraints in Equation 4 with the integer variables b_{min} and b_{max}.

$$(b_{min}(d_{RH} - d_{min}) \geqslant 0) \wedge (b_{max}(d_{max} - d_{RH}) \geqslant 0) \tag{4}$$

2) Obstacle avoidance: The walls in the map are represented as vertical and horizontal line segments. In order to guarantee that the robot will not collide with any wall, it should keep a safe distance ($d_{minWall}$) from them. Equation 5 encodes this constraint, where b_{map} is an integer variable and d_{RWall} is the distance, calculated as the infinity-norm, between the robot's next pose and the nearest wall.

$$b_{map}(d_{RWall} - d_{minWall}) \geqslant 0 \tag{5}$$

3) Maximum forward and angular velocity: The robot should never travel or turn faster than a set limit (v_{Rmax} and ω_{Rmax}, respectively). These limits can either be due to physical limitations of the robot or user preference. Equation 6 shows this requirement, where b_V is an integer variable.

$$b_V(v_{Rmax} - v_R) \geqslant 0 \wedge b_V(\omega_{Rmax} - w_R) \geqslant 0 \tag{6}$$

4) Similar orientation and relative speed: In order to achieve natural side-by-side motion, the robot's orientation and speed should be similar to the human's.

$$\begin{aligned} &(b_{similar}(|\theta_{HN} - \theta_{RN}| - d_{\theta RH}) \leqslant 0) \\ \wedge\ &(b_{similar}(|v_{HN} - v_{RN}| - d_{vRH}) \leqslant 0) \end{aligned} \tag{7}$$

5) Satisfiability of maximum number of constraints: In order to make sure all constraints are satisfied, we add a predicate that requires all the integer variables to sum to 5, forcing each to be 1 as shown in Equation 8.

$$b_{min} + b_{max} + b_{map} + b_V + b_{similar} = 5 \tag{8}$$

The full SMT formula is the conjunction of Equations 3-8. In this work we use the SMT solver dReal [32] that allows for predicates that are nonlinear inequalities. The computation time on a 64 bit desktop machine running Ubuntu 14.04 with 8 GB RAM and a 3.6 GHz processor is approximately 0.02 seconds, which enables real-time control synthesis.

C. Feedback through SMT

If the SMT solver returns UNSAT (unsatisfied) then not all constraints can be satisfied; therefore, we decrease the sum in Equation 8 to 4, allowing for exactly one of the constraints to be violated. If the solver then returns SAT (satisfied), based on the values of b_i we know which constraint is infeasible. This enables us to create specific feedback that we are able to give to the human, similar to feedback one human might give to another through verbal or nonverbal communication.

In addition, we calculate new controls for the robot, such that safety is guaranteed:

- Violation of collision and obstacle avoidance ($b_{min} = 0$ or $b_{map} = 0$): the robot stops immediately ($v = 0, \omega = 0$)
- Violation of maximum distance to human ($b_{max} = 0$): the robot follows the synthesized control but issues a warning to the human
- Violation of maximum forward and angular velocities or similarity constraints ($b_V = 0$ or $b_{similar} = 0$): the robot maintains its previous motion as long as it does not collide with obstacles but issues a warning

If the solver instead still returns UNSAT, this means that more then one constraint is violated. We could continue to decrease the sum of integers, but due to the fact that we want the robot to calculate the controls in realtime, we instead choose to stop the robot and wait until controls can be generated with no more than one constraint violated. This will happen when the human takes corrective action that causes the constraints to become feasible again.

III. Evaluation

We illustrate the advantages of SMT-based control and the use of constraint-specific feedback by simulating joint navigation scenarios. We use a simulated human with an identical navigation policy in all the evaluations. We compare this approach to the state-of-the-art utility-based predictive control presented by [2], [3], [5]. In these simulations, the robot has access to a map of the environment and knows both its pose and the pose of the human.

A. Simulated Human Motion

In order to simulate the human behavior, we used a variant of the *Social Force Model* (SFM) described in [13], prevalent

TABLE I
COMPARISON BETWEEN UTILITY-BASED SIDE-BY-SIDE NAVIGATION AND SMT-BASED NAVIGATION WITH AND WITHOUT FEEDBACK TO THE HUMAN.

