

Trust-based Multi-Robot Symbolic Motion Planning with a Human-in-the-Loop

YUE WANG, Clemson University

LAURA R. HUMPHREY, Air Force Research Laboratory

ZHANRUI LIAO, Clemson University

HUANFEI ZHENG, Clemson University

Symbolic motion planning for robots is the process of specifying and planning robot tasks in a discrete space, then carrying them out in a continuous space in a manner that preserves the discrete-level task specifications. Despite progress in symbolic motion planning, many challenges remain, including addressing scalability for multi-robot systems and improving solutions by incorporating human intelligence. In this paper, distributed symbolic motion planning for multi-robot systems is developed to address scalability. More specifically, compositional reasoning approaches are developed to decompose the global planning problem, and atomic propositions for observation, communication, and control are proposed to address inter-robot collision avoidance. To improve solution quality and adaptability, a dynamic, quantitative, and probabilistic human-to-robot trust model is developed to aid this decomposition. Furthermore, a trust-based real-time switching framework is proposed to switch between autonomous and manual motion planning for tradeoffs between task safety and efficiency. Deadlock- and livelock-free algorithms are designed to guarantee reachability of goals with a human-in-the-loop. A set of non-trivial multi-robot simulations with direct human input and trust evaluation are provided demonstrating the successful implementation of the trust-based multi-robot symbolic motion planning methods.

CCS Concepts: •**Computing methodologies** → **Robotic planning**; *Multi-agent planning*; •**Human-centered computing** → *User models*; •**Theory of computation** → *Verification by model checking*; •**Software and its engineering** → *Model checking*;

Additional Key Words and Phrases: Symbolic Motion Planning; Multi-Robot Systems; Trust; Human-in-the-Loop

ACM Reference format:

Yue Wang, Laura R. Humphrey, Zhanrui Liao, and Huanfei Zheng. 2017. Trust-based Multi-Robot Symbolic Motion Planning with a Human-in-the-Loop. *ACM Trans. Interact. Intell. Syst.* 9, 4, Article 39 (March 2017), 33 pages.

DOI: 0000001.0000001

1 INTRODUCTION

Despite advances in autonomy for robotic systems, human supervision/collaboration is often still necessary to ensure safe and efficient operations in uncertain, dynamic, or noisy environments where

This work is supported by the Air Force Research Laboratory's Summer Faculty Fellowship Program (SFFP) and the Air Force Office of Scientific Research Young Investigator Program under grant no. FA9550-17-1-0050.

Author's addresses: Y. Wang, Z. Liao, and H. Zheng, Department of Mechanical Engineering, Clemson University, Clemson SC 29634; L. R. Humphrey, Wright Patterson Air Force Research Laboratory, OH 45433.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 2160-6455/2017/3-ART39 \$15.00

DOI: 0000001.0000001

robot sensing and perception may not be fully reliable. While ideally autonomous robots are expected to be self-sufficient, there are practical tradeoffs between cost and performance. On one hand, humans excel at high-level decision-making in such environments and can help autonomous robots achieve better performance while keeping design costs low. On the other hand, human error is a main cause of machine malfunctions [1, 56], and human performance degrades when overloaded [13, 17]. When designing autonomous robotic systems, it is therefore important to consider factors related to human-robot interaction (HRI) [26]. However, although much research has been conducted on the development of effective approaches for HRI, extant solutions remain highly specialized and focused on human-machine interface (HMI) design [10]. The modeling, analysis, and implementation of effective HRI remains largely an open problem [30]. The current design process for robotic systems, especially in high-level decision-making and coordination, is still largely one of trial and error. In particular, the process often lacks quantitative models and real-time analytic approaches that could be used to provide safety and performance guarantees. The HRI problem in which a single human must interact with multiple autonomous robots is especially challenging due to the problem size, the need for robot coordination, the possibility of unintended emergent behaviors, etc.

Although much work has been undertaken to characterize physical HRI (pHRI) in terms of safety, performance, adaptability, etc. [27, 34], an important factor to consider with respect to social HRI (sHRI) and cognitive HRI (cHRI) is human trust in autonomous robots. Establishing trust in robots is the bottleneck in the development and integration of HRI systems. Trust can be defined as [45] (page 51)

“the attitude that an agent will help achieve an individual’s goals in a situation characterized by uncertainty and vulnerability.”

Trust is a dynamic feature of HRI [46] that heavily affects a human’s acceptance and hence use of a robot [29, 65]. Humans respond socially to robots, establishing a level of trust to manage workload not possible with mere human endeavor [23, 31]. Without trust, humans are likely to ignore or dismiss the assistance of robots and opt to complete a task on their own. Informed trust is an accurate assessment of when and how much autonomy should be employed, and when to intervene. Both economic and ergonomic benefits can be obtained by properly trusting the reliability of autonomous robots. Humans can either gain or lose trust in robots based on the progress of the task [14]. Moreover, consideration of trust is especially important for the supervisory control of multiple robots, since the tasks must be carefully allocated to ensure time-critical issues are addressed while human workload is kept within acceptable bounds [6, 57].

Some progress in addressing these deficiencies has been made through the application of formal methods – i.e., mathematically-based tools and techniques for system specification, design, and verification [5] – to problems involving HRI [10, 32]. The temporal logics commonly used in formal methods provide a high-level, human-like language for specifying desired properties or behaviors of a system, which can then be used to either verify or synthesize provably correct designs [5, 33]. Formal methods have been utilized in a wide range of applications including verification of HMI designs [10] and implementation of robot motion planning and control, leading to approaches such as symbolic motion planning for autonomous robots [8].

Symbolic motion planning extends traditional motion planning – generally concerned with fast computation of efficient, optimal, and dynamically feasible paths that allow mobile robots to reach a set of goal locations while avoiding obstacles [43] – by also incorporating higher-level goals and constraints expressed as temporal logic specifications, e.g., on the order in which goals should be reached and on regions that should or should not be visited. More specifically, in symbolic robot motion planning, the workspace is discretized, and different regions are given “symbolic” labels. A set of specifications for the robots can then be given in terms of these symbolic labels, e.g., “go

to locations A and B while avoiding obstacles.” Plans that meet the specifications are generated based on the discretized representation of the workspace. These high-level discrete plans are then converted into low-level reference trajectories and hybrid control laws based on robot dynamics in the continuous workspace such that the specifications met in the discrete representation are preserved [9]. This preservation of properties between the continuous and discrete workspace representations is important, since a major goal of symbolic motion planning is providing strong guarantees on the correctness of generated plans. In summary, symbolic motion planning approaches seek to guarantee properties at the discrete, symbolic level while also satisfying the types of continuous-level properties that would be addressed by traditional motion planning approaches.

Research in symbolic motion planning has focused on a number of areas, including efficient workspace discretization and plan computation [18], computation of control protocols for executing plans guaranteed to meet specifications in the presence of uncontrolled factors in the environment [67], sampling-based approaches [36] and approximate dynamic programming approaches [54] that do not require explicit discretization of the workspace. However, challenges remain in addressing the scalability of these approaches for multi-robot systems [41, 42] and in incorporating analysis of human behaviors to improve joint human-robot system performance when a human is able to interact with the system during plan execution [20, 24]. In particular, the specification language has been restricted to static and a priori known environments, and the extension to multi-robot systems is challenging due to the infamous “state-space explosion” problem since both the abstraction and the synthesis algorithms scale exponentially with the dimension of the configuration space [11, 60]. Since the explored system state needs to be stored in memory, this makes the problem intractable for systems with realistic size. Hence, there has been limited work on symbolic motion planning for fully autonomous multi-agent systems [2, 37, 60, 64] and very few recent results on HRI systems based on formal verification [20, 24, 47]. The papers [37, 64] investigate the task decomposition problem to address the scalability in multi-agent planning. The work [64] converts the linear temporal logic (LTL) specification to Büchi automaton and uses automaton-based model checking for multi-agent planning. To find the interdependency among agents, an event-based synchronization approach is proposed. The work [37] also uses automaton-based model checking for decentralized control of multi-agent systems. The decomposability and framework of parallel decomposition are discussed. The paper [60] develops an offline compositional multi-robot motion planning framework based on precomputed motion primitives and employs a satisfiability modulo theories (SMT) solver to synthesize trajectories for the individual robots. The work [2] considers a multi-agent persistent surveillance problem by minimizing the sum of time between two consecutive visits to regions of interest while satisfying their LTL specifications. A receding horizon controller with automata-theoretic approach is designed to plan each agent’s trajectory independently using only local information. The work [4] proposes a framework for controller synthesis based on compositional reactive synthesis for multi-agent systems. Different objectives such as collision avoidance, formation control and bounded reachability are considered. We note that our work differs from these previous works in several ways. With regard to work on multi-agent systems, incorporating uncertainty due to both the human and a priori unknown obstacles in the environment creates new challenges. For instance, [64] provides a distributed approach to multi-robot planning that is similar to ours, but it does not account for possible collisions between robots or uncertainty due to human interaction, which must be handled carefully in order to avoid problems such as deadlock and livelock. With regard to work on HRI systems, our approach does not rely on traditional automata-based or similar models of human behavior in order to make guarantees on system performance, allowing us to address more dynamic characteristics such as trust. We also note that there is a vast body of literature on robot motion planning, and the presented approach serves as a step towards high-level motion

planning for robotic systems with a human-in-the-loop at the discrete, symbolic level rather than an alternative to existing motion planning approaches at the continuous level.

In this paper, we investigate methods for improving the scalability, safety, performance, and adaptability of symbolic motion planning for multi-robot systems, taking into account the effects of human trust. Extending our preliminary works [48, 63], the main contributions of this paper are as follows.

- (1) We develop a dynamic, quantitative, and probabilistic human-to-robot trust model to compute the evolution of trust during real-time robotic operations. Our proposed trust model integrates the time-series trust model [58, 59, 66] and the Online Probabilistic Trust Inference Model (OPTIMo) [68] to compute trust quantitatively - a critical first step to enable trust-based multi-robot symbolic motion planning;
- (2) We propose trust-based specification decomposition to address the scalability issue in distributed multi-robot symbolic motion planning. Compositional reasoning approaches are used to decompose the global task specification into local specifications for each robot. Atomic propositions (AP) that relate to observation, communication, and control [21] are developed to resolve collisions between robots;
- (3) Since high-level human planning is beneficial in certain circumstances but difficult to verify, we design a trust-triggered real-time switching framework to automatically switch between manual and autonomous robot motion planning for tradeoffs between task safety and efficiency. Our main idea is to switch from autonomous to manual mode once sufficient trust has been developed and utilize human judgment to underpin riskier but more efficient solutions to path planning. This is the opposite to a number of other conceptions of HRI where the switch from autonomous to manual mode results from a loss of trust in the robot (e.g. the inverse trust metric in [22] that a robot uses to adapt behavior to increase operator's trust). Also note that here we focus on the high-level switching strategies; the low-level continuous robot motion execution under either manual or autonomous motion planning is still automatic. This differs from the literature on automaton-based switching/shared control synthesis for human-robot teams [24, 47];
- (4) Deadlock- and livelock-free algorithms are proposed to guarantee reachability of all the goals with a human-in-the-loop. We provide a formal proof for the correctness of the overall algorithm and motivate its need through a discussion of inter-robot collision scenarios that cannot be resolved by purely using the methods summarized in (2);
- (5) We perform non-trivial multi-robot simulations with direct human inputs (e.g., through gamepad and mouse with GUI designs) to demonstrate the real-time trust computation and the implementation of the proposed trust-based symbolic robot motion planning methods. We explore these methods in the context of an intelligence, surveillance, and reconnaissance (ISR) scenario. Other possible applications include search and rescue, DARPA Urban Challenge, warehouse management, intelligent transportation systems, etc.

The remainder of this paper is organized as follows. Section 2 introduces the problem setup and symbolic robot motion planning. Section 3 presents a dynamic, quantitative, and probabilistic model of human-to-robot trust. Section 4 outlines the method for trust-based specification decomposition, and Section 5 describes the method for switching between manual and autonomous motion planning. Section 6 proposes deadlock- and livelock-free algorithms to guarantee goal reachability with a human-in-the-loop. A set of simulations integrating Matlab and the NuSMV model checker [53] with direct human inputs is presented in Section 7. Concluding remarks are given in Section 8.