Scenario	Social Navigation Method: Utility-based	SMT without Feedback	SMT with Feedback
Attentive human, known subgoals, fixed environment	Robot maintains position and orientation with respect to human. With high cost, collision is unlikely, but not guaranteed to be avoided.	Robot maintains correct position and orientation with respect to human and is guaranteed to not collide.	Robot maintains correct position and orientation with respect to human and is guaranteed to not collide.
Unknown subgoals	Relies on a known list of subgoals.	Does not rely on knowing subgoals.	Does not rely on knowing subgoals.
New environments	Utility function is tuned to one environment and may not generalize to new environments easily. In practice, can reach positions where there is no possible next move, and robot deadlocks.	Less sensitive to specific environmental parameters. Does not display deadlock.	Less sensitive to specific environmental parameters. Does not display deadlock.
Inattentive human	The utility function cannot guarantee safety and can go into no-next-move unrecoverable deadlock.	Inattentive human can violate specification constraints and lead to unrecoverable deadlock.	Can avoid deadlock, assuming the human reacts according to the feedback given by the robot.

in Human Robot Interaction (HRI) research. In SFM, movement is modeled using attractive and repulsive forces to goals, other agents, and obstacles. Here, the human is attracted to subgoals and repelled from the walls and the robot.

In [2], the authors empirically found that during side-by-side walking, the average human speed is 1.09m/s with 0.11 standard deviation. In our evaluations, we used the same values for human speed, in order to compare these methods.

a) Attentiveness: We demonstrate our systems with two human behaviors: In one, they are *attentive* to the joint navigation situation and choose paths that more easily make way for the robot to move. In the second the human is *inattentive* and chooses paths that may limit the robot's movement options. We achieve these behaviors by placing subgoals in the environment in either useful or non-ideal ways.

B. Feedback

We compare two human-robot systems, one in which the human does not get (or ignores) feedback from the robot, and one where the human adjusts their motion based on the feedback. For the latter, for each violated constraint, we adjust the human's motion as follows:

1) Violation of minimum distance: The human increases their orientation angle with respect to the robot. However, since the robot has stopped, the human also needs to decrease velocity, so the robot can subsequently catch up. Velocity and orientation are picked at random from a normal distribution ($v_H \sim N(0.5,0.1)$ and $\theta_{\mathrm{HN}} \sim N$ ($\theta_{\mathrm{HC}} \pm 0.1$, 0.05)). The sign in the orientation normal distribution depends on the relative position of the robot compared to the human.

2) Violation of maximum distance: The human decreases its speed and angle with respect to the robot. The same values are chosen for the normal distributions ($v_H \sim$ N(0.5,0.1) and $\theta_{\mathrm{HN}} \sim$ N($\theta_{\mathrm{HC}} \pm$ 0.1, 0.05)).

3) Violation of maximum speed: The human only decreases its speed. We pick $v_H \sim$ N(0.5,0.1) and $\theta_{\mathrm{HN}} \sim$ N(θ_{HC}, 0.05).

4) Violation of collision avoidance: The collision avoidance is violated when the human, for example, backs the robot into a corner, still respecting its minimum distance. In this case, we need the human to deviate from the robot, decreasing its velocity more significantly and increasing its angle ($v_H \sim$ N(0.2,0.1) and $\theta_{\mathrm{HN}} \sim$ N($\theta_{\mathrm{HC}} \pm$ 0.1, 0.05)).

5) Violation of similar orientation and velocity: The human's orientation in the next time step should be similar to the robot's previous one, thus $\theta_{\mathrm{HN}} \sim$ N(θ_{RC}, 0.05), as well as

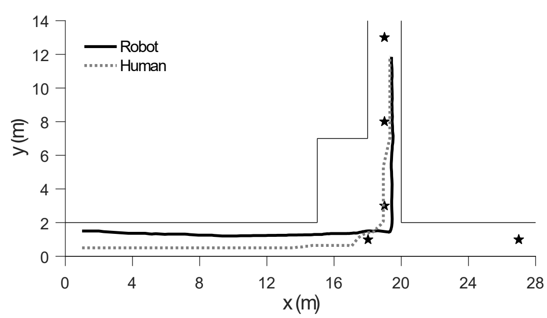

Fig. 2. Human and robot trajectories using utility-based navigation on a simple map with one intersection. Possible subgoals are depicted with stars.

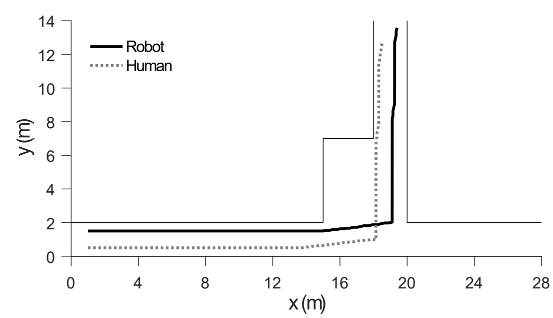

Fig. 3. Human and robot trajectories using SMT-based navigation on a simple map with one intersection.

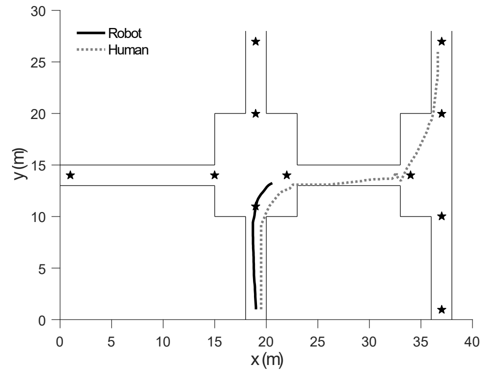

Fig. 4. Human and robot trajectories using utility-based navigation on a complex map with several possible subgoals (depicted as stars). As the space parameters changes, the robot may reach a stopping point as all possible considered points approach $-\infty$ utility.

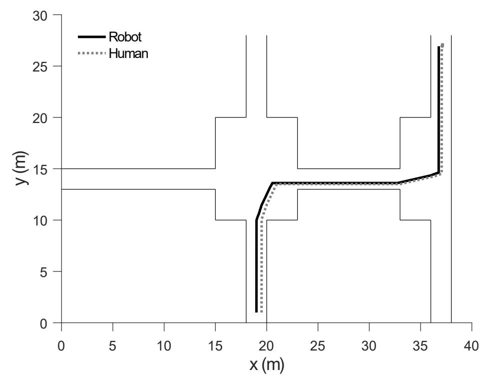

Fig. 5. Human and robot trajectories using SMT-based navigation on a complex map with several possible subgoals. The robot maintains a safe distance from the human and enables the pair to traverse the more complex environment, without need to change the parameters of the SMT formula.

its velocity $v_H \sim$ N(v_R + 0.1, 0.05).

6) Violation of more than one constraint: Since we do not know which constraints were violated, the human's velocity should decrease significantly so the robot can recover to a safe position ($v_H \sim$ N(0.2,0.1) and $\theta_{\mathrm{HN}} \sim$ N($\theta_{\mathrm{HC}} \pm$ 0.1, 0.05)).

IV. Results

Table I shows an overview of the scenarios and attentiveness behaviors compared in this section, and the outcomes for the three navigation controllers used in this section.

A. Attentive human, known possible subgoals

Figure 2 shows the trajectory of the human and the robot with utility-based control. The human is attentive and chooses one of the available subgoals at the intersection. This is comparable to the scenario described in [2], [3]. The utility-based controller handles this situation well.

The SMT solver generates a similarly trajectory in this case (Figure 3). It does so, however, without knowing about the human subgoals, or even possible subgoals in the environment.

B. New or Complex Environments, Unknown Subgoals

The parameters of the utility functions used in previous work and our simulations are based on human data collected in a specific environment. One of the benefits of controllers synthesized from high-level specifications is that they are less sensitive to specific environments. We evaluate this capability by using complex environments, with more corridors, and more possibilities for human navigation decisions. In Figure 4 the strict utility-based navigation predicts the next human pose based on its current velocity, heading, and time step, to be outside the map, which results in the robot stopping its motion.

In contrast, SMT-based navigation is robust to new environments and unknown subgoals as can be seen in Figure 5. In Figure 5, there is a time lag between the trajectories: the human walks in front of the robot in the section of the path parallel to x-axis, thus they do not collide.