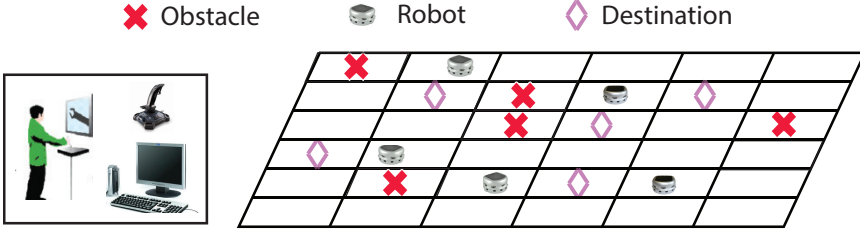


Fig. 1. Multiple robots must reach a set of destinations while avoiding obstacles and collisions with other robots, taking shorter but riskier paths between obstacles with human oversight when trusted to do so. That is, when the human operator trusts a robot, he/she plans for robot motion through visual feedback from a robot's onboard camera using HMI such as keyboard, mouse, or joystick.

2 SYMBOLIC ROBOT MOTION PLANNING CONSIDERING HRI

2.1 Problem Setup

We consider an ISR scenario in which a team of N robots represented by an index set $I_R = \{1, \dots, N\} \subset \mathbb{Z}$, supervised by a human operator, must reach a set of M goal destinations represented by the set $Goals = \{1, \dots, M\} \subset \mathbb{Z}$ while avoiding collisions with stationary obstacles and with each other (i.e. mobile obstacles), as shown in Fig. 1. As is standard in symbolic robot motion planning, the workspace is discretized into Q regions or states represented by the set $W = \{w_1, w_2, \dots, w_Q\}$, which are labeled with relevant properties (e.g., whether they contain an obstacle or goal). The regions can be in different shapes such as points, triangles, polytopes, or rectangles and the union of regions can be used to approximate the workspace of robots in an arbitrarily close manner. Depending on the robot dynamics and the corresponding low-level control, different cell decomposition strategies can be applied, as discussed later. Note this discretization can be performed to an arbitrary degree of accuracy; however, increasing the number of regions significantly increases the computational complexity of the planning problem. Most discretizations therefore significantly overapproximate certain features of the workspace, e.g., only a small portion of a region labeled as “obstacle” contains a real obstacle. The consequence is that planning through the workspace may be overly conservative, since paths that go through regions containing obstacles might be feasible in the continuous workspace. Though theoretically a finer discretization of the space would allow for computation of these paths, this may not be desirable in practice since it will lead to a very large state space which requires significant computational resources. In such a case, a combination of human judgment, assistance, and permission might be needed in order to make the attempt, motivating the need for a human operator.

We assume that the set of M goal destinations is known from the start, and each goal must be reached by at least one robot while collisions with obstacles and between robots are avoided. This set of requirements forms a specification for the scenario. Our proposed planning scheme is implemented in a distributed manner, making use of compositional reasoning approaches to decompose the global specification. More specifically, the goal portion of the specification is decomposed such that each robot is assigned a subset of the goal destinations and locally synthesizes a plan to reach them. For obstacle avoidance, we assume obstacle locations are not known a priori, and so when a robot discovers an obstacle, it re-synthesizes a plan to reach its remaining goals after updating its representation of the workspace. We show that our planning approach is guaranteed to meet these specifications under some mild assumptions. We then take this planning approach and incorporate a method for inter-robot collision avoidance in which the involved robots locally collaborate to

re-synthesize plans. We later modify the algorithm to guarantee the reachability of all M goal destinations by eliminating the possibility of deadlock and livelock.

With respect to human interaction, a quantitative, probabilistic, and dynamic trust model based on robot performance, human performance, joint human-robot fault, human intervention, and feedback evaluation is used to estimate human trust in each of the robots throughout the scenario. This estimate of trust affects the specification decomposition, with more trusted robots assigned more destinations. Trust is also used to determine when the robot should suggest navigating between obstacles, as this requires real-time switching between manual and autonomous motion planning. Human consent for this switching is assumed to depend on the change of trust as well as whether or not the human is currently occupied with other tasks. Each robot is assumed to follow a simple first-order kinematic equation of motion

$$\dot{x}_i = u_i, \quad i \in I_R = \{1, 2, \dots, N\}, \quad (1)$$

where x_i is the robot position and u_i is the motion control input. We assume homogeneous robots with the same dynamics. To guarantee that the high-level discrete motion plan can be realized by some low-level continuous motion control, we use a rectangular partition and a linear quadratic regulator (LQR) controller [40] for the robot kinematics (1) so that the robot always seeks to reach the centroid of the next cell in the discrete path, i.e.,

$$u_i t = -K_i x_i t - x_i^d, \quad K_i = R_i^{-1} P_i t, \quad (2)$$

where K_i is some control gain, x_i^d represents the centroid of a discrete cell, $P_i t$ is found by solving the Riccati equation $P_i t R_i^{-1} P_i t = Q_i$, and R_i , Q_i are the weights on state and control input with a quadratic cost, respectively. Therefore, under this setup, for the high-level discrete plan, there always exists a low-level continuous control under (1) given the rectangular cell partition. This guarantees the robot will never enter an unplanned region while moving between sequential regions in the planned path, allowing us to establish a bisimulation relation between the continuous state space used for control of the robot and the discretized state space used for planning [3]. Since we assume homogeneous multi-robot systems, we can use the same the partition of the workspace for each robot for autonomous motion planning. In more general cases, it has been proven that for any robot with affine dynamics (e.g. the kinematics (1) is a special case of affine dynamics), the low-level continuous control and high-level discrete plan are bisimilar under polygon triangulation [9]. Open source toolboxes such as TuLip (<http://tulip-control.sourceforge.net>) can be used to perform the partitioning/triangulization. Under manual motion planning, when the human trusts a robot, he/she will assign waypoints as high-level guidance. In that case, rather than moving between cell centroids, the robot will navigate autonomously between successive waypoints using the low-level controller (2) where x_i^d becomes the position of a waypoint, and no partition is needed. In our system, the human can see the grid map as displayed in Fig. 15 and Fig. 17, trust level (Fig. 16(a)), visual information, and support robot requests via the robot's onboard camera and HMIs (Fig. 14) and/or graphical user interface (GUI) (Fig. 16(b) and (c)) as depicted in Fig. 1. More details regarding the HMI and GUI designs will be provided in the simulation section 7.

Fig. 2 shows the schematic of the proposed trust-based human-robot collaboration system and details the specification decomposition process and switching framework for one robot. Because robots are good at local tasks given their limited computing, sensing, and communication capabilities, an in-situ autonomous motion planner is developed based on local sensing and communication information to guarantee safe completion of the task. However, since humans excel at high-level planning, we allow the human to intervene in the robot motion planning if required to increase task efficiency. This combination of autonomous and manual motion planning allows the joint system to achieve the global specification efficiently but without overloading the human operator. As

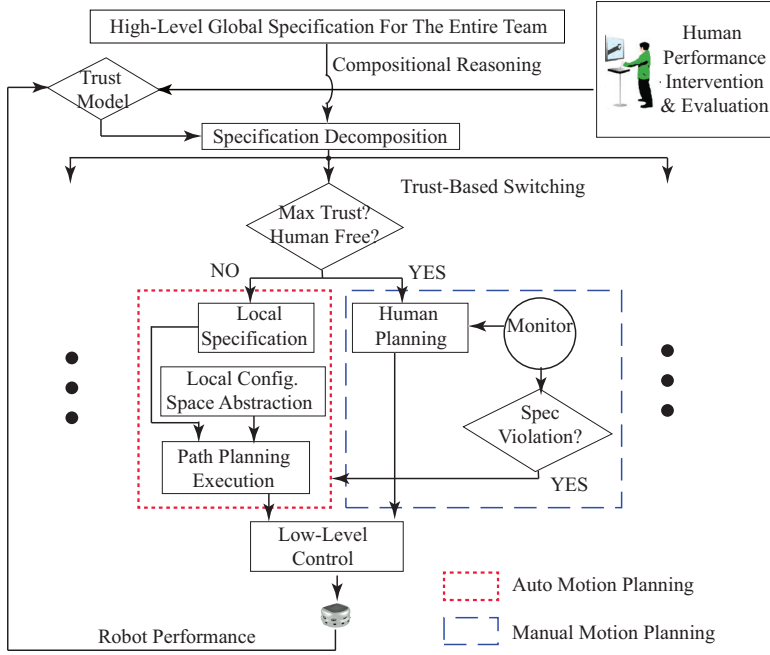


Fig. 2. Schematic diagram of trust-based symbolic motion planning for human and multi-robot collaboration systems.

planning and execution proceeds, human trust in each robot evolves dynamically, with the human choosing to collaborate more often with trusted robots. To reduce computational complexity, the global specification for the HRI team is decomposed using compositional reasoning methods, with more trusted robots assigned more tasks and vice versa. To further integrate human intelligence while guaranteeing task safety, a trust-based real-time switching framework is developed, and an autonomous decision-making aid is designed. By default, the autonomous and safe motion planning is implemented to guarantee task completion based on an over-approximation of the task domain, ensuring that obstacles are avoided. Each robot then carries out the control synthesis process (including local configuration space abstraction, local path planning, and execution) based on its decomposed local specification. However, if the human trusts a robot and is not overloaded, manual motion planning may be requested for more efficient but riskier solutions, e.g., moving between close obstacles. In such a case, the robot behavior under manual motion planning needs to be monitored so that if any event that will violate the local task specification is detected, the autonomous motion planning will be activated again. At this moment, the robot can either alert the operator to this change of operation mode, or leave it to the operator to see that an obstacle has been detected and notice that the robot has returned to its autonomous mode of operation¹. Robot and human performance measurements, joint human-robot fault measurements, direct human intervention, and trust evaluation are used as feedback to update the trust model and specification decomposition.

¹In our simulation Section 7, the robot does not alert the change of mode when the autonomous mode is reactivated.

2.2 Symbolic Robot Motion Planning

Symbolic robot motion planning is performed over a discrete abstraction of the robot's workspace, as previously described. The set of discrete cells, the properties each cell satisfies, and the movements allowed between cells in the workspace are often represented as a finite transition system, defined as follows.

Definition 2.1 (Finite Labeled Transition System). A finite labeled transition system is a tuple $TS = S, Act, \delta, I, \Pi, L$ consisting of

- (1) a finite set of states S ,
- (2) a finite set of actions Act ,
- (3) a transition relation $\delta \subseteq S \times Act \times S$,
- (4) a set of initial states $I \in S$,
- (5) a set of atomic propositions (AP) Π ,
- (6) and a labeling function $L : S \rightarrow 2^\Pi$.

We define a path of a transition system as a sequence of states $\sigma = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \dots$, where $s_k \in S$ is the state at step $k \geq 0$, $s_0 \in I$, and $s_k, \alpha_k, s_{k+1} \in \delta$ with $\alpha_k \in Act$. We define a trace as a sequence of labels that denote which APs are true in each state, i.e., $L\sigma = Ls_0Ls_1Ls_2 \dots$

Let us define a transition system TS_w to represent the discrete abstraction of the workspace. Then W is the set of cells in the workspace, Act_w is the set of actions that represent the robot moving between cells, δ_w encodes pairs of adjacent cells, I_w is the set of possible starting cells (in this case, all cells without obstacles), Π_w contains atomic propositions π_j^g for $j \in Goals$ and π_j^b for $j \in Obs$ representing the presence of obstacles or goals in workspace cell j , and L labels cells that contain obstacles or goals. At the start of the scenario, each robot $i \in I_R$ performs planning using a local copy TS_i of TS_w in which obstacle locations are not initially known, i.e., the labeling function L_i does not initially return π_j^b for any $s \in S_i$. As each robot i learns the location of obstacles, the labeling function of TS_i is updated.