C. Inattentive human behavior

The above examples showed human behavior that takes paths leaving room for the robot to navigate. However, if we add inattentive behavior, such that the robot becomes cornered, the SMT solver—while still maintaining the safety guarantees—is likely to enter a deadlock due to the violation of the specified constraints. In the example in Figure 6 the human veers to the right, and in order to maintain similar speed and orientation with the human, the robot does the same. However, this behavior violates the collision avoidance constraint, making the robot stop. The human, unaware of the robot's reason for stopping, continues on their trajectory violating a second constraint, the maximum distance. In the same scenario, the utility-based controller (Figure 7) also fails to find a valid next pose.

One of the benefits of SMT-based control is that the robot can pin-point the reason why the constraints were not satisfied, and give feedback to the human. Figure 8, shows that if the human responds to the feedback ("you are too close"), the human-robot pair can continue to move. Specifically, the feedback provided by the robot adjusts the human velocity and orientation based on the strategies described in Section III-B, enabling the pair to recover and reach their destination.

Figure 9 shows the distance between the human and the robot using all three strategies with an inattentive human, as depicted in Figures 6, 7, and 8. The SMT-based controller without feedback loses track of the human when the constraints are violated. The utility-based controller also loses track of the human when all possible next poses have $-\infty$ utility, after first violating the safety distance. The SMT controller with feedback, in contrast, is able to recover from violations of the specifications, and return to a correct trajectory.

Finally, Figure 10 demonstrates the flexibility of the SMT-based controller with feedback and shows that it can adapt to a variety of human trajectory choices, through complex maps, and with an inattentive human strategy.

V. Discussion

Our simulation results indicate that controllers synthesized using formal representations of the robot's constraints are a promising approach to joint navigation. Using constraints inspired by existing HRI methods, we can keep the same

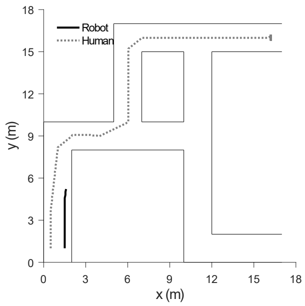

Fig. 6. SMT-based navigation on a more complex map with the human veering inattentively to the right. This action violates the maximum distance and collision avoidance constraints, thus the robot stops while the inattentive human continues walking.

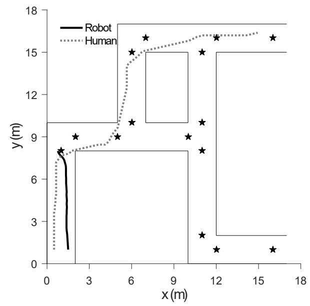

Fig. 7. Utility-based navigation on a more complex map with the human veering inattentively to the right. Near the first subgoal, the robot can not compute a valid next pose so it stops and the controller terminates.

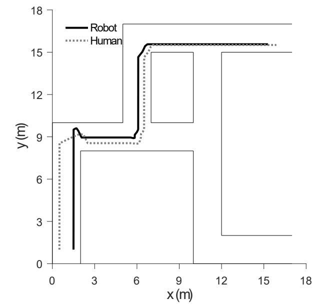

Fig. 8. SMT-based navigation with feedback on a more complex map with the human veering inattentively to the right. Here, the robot gives feedback to the human, who corrects their trajectory and the pair reaches the goal at top right.

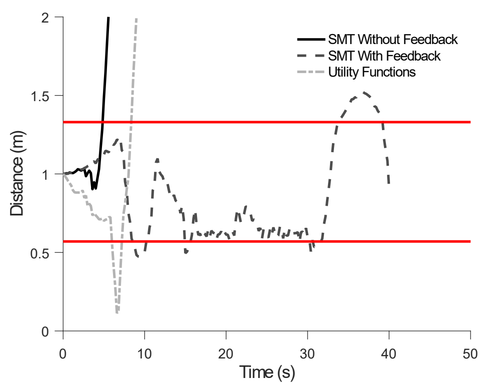

Fig. 9. Distance between the robot and human in the simulations shown in Figures 6, 7, and 8. Without feedback, the utility function and the SMT-based approaches result in the robot stopping, therefore the distance increases past the maximum allowed distance and continues to increase. When the robot provides feedback, transient violations are recoverable.