Specifications for symbolic robot motion planning can be expressed as LTL formulae [5]. An LTL formula φ is formed from APs, propositional logic operators, and temporal operators according to the grammar

$$\varphi ::= \text{true} \mid \pi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 U \varphi_2, \quad (3)$$

where π is an AP, \wedge is the propositional logic operator “and,” \bigcirc is the temporal operator “next,” and U is the temporal operator “until.” From these, the standard propositional operators \vee “or,” \rightarrow “implies,” etc. can be derived in addition to the temporal operators \square “always” and \diamond “eventually.” The formula $\square\pi$ is true for a trace if propositional formula π is true in every state of the trace, $\diamond\pi$ is true if π is true in some state of the trace, and $\diamond\square\pi$ is true if π is true in some state of the trace and all states thereafter. LTL can be used to specify expressive motion tasks such as reachability, (“reach π eventually”, i.e., $\diamond\pi$), obstacle avoidance and safety (“always avoid π ”, i.e., $\square\neg\pi$), convergence and stability (“reach π eventually and stay there for all future times”, i.e., $\diamond\square\pi$), and sequencing and temporal ordering of different tasks that can otherwise be difficult to specify in traditional path planning.

Given an LTL plan specification φ and a finite labeled transition system TS , a high-level discrete plan that satisfies the specification can then be synthesized using a model checking approach. Traditionally, a model checker verifies whether a system TS satisfies a specification φ , written $TS \models \varphi$. If not, it returns a counterexample trace $\sigma \in \text{traces}TS$ that does not satisfy the specification, i.e., $L\sigma \not\models \varphi$. Model checking TS against the negation of the specifications returns a trace that satisfies the specifications, since $L\sigma \not\models \neg\varphi \rightarrow L\sigma \models \varphi$. For symbolic robot motion planning, this

approach generally produces short paths relatively quickly [32]. Here, we generate plans using this approach with the NuSMV model checker [53]. For a team of robots TS_i , $i = \{1, 2, \dots, N\}$ with each TS_i as the discretized abstraction of the robot motion within the workspace, to satisfy a feasible global specification φ , we can then use this approach with the model checker NuSMV to generate motion plans for the multi-robot team. That is, the motion plan for each robot will encode a single path $\sigma^i = s_0^i \xrightarrow{\alpha_0^i} s_1^i \xrightarrow{\alpha_1^i} s_2^i \dots$ obtained by model checking, where $s_k^i \in S_i$ is the robot i 's state at step $k = 0, 1, \dots$ and pairs of sequential states $s_k^i \xrightarrow{\alpha_k^i} s_{k+1}^i \in \delta_i$ with $\alpha_k^i \in Act_i$ represent feasible transitions between states, i.e., direct transitions between states that are achievable in the continuous workspace.

For the ISR scenario illustrated in Fig. 1, we are interested in specifications of the form

$$\varphi = \underbrace{j \in Goals}_{\text{Reachability}} \Diamond \pi_j^g \wedge \underbrace{\bigwedge_{i=1}^N \square \pi_{ij}^o \rightarrow \neg \bigcirc \pi_j^b}_{\text{Obstacle Avoidance}} \wedge \underbrace{\bigwedge_{i=1}^N \square \pi_i^c \rightarrow \neg \bigcirc \pi_i^u}_{\text{Robot Collision Avoidance}}. \quad (4)$$

The above LTL specification consists of a set Π of APs whose truth values are determined by the workspace that the human-robot team is operating in and observed by the robots' sensors. The value of each $\pi \in \Pi$ is a Boolean valuable. The propositions of the form π_j^g for $j \in Goals$ and π_j^b for $j \in Obs$ label regions containing *Goals* and *Obstacles*. The term $j \in Goals \Diamond \pi_j^g$ indicates the reachability specifications, i.e., all the goals will be eventually visited by a robot. Propositions of the form π_{ij}^o , π_i^c , and π_i^u represent the observation, communication, and control APs. The term $\bigwedge_{i=1}^N \square \pi_{ij}^o \rightarrow \neg \bigcirc \pi_j^b$ indicates the obstacle avoidance specification, i.e., it is always true that once robot i 's sensor observes an obstacle j , the robot will not move to cell w_j labeled with π_j^b at the next step. The term $\bigwedge_{i=1}^N \square \pi_i^c \rightarrow \neg \bigcirc \pi_i^u$ indicates the inter-robot collision avoidance specification, i.e., it is always true that once robot i finds another robot within its communication range, they will exchange goal, obstacle, and planned path information, and robot i will not implement the LQR controller (2) at the next step (but wait or replan). Note that the while the set *Goals* is known by each robot at the start of the scenario, the set *Obs* is not. Rather, each robot must update its transition system model over time as it discovers obstacles, replanning when necessary. Similarly, propositions of the form π_{ij}^o , π_i^c , and π_i^u can only be evaluated during execution and are therefore only used to trigger the replanning process. This allows us to focus on distributed multi-robot symbolic motion planning for scalability. More details regarding the specification decomposition will be further explained in Section 4.

3 COMPUTATIONAL TRUST MODEL

Existing trust models include argument-based probabilistic models [12, 51], qualitative models [35, 51], time-series models [25, 44, 45, 52], neural net models [19], regression models [15], and the most recent Bayesian dynamic network based model called OPTIMo [68]. In particular, the time-series model characterizes the dynamic relationship between human trust in robots and the independent variables. Unlike most other types of models, the time-series model proposed by Lee and Morray is based on prior trust, robot performance, and joint human-robot fault [44]. This model reflects the dynamic changes of trust and may be suitable for both real-time analysis and prediction of trust for control allocation. The OPTIMo trust model uses a dynamic Bayesian network (DBN) to infer beliefs over the human's latent trust states, or the degree of trust in near real-time, based on the history of observed interaction experiences. Although a performance-centric model, OPTIMo offers a probabilistic and dynamic framework to estimate and predict trust in complex tasks. There are also several computational trust models available [28, 49]. However, these models only characterize a robot's trustworthiness based on its task performance and do not consider the human influence. In

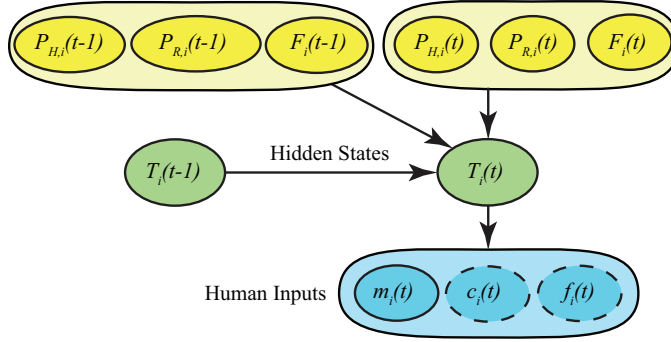


Fig. 3. A dynamic Bayesian network (DBN) based model for dynamic, quantitative, and probabilistic trust estimates.

this section, we integrate the time-series trust model and the OPTIMo trust model and develop a new mathematical model for estimating human trust in a robot, which is dynamic, quantitative, and probabilistic, and takes into account joint human-robot performance.

Since human behaviors are notoriously difficult to model, predict, and verify and the task environment is usually complex and uncertain, probabilistic analysis must be utilized to capture these uncertainties in trust estimates. Our proposed integrated trust model is shown in Fig. 3. First, let T_{ik} denote human trust in robot i , $i = 1, 2, \dots, N$, which is a hidden random variable taking values from 0 to 1. This assumption is made because trust is difficult to measure directly in real-time and is usually measured subjectively after each experiment session. We use solid green ellipses to represent T_i in the figure, indicating that this is a process evolving in real-time. The discrete trust state results in a standard DBN model. The actual realization and the sequence of the process, i.e., dynamic evolution of trust over time, is hidden. Based on our previous works involving creation of a time-series trust model [58, 59, 66], we identify three major factors impacting trust, i.e., robot performance $P_{R,i}$, human performance $P_{H,i}$, and joint human-robot system fault F_i . These factors are shown in solid yellow ellipses in the figure. Following the OPTIMo model, we also have the human inputs m_{ik}, c_{ik}, f_{ik} represented by solid and dashed blue ellipses in the figure, with dashed ellipses indicating intermittent observations. This is because it might not be practical to have human inputs all the time. The term $m_{ik} \in \{0, 1\}$ represents human intervention (i.e., switches between manual and autonomous modes) in motion planning, and its default value of zero indicates no intervention. Hence, m_{ik} can be measured and updated in real-time. The term $c_i \in \{-1, 0, +1\}$ represents change in trust as reported by the human, with -1 indicating a decrease in trust, 0 indicating no change, and +1 indicating an increase in trust. The term $f_i \in [0, 1]$ represents subjective trust feedback, which is a continuous value between 0 and 1. Both c_i and f_i only require occasional observations. That is, the participants will only be asked to provide trust change c_{ik} and trust feedback f_{ik} periodically. This ensures there is not much additional cognitive workload for the human operator during multi-robot cooperative tasks.

The conditional probability distribution (CPD) of human trust in robot i at time t based on the previous trust value $T_{it} - 1$ and the above causal factors can be expressed as a Gaussian distribution with mean value \bar{T}_{it} and covariance σ_{it} :

$$\begin{aligned} & \text{Prob} T_{it} | T_{it} - 1, P_{R,it}, P_{R,it} - 1, P_{H,it}, P_{H,it} - 1, F_{it}, F_{it} - 1 \\ &= \mathcal{N}(T_{it}; \bar{T}_{it}, \sigma_{it}), \end{aligned} \quad (5)$$

$$\text{where } \bar{T}_i t = A T_i t - 1 + B_1 P_{R,i} t - B_2 P_{R,i} t - 1 + C_1 P_{H,i} t - C_2 P_{H,i} t - 1 + D_1 F_i t - D_2 F_i t - 1. \quad (6)$$

Here, $\bar{T}_i t \in 0, 1$ represents the mean value of human trust in a robot i at time t , $P_{R,i} \in 0, 1$ represents performance of robot i , $P_{H,i} \in 0, 1$ represents human performance, $F_i \in 0, 1$ represents faults made in the joint human-robot system and σ_i reflects the variance in each individual's trust update. The coefficients $A, B_1, B_2, C_1, C_2, D_1, D_2$ can be determined by data collected from human subject tests (see the author's previous work [58] for more details). Here, these coefficients are further scaled such that \bar{T}_i is normalized. See Fig. 18 for an illustration of the dynamics of mean trust \bar{T}_i .

In this scenario, robot performance $P_{R,i}$ is modeled as a function of “rewards” the robot receives when it identifies an obstacle or reaches a goal destination:

$$P_{R,i} t = C_O \frac{N_{O_i} t}{\sum_{i=1}^N N_{O_i} t} + C_G \frac{N_{G_i} t}{\sum_{i=1}^N N_{G_i} t}$$

where N_{O_i} and N_{G_i} are the number of obstacles detected and goals reached by the robot i up to time t , and $C_O \in 0, 1$ and $C_G = 1 - C_O \in 0, 1$ are corresponding positive rewards chosen such that $P_{R,i} t$ is normalized. This allows the robot to earn trust as it learns details of the environment.

Human performance is calculated based on workload and the complexity of the environment surrounding the robot with which the human is currently collaborating. The concept of utilization ratio, γ , is used to measure workload [62]

$$\gamma = \gamma - 1 + \frac{\sum_{i=1}^N m_i t - \gamma t - 1}{\tau} \quad (7)$$

where $m_i t = 1$ if the human is collaborating with robot i and 0 otherwise, and τ can be thought of as the sensitivity of the operator. Assuming a human can only collaborate with one robot at a time, i.e., manually assigning paths through obstacles for the chosen robot, (7) allows workload to grow or decay between 0 and 1. Complexity of the environment is based on the number of obstacles that lie within sensing range r_i of collaborating robot i at time t . The human's superior capability in creating more detailed paths will be enhanced in more complex environments, leading to increased performance in the presence of more obstacles. On the other hand, human performance decreases with respect to workload. Therefore, $P_{H,i} t$ can be modeled as follows:

$$P_{H,i} t = \begin{cases} 1 - \gamma^{S_{O_i} t + 1} & \text{if } m_i t = 1 \\ 1 - \gamma & \text{if } m_i t = 0 \end{cases} \quad (8)$$

where S_{O_i} is the number of obstacles within sensing range of collaborating robot i , reflecting the environmental complexity. Fig. 4 shows the change of human performance with respect to workload γ and environmental complexity S_{O_i} .

Faults in the system are modeled as the “penalty” the robot receives when it enters an obstacle region or detects an obstacle on its planned path: $F_i t = -\frac{N_{H_i} t}{N_{O_i} t}$ where $N_{H_i} t$ is the total number of obstacle regions robot i has entered before sensing the corresponding obstacle up to time t . Note that faults can originate from both the robot and the human, i.e., human trust in a robot will decrease even if the robot enters an obstacle region under manual motion planning.

Based on the DBN, we can quickly establish the belief update of trust, i.e., $bel T_i t = \text{Prob} T_i t | P_{R,i} 1 : t, P_{H,i} 1 : t, F_i 1 : t, m_i 1 : t, c_i 1 : t, f_i 1 : t, T_i 0$, using the forward algorithm by applying the principle of dynamic programming to avoid incurring exponential computation time due to the increase of t . We

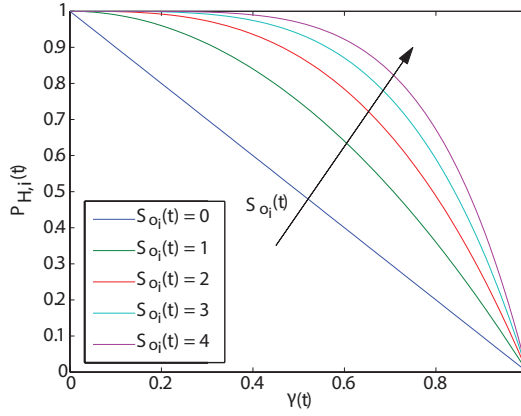


Fig. 4. Plot of human performance when collaborating with a robot i .

can first compute

$$\begin{aligned} & \overline{bel}T_{it}, T_{it} - 1 \\ = & \text{Prob}m_{it}|T_{it}, T_{it} - 1 \text{Prob}c_{it}|T_{it}, T_{it} - 1 \text{Prob}f_{it}|T_{it} \cdot \\ & \text{Prob}T_{it}|T_{it} - 1, P_{R,it}, P_{R,it} - 1, P_{H,it}, P_{H,it} - 1, F_{it}, F_{it} - 1, \text{bel}T_{it} - 1, \end{aligned}$$

where $\text{Prob}m_{it}|T_{it}, T_{it} - 1$ is the probability of human intervention, $\text{Prob}c_{it}|T_{it}, T_{it} - 1$ is the probability of a trust change given current and prior trust, and $\text{Prob}f_{it}|T_{it}$ is the probability of subjective trust evaluation, respectively, which can follow a similar sigmoid distribution as in [68]. For example, the CPD of human intervention based on trust can be modeled as follows

$$\text{Prob}m_{it} = 1|T_{it}, T_{it} - 1 = 1 + \exp(-\omega_1 T_{it} - \omega_2 T_{it} - 1)^{-1} \quad (9)$$

where ω_1 and ω_2 are positive weights and this CPD indicates higher willingness to collaborate with a robot (intervention in path planning) when the human trust is higher. It follows that

$$\text{bel}T_{it} = \frac{\overline{bel}T_{it}, T_{it} - 1 dT_{it} - 1}{\overline{bel}T_{it}, T_{it} - 1 dT_{it} - 1 dT_{it}}. \quad (10)$$

The network parameters for the DBN such as ω_1 and ω_2 in Equation (9) can be learned by the well-known expectation maximization (EM) algorithm [50]. For the implementation of the EM algorithm, we summarize the steps as follows:

- Expectation steps:
 - (1) Calculate the filtered trust belief $\text{bel}T_{it}$ at the end of the training simulation $t = t_f$ forward in time assuming a uniform initial trust belief;
 - (2) Calculate the smooth trust belief $\text{bel}_s T_{it} = \text{Prob}T_{it}|P_{R,i}1 : t_f, P_{H,i}1 : t_f, F_{i1} : t_f, m_{i1} : t_f, c_{i1} : t_f, f_{i1} : t_f, T_{i0}$ for all data backward in time given that $\text{bel}_s T_{it_f} = \text{bel}T_{it_f}$;
 - (3) Take the expectation for each smooth trust belief to get a single sequence of trust states.
- Maximization step: Use this calculated sequence of trust states at the Expectation steps, along with other performance $P_{R,i}, P_{H,i}, F_i$ and human inputs m_i, c_i, f_i to find the optimized parameters for each CPD separately.

Note that this parameter learning process of the DBN is performed offline during the training session and hence will not affect the functionality of the system and the user experience during the real-time

operation. A separate trust model should be trained based on each user's experience using their respective trust input data m_i, c_i, f_i as well as the performance measures $P_{H,i}, P_{R,i}, F_i$.

4 TRUST-BASED SPECIFICATION DECOMPOSITION

Available methods for multi-robot symbolic motion planning have mainly focused on fully autonomous systems and can be summarized into two main types: centralized and decentralized solution approaches. Centralized solutions treat the robot team as a whole and have a large global state space formed by taking the product of the state spaces of all the robots [41, 42], which is too large to handle in practice. Decentralized solutions tend to give local specifications to individual robots, which results in a smaller state space but often sacrifices guarantees on global performance [21]. Here, we propose a distributed solution for human-robot symbolic motion planning. Ideally, if all the obstacles are known beforehand and the speed profile of each robot is given, optimal and collision-free paths can be designed offline without the necessity of collaboration. In this work, we consider a gradually learned environment with intermittent communications when robots are within the communication range, which therefore requires collaboration between robots for obstacle and collision avoidance. We first present a method for addressing the collision avoidance task in Section 4.1 and then a method for decomposing the specification for individual tasks in Section 4.2. In Section 6, deadlock- and livelock-free algorithms are developed to guarantee that all goals are reached while all obstacles and robot collisions are avoided with a human-in-the-loop.

4.1 Specification Updates Based on Atomic Propositions for Observation, Communication, and Control

Here we introduce the atomic propositions π_i^o , π_{ij}^c , and π_i^u in (4) for each robot i , which correspond to observation, communication, and control. All these atomic propositions (APs) are dynamically checked and updated based on the robots' sensing. A similar approach has been used in decentralized multi-robot tasking in [21], and here we extend it to distributed multi-robot systems that must meet a global specification. The observation proposition π_{ij}^o for robot i is true if an obstacle j in workspace cell w_j is within its sensing range r_i and false otherwise:

$$\pi_{ij}^o t = \begin{cases} \|x_{it} - x_o^j\| \leq r_i & \text{true} \\ \|x_{it} - x_o^j\| > r_i & \text{false} \end{cases}, j \in Obs,$$

where x_o^j represents the actual position of an obstacle j . When π_{ij}^o is true, robot i senses an obstacle j along its planned path. The robot then updates its model of the workspace by modifying its transition system representation TS_i to label the state representing workspace cell w_j with proposition π_j^b . The robot then resynthesizes its plan using the updated version of TS_i according to the procedure described in Section 2.2, which then guarantees the specification $\Box \pi_{ij}^o \rightarrow \neg \bigcirc \pi_j^b$.

The communication proposition π_i^c for robot i is true if another robot j is within its communication range ρ_i and false otherwise:

$$\pi_i^c t = \begin{cases} \|x_{it} - x_{jt}\| \leq \rho_i & \text{true} \\ \|x_{it} - x_{jt}\| > \rho_i & \text{false} \end{cases}, j \neq i, = 1, 2, \dots, N.$$

When π_i^c is true, robots i and j can communicate with each other to exchange goal assignment, obstacle, and planned path information, allowing them to learn features of the environment they have not yet explored themselves and resynthesize their plans to avoid obstacles if necessary. This information can also be used to detect possible collisions between the two robots.

We next introduce the control proposition. When π_i^u is true, robot i is executing the nominal LQR control law (2); when false, the robot pauses or replans its path autonomously:

$$u_i t = \begin{cases} \text{LQR} & \pi_i^u t \\ \text{wait or replan} & \neg \pi_i^u t \end{cases}.$$

When the communication proposition π_i^c is true, robot i has detected a potential collision and communicates its path with involved robot j . At this moment, the control proposition π_i^u is set to false, and the robot takes one of two actions dependent upon the collision type to guarantee the specification $\Box \pi_i^c \rightarrow \neg \bigcirc \pi_i^u$. Both waiting and replanning can be accomplished autonomously.

Fig. 5 illustrates examples for possible collision scenarios. The robots' communication ranges are represented using dashed circles and their positions by solid circles. The arrows represent the path information communicated among the robots. We outline the collision avoidance method using the APs for communication and control (i.e., $\bigwedge_{i=1}^N \Box \pi_i^c \rightarrow \neg \bigcirc \pi_i^u$) as follows.

ASSUMPTION 4.1. *Under the collision avoidance scenarios, without loss of generality, we assume that a robot with the smaller index $i \in I_R$ will always have higher priority in replanning and stop first if necessary. Similar prioritization policies can be imposed and the rest of the method will not be affected.*

Fig. 5(a) demonstrates the collision scenario when two robots will enter into a same cell at the next step. In this case, Robot 1 will stop and wait for Robot 2's next move according to our prioritization policy. Fig. 5(b) shows the scenario when two robots are at adjacent cells and will run into each other at the next step. In this case, both robots will replan to avoid collision by taking the other robot as an obstacle. Fig. 5(c) illustrates the scenario when three robots will run into a same cell at the next step. In this case, Robot 1 will stop first and Robot 2 and 3 will either stop or replan based on their collision type (i.e., the scenario shown in either Fig. 5(a) or 5(b)). Note that this simple protocol is not guaranteed to be deadlock-free in arbitrary environments. For instance, if two robots must pass by each other through a 1-cell wide "corridor" of cells between two sets of obstacles and there is no other path to their goals, they will not be able to reach them unless they trade goals, leading to deadlock. Even if trading goals between robots is enabled, then over the course of multiple collision avoidance replanning events, robots can potentially trade and re-trade goals in such a way that some set of the goals are not ever reached, leading to livelock. In Section 6, we present a more sophisticated protocol that is guaranteed to be both deadlock- and livelock-free.

4.2 Compositional Reasoning for Reachability and Obstacle Avoidance

A general compositional reasoning approach can be used to show that the robots are able to collectively fulfill the reachability and obstacle avoidance portions of the global specification using a distributed planning approach. Note that this reasoning neglects the possibility of inter-robot collisions and switching to human control, which are addressed separately in Sections 4.1, and 5, and an overall solution will be provided in Section 6. Compositional reasoning in this context relies on concepts of interleaving of transition systems and unconditional fairness, which are defined as follows.

Definition 4.2 (Interleaving of Transition Systems). The interleaving of two transition systems $TS_i = S_i, Act_i, \delta_i, I_i, \Pi_i, L_i, i = 1, 2$ is defined as $TS_1 ||| TS_2 = S_1 \times S_2, Act_1 \cup Act_2, \delta, I_1 \times I_2, \Pi_1 \cup \Pi_2, L$, where transition relation δ is $s_1, s_2 \rightarrow s'_1, s_2$ if $s_1 \xrightarrow{\alpha} s'_1$ for $\alpha \in Act_1$ and $s_1, s_2 \rightarrow s_1, s'_2$ if $s_2 \xrightarrow{\alpha} s'_2$ for $\alpha \in Act_2$, and labeling function L is $Ls_1, s_2 = Ls_1 \cup Ls_2$.

Definition 4.3 (Unconditional Fairness). A path $\sigma = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \dots$ of a transition system TS is unconditionally fair with respect to a set of actions $A \subseteq 2^{Act}$ if it contains infinitely many instances

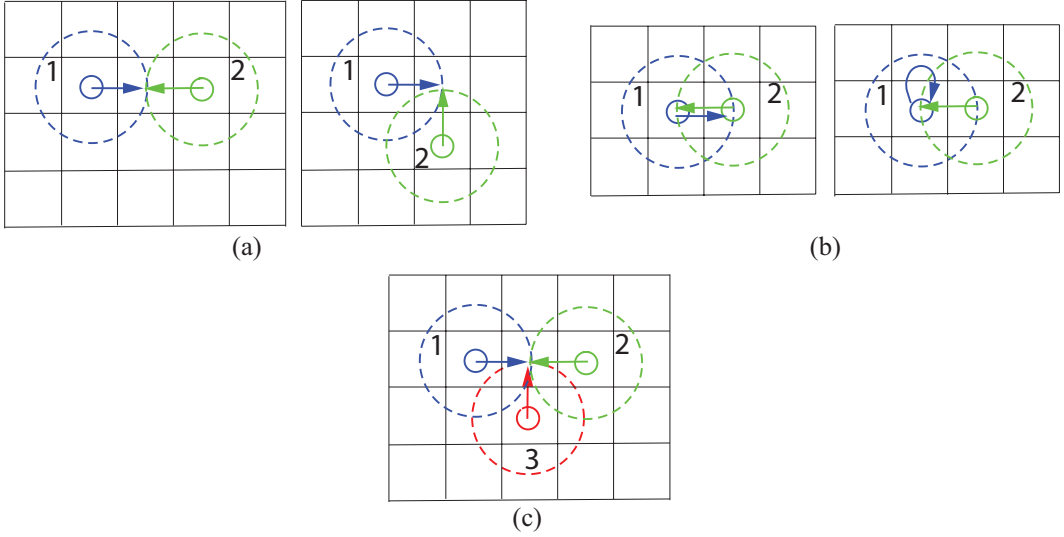


Fig. 5. Collision avoidance scenarios.

of each action $\alpha \in A$. A transition system TS satisfies φ under an unconditional fairness assumption $\mathcal{F} \subseteq 2^{Act}$, denoted $TS \models_{\mathcal{F}} \varphi$, if all paths in TS that are unconditionally fair with respect to \mathcal{F} satisfy φ .