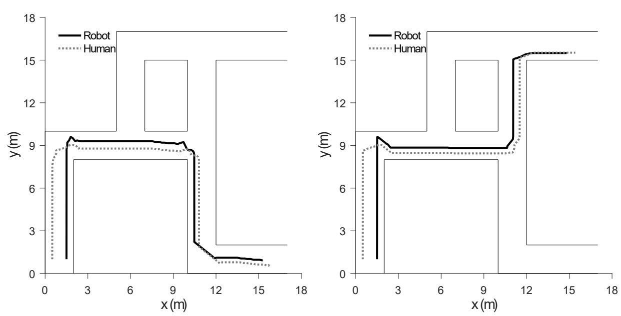

Fig. 10. Human and robot trajectories using SMT-based navigation with feedback on a more complex map can also handle a variety of human path choices with an inattentive human, without additional parameter tuning.

characteristics of natural collaborative motion with the added benefit of safety and feedback. Using a traditional state-of-the-art HRI controller there is a chance, however unlikely, that the optimization returns a motion plan that drives the robot towards an obstacle or even the human. This is prevented by the use of the continuous SMT constraints.

In addition, our method does not rely on knowing anything about the environment or subgoals, which allows us to generalize the robot's behavior more easily. We demonstrate this by using the same SMT formulation for a variety of environments and paths through these environments (see Figure 10).

In its effort to maintain safety, the SMT-based controller, however, stops when the safety constraints cannot be met. This can result in a deadlock, especially if the human is inattentive to the joint navigation and chooses a path that forces the robot into a position from which there are no safe or correct paths.

An additional benefit of controllers synthesized from formal specifications addresses this issue, by providing feedback to the human about what caused the robot to stop. We show that if the human takes this feedback into account and adjusts their behavior based on the specific problem the robot communicates to them, joint navigation can be resumed.

VI. Conclusion, Limitations, and Future Work

We describe a novel approach to robot social navigation that integrates insights and techniques from Formal Methods and HRI to achieve joint human-robot navigation. We demonstrate the feasibility of an SMT-based synthesized controller which can recover from deadlocks by giving the human feedback on failure to achieve the specified constraints.

One of the limitations of this work is that we have demonstrated and compared the controllers in a small set of environments. In the future we plan to extend our evaluation to real human trajectories, more complex models for both the human and the environment, and multiple agents.

As a next step, we plan on implementing these controllers on a physical robot, and evaluating it in human-subject experiments with respect to the task performance of the robot and people's perceived safety and comfort. We also plan to integrate optimization methods, in order to not only find one correct and safe path, but the best path (according to relevant metrics) for given formal specifications.

Still, the results from this work show promise that using Formal Methods and controllers automatically synthesized from a high-level task specification can provide safety, generalizability, and explainability beyond existing computational methods for HRI, and should be considered for other HRI domains.

REFERENCES

[1] A. Thomaz, G. Hoffman, and M. Cakmak, "Computational human-robot interaction," *Foundations and Trends® in Robotics*, vol. 4, no. 2-3, pp. 105–223, 2016.

[2] L. Y. Morales Saiki, S. Satake, R. Huq, D. Glas, T. Kanda, and N. Hagita, "How do people walk side-by-side?: using a computational model of human behavior for a social robot," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 301–308.

[3] R. Murakami, L. Y. Morales Saiki, S. Satake, T. Kanda, and H. Ishiguro, "Destination unknown: walking side-by-side without knowing the goal," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM, 2014, pp. 471–478.

[4] R. Gockley, J. Forlizzi, and R. Simmons, "Natural person-following behavior for social robots," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*. ACM, 2007, pp. 17–24.

[5] 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.

[6] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for robots: Guarantees and feedback for robot behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-104838

[7] R. Alur, S. Moarref, and U. Topcu, "Counter-strategy guided refinement of GR(1) temporal logic specifications," in *Formal Methods in Computer-Aided Design, FMCAD 2013, Portland, OR, USA, October 20-23, 2013*. IEEE, 2013, pp. 26–33. [Online]. Available: http://ieeexplore.ieee.org/document/6679387/

[8] V. Raman and H. Kress-Gazit, "Explaining impossible high-level robot behaviors," *IEEE Transactions on Robotics*, vol. 29, p. 94–104, 2013. [Online]. Available: http://dx.doi.org/10.1109/TRO.2012.2214558

[9] V. Raman, C. Lignos, C. Finucane, K. Lee, M. Marcus, and H. Kress-Gazit, "Sorry dave, i'm afraid i can't do that: Explaining unachievable robot tasks using natural language," in *Robotics: Science and Systems*, 2013.