Recall that, neglecting inter-robot collisions and switching to manual motion planning, each robot i will eventually complete execution of a specific path σ^i that reaches its assigned goals while avoiding all obstacles according to the procedure and reasoning given in Section 2.2. Note that this can be seen as a special case of a transition system comprising only one path. Let us then represent this path as transition system TS_{σ}^i and note that $TS_{\sigma}^i \models j \in Goals_i \diamond \pi_j^g \wedge \bigwedge_{i=1}^N j \in Obs \square \pi_{ij}^o \rightarrow \neg \bigcirc \pi_j^b$. Assuming each robot can take an infinite number of actions over time – a natural assumption since the robots act independently – we have the following lemma.

LEMMA 4.4. *Assuming there exists at least one path to each assigned goal, there are no inter-robot collisions, and there is no switching to manual motion planning, the robots are guaranteed to eventually reach all assigned goals while avoiding collisions with obstacles under an assumption of unconditional fairness.*

PROOF. The combination of all robots acting in the workspace together is $TS_{RW} = TS_{\sigma}^1 ||| \dots ||| TS_{\sigma}^N$. Assuming the robots act independently, each robot is able to take an infinite number of actions, and all actual execution paths of TS_{RW} will be unconditionally fair with respect to $\mathcal{F} = \bigcup_{i=1}^N Act_i$.

Note that every path in TS_{RW} is formed by interleaving the individual paths represented by each TS_{σ}^i where $\sigma_i \models j \in Goals_i \diamond \pi_j^g$. Then $TS_{RW} \models_{\mathcal{F}} j \in Goals \diamond \pi_j^g$, since all actions $\alpha^i \in Act_i$ that enable each TS_{σ}^i to reach states satisfying each goal $\pi_j^g \in Goals_i$ will be executed, and $\bigcup_{i \in I_R} Goals_i = Goals$. Furthermore, since all robots satisfy $TS_{\sigma}^i \models j \in Obs \square \pi_i^c \rightarrow \neg \bigcirc \pi_j^b$, then $TS_{RW} \models \bigwedge_{i=1}^N j \in Obs \square \pi_i^c \rightarrow \neg \bigcirc \pi_j^b$, and there are no states for any TS_{σ}^i in which π_j^b for $j \in Obs$ holds. \square

REMARK 1. *The trust belief estimate based on Eq. (10) will determine the specification decomposition, with more trusted robots assigned more destinations. Since trust is dynamically involving, this robot assignment will be updated on the fly.* \bullet

5 REAL-TIME TRUST-BASED SWITCHING BETWEEN MANUAL AND AUTONOMOUS MOTION PLANNING

In this section, we utilize trust analysis in a real-time switching framework to enable switches between manual and autonomous motion planning. Recall from Section 2 that although the autonomous motion planning is guaranteed to be correct, it is usually conservative due to overapproximation of the environment. So while the autonomous motion planning is safe, more efficient but riskier paths – in this case, paths between obstacles in adjacent regions – may exist. If the human trusts a robot’s ability in navigating between two obstacles, the human can choose to construct a more efficient path between the obstacles based on, e.g., sensory information about the obstacles supplied by the robot.

REMARK 2. *Under the autonomous motion planning mode, to guarantee the bisimulation between high-level discrete motion planning and low-level continuous control, we utilize a nominal LQR controller (2), and hence a robot moves between centroids of adjacent cells. That is to say, the cells are assumed to be four-cornered and a robot can only move “left”, “right”, “up”, and “down” at the next step. However, under the manual motion planning mode, there is no such restriction and a robot can move in any direction following the human assigned waypoints.* •

Fig. 6 shows an example of how trust can benefit the motion planning of an autonomous robot, generated by integrating the NuSMV model checker with Matlab. The environment of the system is represented in Fig. 6. The robot begins in cell 5, represented by the circle, and must reach the goal in cell 15, represented by the diamond. The obstacles are represented in the continuous space by filled polygons and in the discrete space by cells marked with “X”s. As Fig. 6(a) shows, the autonomous motion planner avoids regions marked by Xs and generates a safe but lengthy path to the goal; while there is a gap between the two obstacles through cell 10, the overapproximation prevents the robot from generating a path through this region. If the human operator trusts the robot’s ability to follow a path between the obstacles and is not overloaded, the manual motion planning mode will be activated and the benefits can clearly be seen in Fig. 6(b). This tradeoff between safety and efficiency motivates the use of trust-based switching between manual and autonomous motion planning.

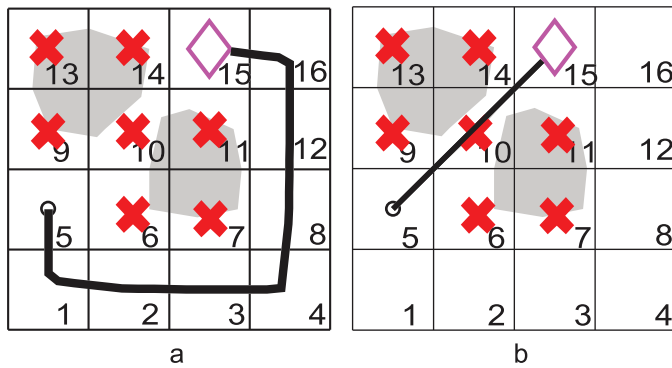


Fig. 6. (a) Safe autonomous motion planning in low trust scenario, and (b) advanced manual motion planning in high trust scenario.

This high trust path, however, is inherently risky since information regarding the environment in a typical scenario is limited to what the robot has sensed and provided to the human through visual feedback. It is therefore necessary to develop a monitoring and switching framework such that the

autonomous safe motion planning can be reactivated if any new obstacle in path is detected and a collision is about to happen under the path planned by the human. The overall architecture of our proposed framework is based on two frameworks which were originally designed for augmenting system safety. The first framework is called the Simplex architecture, which guarantees application-level safety by “using simplicity to control complexity” [7, 61], and the second one is called the Monitoring and Checking (MaC) framework, which “bridges the gap between formal specification and verification” and was developed originally to assure the correctness of program execution at runtime [39]. Fig. 7 shows the schematic of the proposed monitoring and trust-based switching framework. This framework is logically divided into five subsystems: Motion Planner, Controller, Monitor, Checker, and Decision Maker. This system is designed to be able to control the robot in two modes: Manual versus Autonomous. The advanced sub-system, i.e., manual mode, is less safe since it allows the human to plan riskier paths. The baseline subsystem refers to the autonomous mode and uses the symbolic motion planner, which is guaranteed to be correct and hence safe. In the Monitor subsystem, we have two modules: Filter and Event Recognizer. The filter is designed to extract the motion state information of a robot and send it to the event recognizer. The event recognizer detects an event from the values received from the filter based on event definitions provided by a monitoring script. The monitoring script maps the system states to the events at the requirement level to enable analysis by the system checker. Here, events are defined according to the condition $\|x_i - x_o^j\| \leq r_o$, where x_o^j is the obstacle position and r_o is some minimally acceptable distance between the robot and the obstacle. Should the robot come within this distance, an event will be detected. According to the mean trust equation (6), this event detection also leads to fault penalty and hence lowers trust in the robot, leading to a re-evaluation of the assigned tasks. The result is that other more trusted robots may be re-assigned some of the destinations that were originally assigned to the robot that generated the fault. Once this re-evaluation is performed, the operator is free to continue working with the same or another robot depending on the change of levels of trust. Once the event recognizer detects an event, it will send the information to the checker module. The checker checks whether or not the current execution of the system meets the specification. Based on the information received from the checker, the decision module determines under which mode the system should run for motion planning. More specifically, the trust-based decision module first computes the trust belief distribution based on Equation (10) for each robot i , then the corresponding trust value that yields the maximum likelihood is obtained. Next, the current maximum likelihood trust is compared with the maximum likelihood trust at the previous time step and the change of trust value can be calculated. The decision module will suggest that the human collaborate with the robot that has the highest trust increase beyond a certain threshold. In case multiple robots have the same highest trust change, some priority criterion can be used to choose an individual robot. See Fig. 16 for an example of the GUI design used in our simulation for a robot requesting manual motion planning based on trust comparison.

REMARK 3. *From the proof of Lemma 4.4 it is shown that given no collision among robots, each robot will generate a path that will eventually guarantee reaching its goals, with the bisimulation property given in Section 2 guaranteeing the robot will follow this path correctly. The human generated plans provide a more efficient way of reaching goals, but can be risky if there is incomplete environmental information. However, using the monitoring and switching framework outlined above, the robot is guaranteed to never collide with an obstacle and will return to the proven safe method. This switching will continue until all goals are reached or a complete obstacle map is generated. Assuming that the human is guaranteed to never permanently take control of one of the robots, the fairness assumption used in Lemma 4.4 is satisfied.* •

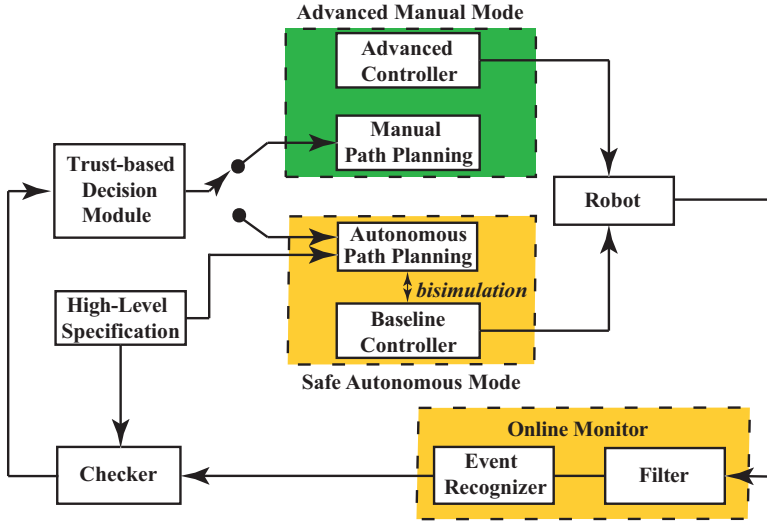


Fig. 7. The overall structure of the monitoring and trust-based switching framework

6 DEADLOCK- AND LIVELOCK-FREE ALGORITHMS

As discussed briefly in Section 4.1, the previously described collision avoidance protocol does not prevent deadlock, e.g., cases in which two or more robots cannot move toward their goals because they are blocking each other's paths. Naively allowing robots to trade goals in such a case could result in livelock situations [55], in which robots could be cyclically re-trading goals over multiple encounters while never actually reaching them. To address the problem, we propose the following deadlock- and livelock-free algorithms, which collectively guarantee the reachability property $\prod_{j \in \text{Goals}} \Diamond \pi_j^g$, i.e., that all goals are eventually visited by a robot.

Let each robot $i \in I_R$ have a “current” goal $g_i \in \text{Goals}$ and a set of “next” goals $\text{Goals}'_i \subset \text{Goals}$ such that $g_i \cap \text{Goals}'_i = \emptyset$, $\text{Goals}'_i \neq \emptyset \rightarrow g_i \neq \emptyset$, and the union of all current and next goals for all robots is $G_R = \bigcup_{i \in I_R} g_i \cup \text{Goals}'_i$, with $G_R = \text{Goals}$ at initialization. Let us also suppose that $g_i \cup \text{Goals}'_i \cap g_j \cup \text{Goals}'_j = \emptyset$ for all $i, j \in I_R$ where $i \neq j$. This last condition is not strictly necessary to ensure that all goals are eventually reached, but it helps avoid duplicate visits to goals by multiple robots and the need to remove a goal from a robot's set of goals if another robot is observed at the goal location. Denote the planned path for robot i as $\sigma^i = s_0^i \xrightarrow{\alpha_0^i} s_1^i \xrightarrow{\alpha_1^i} s_2^i \dots$ and let $\sigma^{ik..}$ denote the suffix of σ^i starting at k , i.e., $s_k^i \xrightarrow{\alpha_k^i} s_{k+1}^i \xrightarrow{\alpha_{k+1}^i} s_{k+2}^i \dots$. Then deadlock- and livelock-free algorithms that ensure all the goals are reached under a set of mild assumptions are as follows.

The assumptions and proof of correctness are as follows.

ASSUMPTION 6.1. *There is an obstacle-free path of finite length between every pair of obstacle-free cells.*

ASSUMPTION 6.2. *For every robot $i \in I_R$, the communication range ρ_i is greater than the size of any single cell.*

ASSUMPTION 6.3. *The human operator can only choose to control a robot if the robot requests to move through a set of obstacles. Human control of the robot ends in finite time.*

ALGORITHM 1: Deadlock- and livelock-free, human-in-the-loop symbolic motion planning $\forall i \in I_R$

Input: $Goals'_i \subset Goals$ and $g_i \in Goals$ such that $g_i \cap Goals'_i = \emptyset$, $Goals'_i \neq \emptyset \rightarrow g_i \neq \emptyset$, and the union of all current and next goals for all robots G_R , with $G_R = Goals$ at initialization.

Output: Deadlock- and livelock-free, human-in-the-loop symbolic path planning for every $i \in I_R$

```

for  $i++$  do
  Plan Path(); //ALGORITHM 2
  repeat
    Follow Path(); //ALGORITHM 3
  until  $false$ ;
end

```

ALGORITHM 2: Plan Path()

if $g_i \neq \emptyset$ and no other robots are in communication range **then**

 Plan a path σ^i to g_i ;

else
if $g_i \neq \emptyset$ and other robots $I \subseteq I_R$ are in communication range **then**

 Ensure that for $g^* = \min\{g_i \mid i \in I\}$, the currently assigned robot i can reach it given the position of the other robots, or it is assigned to the robot j closest to g^* such that $\sigma^j = \sigma^i k..$ for some $k > 0$;

 Reassign all other goals in $_{i \in I} Goals'_i \cup g_i \setminus g^*$ to robots in I ;

end
end

ASSUMPTION 6.4. Every robot executes actions infinitely often, where remaining in the current cell for a finite amount of time is considered valid action. All actions complete in finite time.

LEMMA 6.5. Every robot i is able to plan a path σ^i to its current goal g_i such that the number of actions needed to reach g_i is finite, thus preventing deadlock.

PROOF. This follows from Assumption 6.1 in the case that the planned path does not include any obstacle cells, additionally from Assumption 6.3 if the planned path does include obstacle cells due to the human engaging manual planning, and the fact that when multiple robots are in communication range, the inter-robot collision protocol never assigns a current goal g_i to a robot that cannot reach it given the current position of the other robots. \square

LEMMA 6.6. Whenever two or more robots with indices $I \subseteq I_R$ cooperatively replan their paths according to the inter-robot collision protocol, if the goal $g^* = \min\{g_i \mid i \in I\}$ is reassigned, the number of actions needed by the newly assigned robot to reach it decreases, thus preventing livelock.

PROOF. Due to Assumption 6.1, if the robot i assigned to g^* is unable to continue on its planned path to g^* , it can only be because one or more other robots occupy cells on this path. The “Path Plan” algorithm reassigns g^* to the robot j that requires the least number of cell transitions to reach g^* along the original path of robot i . By Assumption 6.1, there exists a valid path $\sigma^j = \sigma^i k..$ for some $k > 0$, and the number of actions in the path σ^j to the goal cell g^* is less than the original path σ^i . \square

THEOREM 6.7 (REACHABILITY). Every goal is eventually reached by a robot.

PROOF. Initially, let G_R be the union of all current and next goals for all robots, i.e., $G_R = \cup_{i \in I_R} g_i \cup Goals_i = Goals$. The proof is by induction.

ALGORITHM 3: Follow Path()

```

repeat
    Drive robot using the LQR law (2);
until Obstacles or other robots in range, or human intervention;
if Obstacle in observation range  $r_i$  then
    Record obstacle;
    if Obstacle in path then
        Plan Path(); //ALGORITHM 2
    end
    Follow Path(); //ALGORITHM 3
end
if Other robots in communication range  $\rho_i$  then
    Share obstacle and path data;
    Plan Path(); //ALGORITHM 2
    Follow Path(); //ALGORITHM 3
end
if Human control requested and granted then
    repeat
        if Obstacle too close then
            release human control;
            break;
        end
        if Exited obstacle region then
            Release human control;
            break;
        end
        Plan Path(); //ALGORITHM 2
        Follow Path(); //ALGORITHM 3
    until Human is not in control;
end
if  $g_i$  reached then
    if  $Goals'_i \neq \emptyset$  then
        Set  $g_i$  to a goal in  $Goals'_i$ ;
        Set  $Goals'_i$  to  $Goals'_i \setminus g_i$ ;
    end
    if  $Goals'_i = \emptyset$  then
        break;
    end
end

```

Base case: $g_i = \min G_R$ for some robot i . Either robot i reaches g_i by Lemma 6.5 or some other robot reaches it by Lemma 6.6. G_R is then set to $G_R \setminus g_i$, decreasing the size of G_R by 1.

Induction: $g_i > \min G_R$ for some robot i . Either robot i reaches g_i by Lemma 6.5, G_R is then set to $G_R \setminus g_i$, and the size of G_R reduces by 1, or the robot enters the collision avoidance protocol with a set of robots and one of those other robots continues progress on g_i or some $g_j < g_i$ by Lemma 6.6.

Since G_R is finite, eventually $G_R = \emptyset$ and all goals are reached. \square

Under the above deadlock- and livelock-free algorithms, we can resolve collision avoidance among 3 robots by goal reassignment, a problem we could not previously solve using a more simple protocol. We illustrate the solution approach as follows.

Fig. 8 shows two example collision scenarios among 3 robots. Robots 1, 2, 3 are within each other's communication range and hence can exchange obstacle, planned path, as well as goal assignment information. As illustrated in the figure, Robot 1 and 2 will run into the same cell at the next move (i.e., the collision type depicted in Fig. 5(a)); Robot 2 and 3 are at adjacent cells and will run into each other at the next move (i.e., the collision type depicted in Fig. 5(b)). In this case, simply letting one of the robots wait and the other two replan might trap the robots in a deadlock situation. Instead, we can exchange the robots goals (i.e., assign the goals to robots that travel from cells closest to them) as in algorithms 1-3 to resolve the possible collision. Other similar collision scenarios among 3 robots can be resolved using the same path replan and goal reassignment methods.

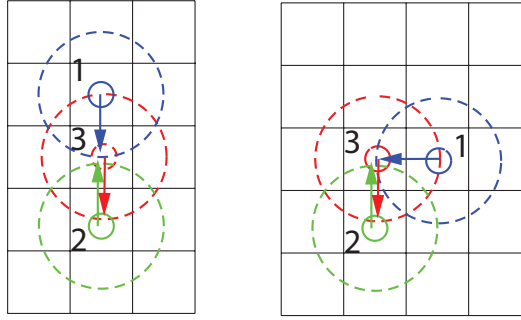


Fig. 8. Example collision scenarios among 3 robots: Robot 1 and 2 will run into a same cell at the next move; Robot 2 and 3 are at adjacent cells and will run into each other at the next move.

Fig. 9 shows another two example collision scenarios among 3 robots. At step a, Robot 2 and 3 will run into a same cell at the next move (i.e., the collision type depicted in Fig. 5(a)); Robot 1 follows Robot 2 and there is no collision. According to Assumption 4.1, a robot with smaller index number will stop first, therefore Robot 2 will stop and wait for Robot 3's next move. However under this policy, at step b, Robot 1 and 2 will collide with each other. In this case, the robots' goals will be reassigned according to the above algorithms to resolve the possible collision. Other similar collision scenarios among 3 robots can be resolved using the same methods.

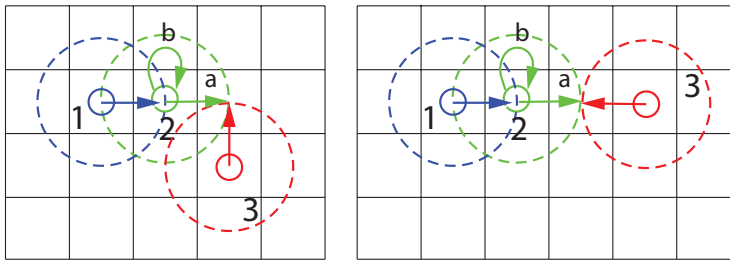


Fig. 9. Example collision scenarios among 3 robots: Robot 2 and 3 will run into a same cell at the next move and hence Robot 2 waits, however under this policy Robot 1 and 2 will collide at the next move.

Livelock is harder to visualize since it is not due to single collision avoidance event but rather the possibility of goals being cyclically reassigned over the course of multiple collision avoidance events. It is also rare since it would require just the right circumstances in regards to the positioning of obstacles and the exact reassignment of goals during each collision event. Theorem 6.7 suffices to show that the algorithms are livelock-free.

7 SIMULATION

In this section, a set of simulations of the ISR scenario of Section 2 is used to demonstrate our methods. The simulation is conducted in Matlab with model checking performed using NuSMV. We show the simulation results for distributed symbolic motion planning for multiple autonomous robots under the proposed obstacle and inter-robot collision avoidance protocols in Section 7.1, the human input devices, manual motion planning approach, and GUI designs for trust measurements in Section 7.2, and the overall trust-based motion planning strategy with a human-in-the-loop in Section 7.3.

7.1 Distributed Motion Planning for Multiple Autonomous Robots

Fig. 11 shows an example environment that consists of 3 robots with marked index numbers, 6 goals marked by diamonds, and 12 obstacles marked by crosses. The obstacles in the environment are initially unknown by the robots until they are gradually sensed. The sensor range r_i of a robot is marked by a dashed circle around it. In our simulations, we set the range in a way such that a robot can always observe the 8 neighboring cells around it. Once an obstacle is sensed, its position becomes known to that individual robot. The communication range ρ_i of a robot is set to be the same as the sensing range. The obstacle information, planned path, and goal assignment information are communicated with other robots when they come within communication range. The robot paths are demonstrated using bold line segments. The robot start positions are marked by rectangles and the robot current positions are marked by circles.

To find a path such that the local subspecification is satisfied, the symbolic motion planning approach described in Section 2.2 is used by each individual robot. To address computational complexity, the compositional reasoning concept is used to decompose (4) into local subspecifications as described in Section 4.2. To further reduce computational complexity, each robot only computes its path over a local subset (see the black boxes in Figs. 10 and 12). This subset is determined using the robot's knowledge of its assigned goals and the location of obstacles that have been sensed. Using this method, every path generated by an individual robot will end in either reaching its goals or detecting an obstacle and will continue until all obstacles are detected. To prevent collisions among the robots, the communication and control protocols detailed in Section 4.1 and the deadlock- and livelock-free algorithms are used whenever the robots are within each other's communication range. Therefore, the robots are guaranteed to eventually reach their goals.

Fig. 10 shows the motion planning process to avoid the collision scenario as shown in Fig. 8. Fig. 11 shows the final path of each robot and marks the key moments (a)-(d) corresponding to time step $t = 1$ to $t = 450$ as illustrated in Fig. 10. Fig. 12 shows the motion planning process to avoid the collision scenario as shown in Fig. 9. Fig. 13 shows the final path of each robot and marks the key moments as illustrated in Fig. 12. The motion planning process for the collision scenarios depicted in Fig. 5 can be demonstrated in a similar fashion.