[10] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[11] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, A. Sisbot, R. Alami, and T. Siméon, "How may i serve you?: A robot companion approaching a seated person in a helping context," in *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, ser. HRI '06. New York, NY, USA: ACM, 2006, pp. 172–179.

[12] L. Takayama and C. Pantofaru, "Influences on Proxemic Behaviors in Human-Robot Interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS'09*, no. 2009, 2009, pp. 5495–5502.

[13] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[14] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443–455, 2014.

[15] P. Ratsamee, Y. Mae, K. Ohara, M. Kojima, and T. Arai, "Social navigation model based on human intention analysis using face orientation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 1682–1687.

[16] E. Sisbot, L. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 874–883, Oct 2007.

[17] 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 *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. ACM, 2009, pp. 109–116.

[18] S. Satake, T. Kanda, D. Glas, M. Imai, H. Ishiguro, and N. Hagita, "A robot that approaches pedestrians," *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 508–524, April 2013.

[19] Y. Kobayashi, Y. Kinpara, E. Takano, Y. Kuno, K. Yamazaki, and A. Yamazaki, "Robotic wheelchair moving with caregiver collaboratively depending on circumstances," in *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2011, pp. 2239–2244.

[20] E. Prassler, D. Bank, B. Kluge, and M. Hagele, "Key technologies in robot assistants: Motion coordination between a human and a mobile robot," *Transactions on Control, Automation and Systems Engineering*, vol. 4, no. 1, pp. 56–61, 2002.

[21] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. J. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," *arXiv preprint arXiv:1806.00109*, 2018.

[22] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Automated composition of motion primitives for multi-robot systems from safe ltl specifications," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1525–1532.

[23] R. R. da Silva, B. Wu, and H. Lin, "Formal design of robot integrated task and motion planning," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 6589–6594.

[24] S. Nedunuri, S. Prabhu, M. Moll, S. Chaudhuri, and L. Kavraki, "Smt-based synthesis of integrated task and motion plans for mobile manipulation," in *Proc. of IEEE Intl Conf. on Robotics and Automation*, 2014.

[25] V. Raman and H. Kress-Gazit, "Analyzing unsynthesizable specifications for high-level robot behavior using ltlmop," in *23rd International Conference on Computer Aided Verification*, 2011, pp. 663–668.

[26] C. Lignos, V. Raman, C. Finucane, M. P. Marcus, and H. Kress-Gazit, "Provably correct reactive control from natural language," *Auton. Robots*, vol. 38, no. 1, pp. 89–105, 2015. [Online]. Available: https://doi.org/10.1007/s10514-014-9418-8

[27] W. Li, L. Dworkin, and S. A. Seshia, "Mining assumptions for synthesis," in *9th IEEE/ACM International Conference on Formal Methods and Models for Codesign, MEMOCODE 2011, Cambridge, UK, 11-13 July, 2011*, S. Singh, B. Jobstmann, M. Kishinevsky, and J. Brandt, Eds. IEEE, 2011, pp. 43–50. [Online]. Available: https://doi.org/10.1109/MEMCOD.2011.5970509

[28] L. Feng, M. Ghasemi, K. Chang, and U. Topcu, "Counterexamples for robotic planning explained in structured language," *CoRR*, vol. abs/1803.08966, 2018. [Online]. Available: http://arxiv.org/abs/1803.08966

[29] C. Barrett, P. Fontaine, and C. Tinelli, "The SMT-LIB Standard: Version 2.5," Department of Computer Science, The University of Iowa, Tech. Rep., 2015.

[30] C. Barrett, C. L. Conway, M. Deters, L. Hadarean, D. Jovanović, T. King, A. Reynolds, and C. Tinelli, "CVC4," in *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, 2011, pp. 171–177.

[31] L. De Moura and N. Bjørner, "Z3: an efficient smt solver," in *Proceedings of the Theory and practice of software, 14th international conference on Tools and algorithms for the construction and analysis of systems*, ser. TACAS'08/ETAPS'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 337–340.

[32] S. Gao, S. Kong, and E. M. Clarke, "dreal: An smt solver for nonlinear theories over the reals," in *International Conference on Automated Deduction*. Springer, 2013, pp. 208–214.