7.2 Human Input Devices, Manual Motion Planning, and GUI Design

A human subject can choose to collaborate with a robot using a gamepad (see Fig. 14) and replan a robot path using a mouse (see Fig. 15). Fig. 16 shows the GUI designs used in the simulation for human intervention and collaboration. Fig. 16(a) shows the dynamic evolution of the maximum

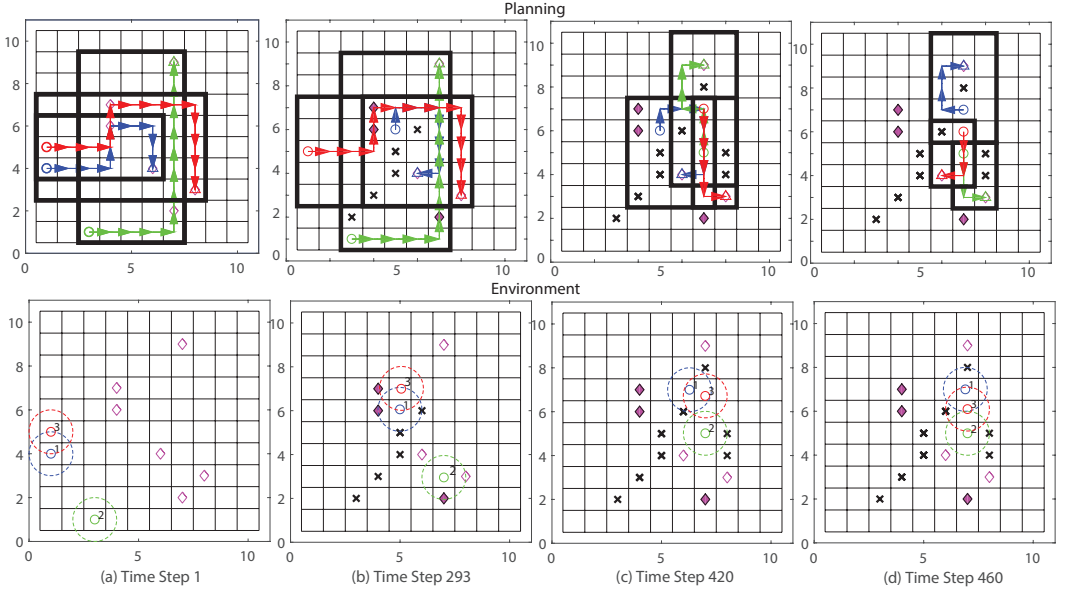


Fig. 10. Progression of simulation of the inter-robot collision scenario in Fig. 8 (a) at time step $t = 1$ (initial plan and robot position), (b) $t = 293$ (new obstacle at (6,6) detected in the path and Robot 1 replans), (c) $t = 420$ (Robots 2 and 3 communicate each other's obstacle information and replan), and (d) $t = 450$ (Robots 1,2, and 3 enter into the collision scenario in Fig. 8 and hence reassign goals and replan.)

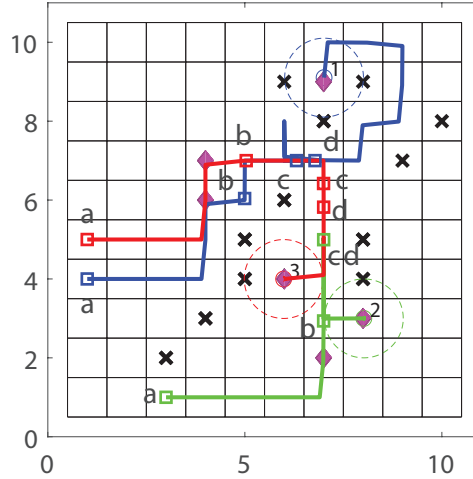


Fig. 11. Final paths and key moments (marked by rectangles from step (a)-(d)) of each robot under the simulation scenario shown in Fig. 10.

likelihood trust for all robots and compares the change of trust with a preset threshold. Once the change of maximum likelihood trust exceeds the threshold, the robot with the largest trust increase will request manual motion planning (e.g., Robot 3 requests that the human chooses a path for it as

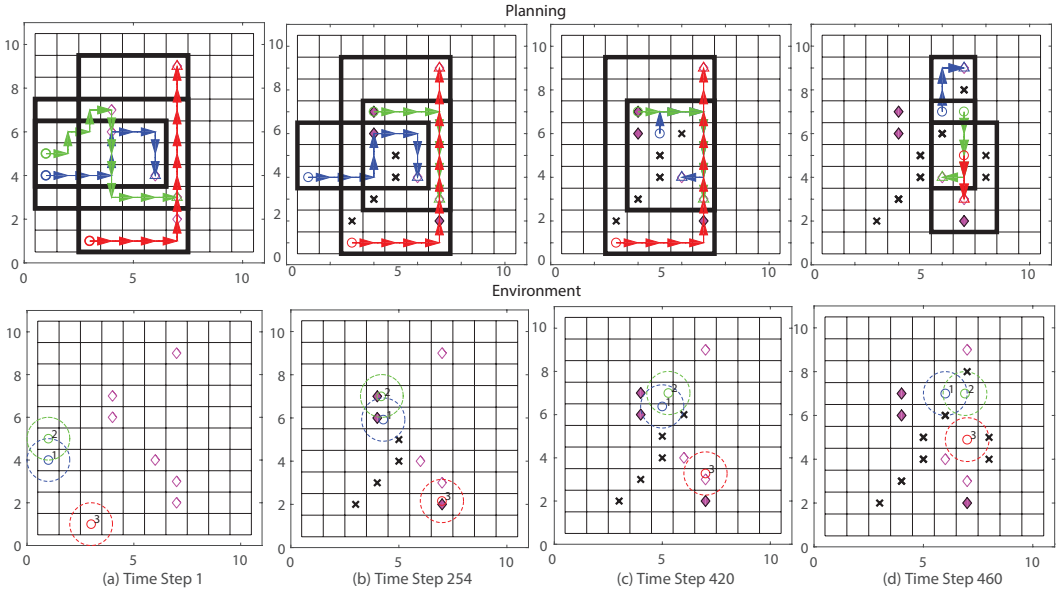


Fig. 12. Progression of simulation of the inter-robot collision scenario in Fig. 9 (a) at time step $t = 1$ (initial plan and robot position), (b) $t = 250$ (Robots 1 and 2 are within each other's communication ranges; Robot 2 finds new obstacles in the path and hence replans), $t = 310$ (new obstacle at (6,6) detected in the path and Robot 1 replans), and (d) $t = 391$ (Robots 1,2, and 3 enter into the collision scenario in Fig. 9 and hence reassign goals and replan.)

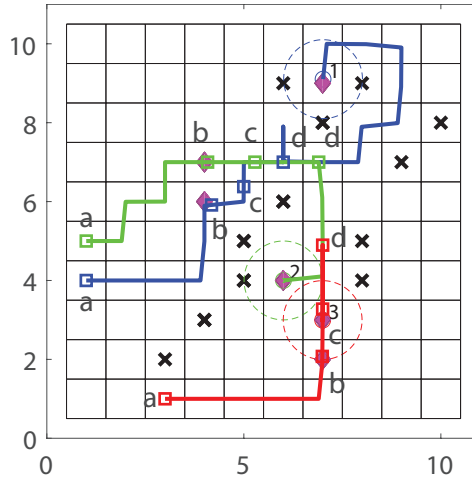


Fig. 13. Final paths and key moments (marked by rectangles from step (a)-(d)) of each robot under the simulation scenario shown in Fig. 12.

shown in the figure where the threshold is set as 0.07). Fig. 16(b) shows the GUI for measuring the trust change c_i to be used in the calculation of trust belief (10) for all robots where “Lose” corresponds to “-1”, “Unchange” corresponds to “0”, and “Gain” corresponds to “+1”. Table 1 shows



Fig. 14. Human input devices: gamepad and mouse. The gamepad buttons are used to select and confirm a robot to collaborate with, e.g., “blue” button for Robot 1, “green” button for Robot 2, “red” button for Robot 3, and “yellow” button for releasing the human. The mouse is used to give waypoints to a robot manually.

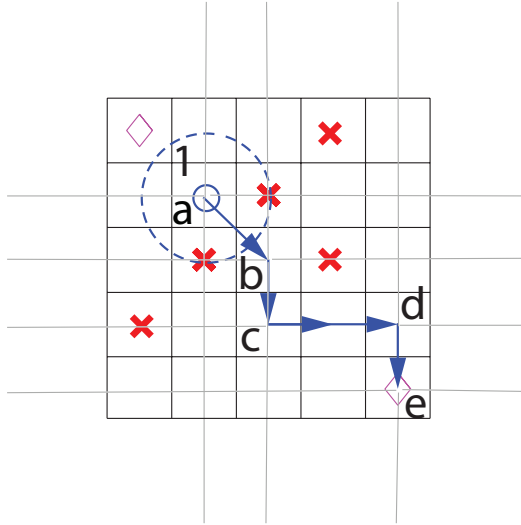
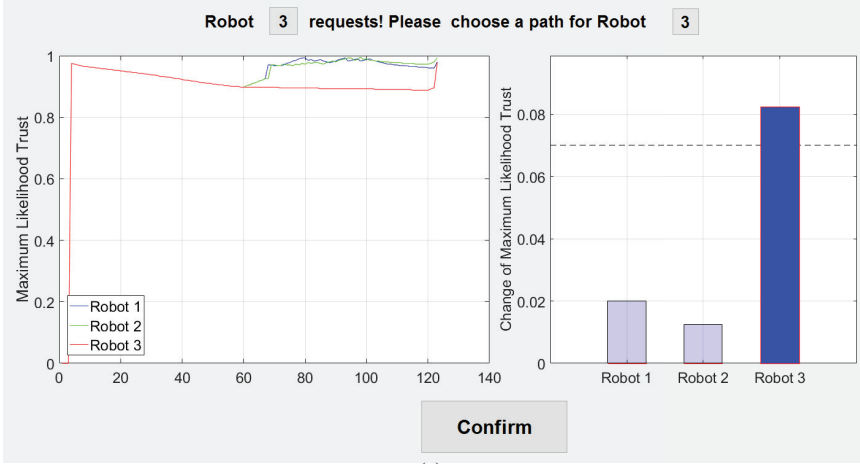


Fig. 15. Illustration of manual motion planning using a mouse. The human assigns waypoints starting from a, passing between two obstacles to point b, and then point c, d, and eventually reaching the goal e.

the detailed instructions given to the human operator to interpret the GUI. This GUI measure is shown to the human subject every 35 time steps in the simulation. Fig. 16(c) shows the GUI for measuring the trust feedback f_i , where “full distrust”, “medium trust”, “neutral”, “medium trust”, and “full trust” span the spectrum from 0 to 1 and is a continuous scale. This GUI measure is shown to the human subject every 100 time steps. After 100 time steps, the operator is asked to provide an estimation of their degree of trust towards each robot in the system based on his/her cumulative interaction with the individual robot. The gaps of 35 and 100 time steps are selected not only to ensure that the operator



Trust Change?

Robot 1 ☐ Lose ☒ Unchange ☐ Gain

Robot 2 ☐ Lose ☒ Unchange ☐ Gain

Robot 3 ☐ Lose ☒ Unchange ☐ Gain

Confirm

(b)

Detailed description: This GUI design (b) is titled 'Trust Change?'. It contains three rows, one for each robot. Each row has a radio button for 'Lose', 'Unchange', and 'Gain'. For all three robots, the 'Unchange' option is selected. A 'Confirm' button is at the bottom.

What is your degree of trust in the robot right now?

Robot 1

Robot 2

Robot 3

full distrust distrust neutral trust full trust

Confirm

(c)

Detailed description: This GUI design (c) is titled 'What is your degree of trust in the robot right now?'. It contains three rows, one for each robot, each with a horizontal range slider. Below the sliders are five labels: 'full distrust', 'distrust', 'neutral', 'trust', and 'full trust'. A 'Confirm' button is at the bottom.

Fig. 16. GUI designs: (a) Comparison of robots' maximum likelihood trust and request for manual motion planning, (b) Measure of trust change c_i , and (c) Measure of trust feedback f_i .

will not be overwhelmed, but also to collect sufficient human trust change and trust level data. For example, a gap of 25 seconds is selected for trust evaluation of 1 robot in [16].

Table 1. Instructions for GUI in Fig. 16(b)

YOUR OBSERVATION	PERFORMANCE CHANGE	ACTION ON GUI
A robot detects an obstacle besides its path and does not need to change path	Robot performance increases	Choose “Gain”
A robot reaches a goal destination	Robot performance increases	Choose “Gain”
A robot senses an obstacle on its path and needs to change path	Robot performance decreases	Choose “Lose”
A robot senses another robot on its path and needs to change path	Robot performance decreases	Choose “Lose”
An unknown obstacle on human-planned path and activate auto. planning	Robot performance decreases	Choose “Lose”
No change	Robot performance unchanged	Choose “Unchange”

7.3 Overall Simulation for Trust-Based Multi-Robot Motion Planning with a Human-in-the-Loop

Before implementation of the overall strategy, we run a training session. The purpose is to collect the human input data and performance measures for parameter identification of the trust model. The elapsed time for the training session is 158.59 seconds. After the training session, we run four iterations of the EM algorithm (256.12 seconds) to learn the parameters until the errors converge within the limit 1×10^{-10} .

We finally show the simulation results of 3 robot symbolic motion planning with a human-in-the-loop. The goal of the human-robot team is to successfully reach each goal while avoiding all collisions, meeting the global specification (4). For this scenario, trust levels are assumed to be equal at the start of the simulation. Fig. 17 shows the final paths traveled by all robots under the trust-based switching framework. Note that Robot 1 plans a path passing through obstacles for task efficiency. Fig. 18 shows the evolution of mean trust \bar{T}_i , $i = 1, 2, 3$ over time (Equation (6)). Figs. 19-21 show the human performance $P_{H,i}$, robot performance $P_{R,i}$, fault F_i , trust belief $belT_i$, human intervention m_i , trust change c_i , and trust feedback f_i for Robot 1-3.

Initially, Robot 1 is assigned to goals (8,3) and (4,6), Robot 2 to goals (6,2), (7,9), and Robot 3 to goals (10,4), (4,7), (10,10). At the beginning, Robot 1 senses an obstacle on its path and Robot 2 senses an obstacle on its left side. Therefore, in the first trust change question (see Fig. 16(b)), Robot 1’s trust decreases by 1 and Robot 2’s trust increases by 1. That is to say, in the trust question, when a robot senses an obstacle on its path, the human subject should choose “Gain”. When a robot senses an obstacle on its path or meets another robot and hence changes its path, one should choose “Lose”. Otherwise, one should keep choosing “Unchange”.

When all three robots reach their first goals at time step $t = 124$, the trust increase of Robot 3 goes beyond the threshold (set as 0.05 here). Hence, Robot 3 requests the human to intervene (see Fig. 16(a)) and the human plans a new path for Robot 3 (as shown in Fig. 21(c) with $m_3t = 124 = 1$). However, there is an unknown obstacle in the path. Once this unknown obstacle is sensed, a fault occurs at $t = 246$ (Fig. 21(a)) and Robot 3’s trust drops (Fig. 21(b)). Robot 3 then gives its current goal (10,4) to Robot 1 (whose trust change is the highest at this moment). Robot 3 continues to its last left goal (10,10) and stops and stays there at $t = 342$.

For Robot 1, as shown in Fig. 19, $P_{R,1}$ increases at time step $t = 122$, and then $P_{H,1}$ drops at time step $t = 126$ with the trust belief increases and drops, correspondingly. At $t = 366$, Robot 1 meets

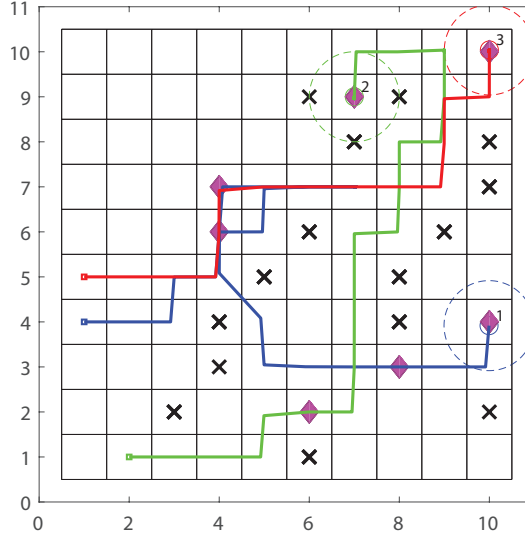


Fig. 17. Final paths for 3-robot symbolic motion planning switching between manual and autonomous motion planning mode.

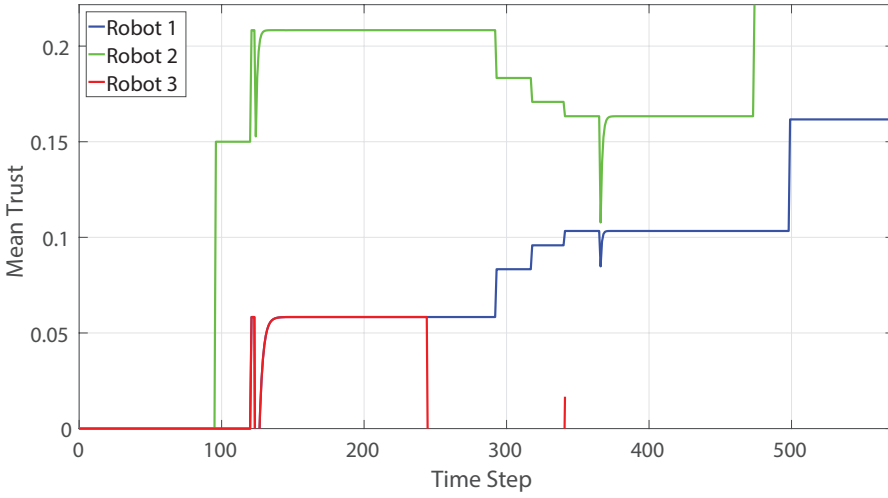


Fig. 18. Evolution of mean trust \bar{T}_i for all 3 robots.

obstacles, leads to trust increase, and requests for human intervention. The human subject plans a path passing through obstacles and reaches its left goals (see Fig. 17). At $t = 500$, $P_{R,1}$ and trust increases accordingly. Robot 1 finally stops at $t = 572$ after reaching all the goals including the newly assigned goal (10,4).

Robot 2 is in autonomous mode throughout the simulation. As shown in Fig. 20, at time steps 125 and 367, due to the human interaction with the other two robots, $P_{H,2}$ drops and trust drops

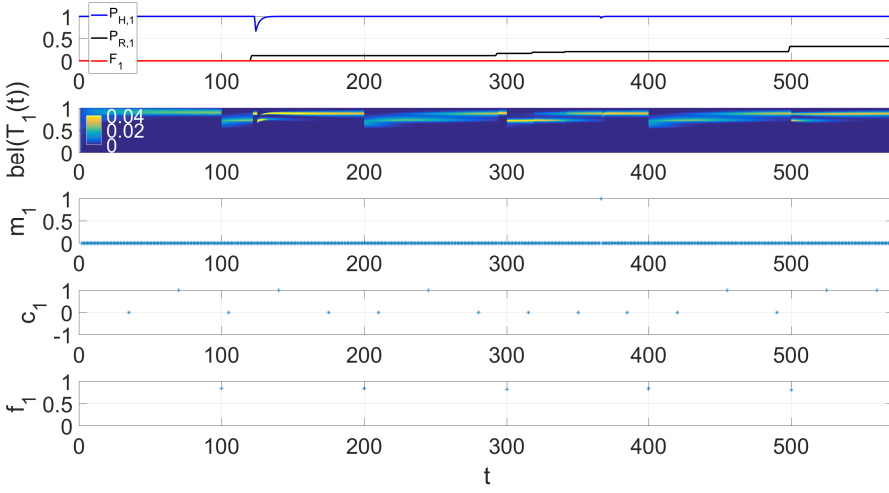


Fig. 19. Robot 1: (a) Evolution of human performance $P_{H,1}$, robot performance $P_{R,1}$, and fault F_1 , (b) trust belief $bel(T_1(t))$, (c) human intervention m_1 , (d) trust change c_1 , and (e) trust feedback f_1 .

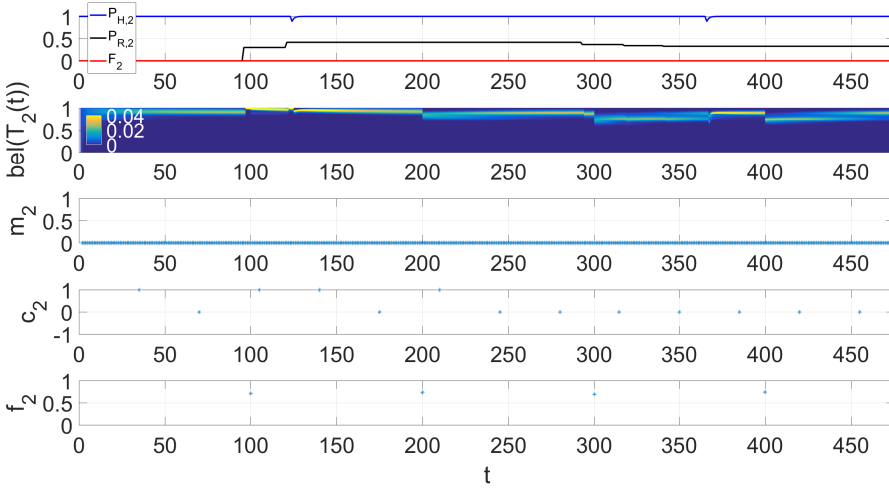


Fig. 20. Robot 2: (a) Evolution of human performance $P_{H,2}$, robot performance $P_{R,2}$, and fault F_2 , (b) trust belief $bel(T_2(t))$, (c) human intervention m_2 , (d) trust change c_2 , and (e) trust feedback f_2 .

correspondingly. At time steps 97 and 122, Robot 2 senses obstacles. $P_{R,2}$ increases as a result and trust increases accordingly. At $t = 319$, $P_{R,2}$ drops and trust drops correspondingly. The drop is because Robot 2 meets an obstacle on its planned path. Robot 2 stops at $t = 475$ after reaching all of its assigned goals.

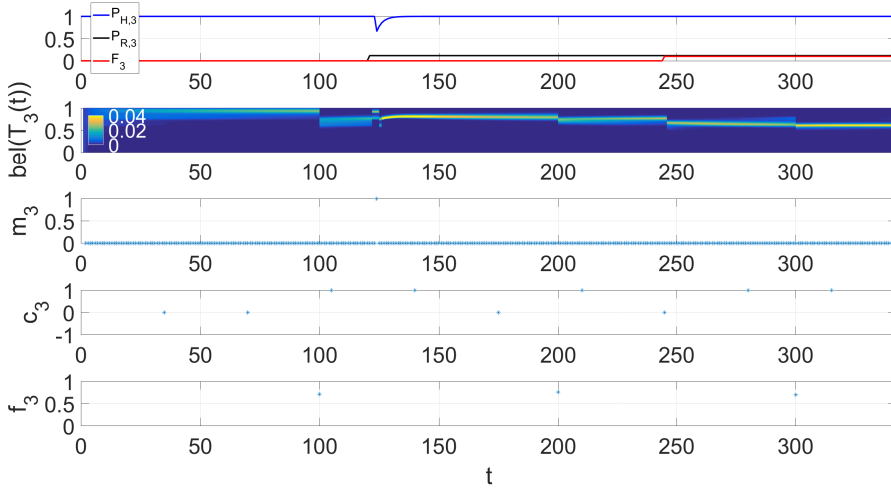


Fig. 21. Robot 3: (a) Evolution of human performance $P_{H,3}$, robot performance $P_{R,3}$, and fault F_3 , (b) trust belief $bel(T_3(t))$, (c) human intervention m_3 , (d) trust change c_3 , and (e) trust feedback f_3 .

8 CONCLUSIONS

In this paper, we have explored methods to address scalability, safety, performance, and adaptability of symbolic motion planning for multi-robot systems that interact with a human operator, with interactions affected by human trust. A quantitative, dynamic, and probabilistic model is developed to compute human trust in a robot during a collaborative ISR task. Scalability is addressed by decomposing portions of the global specification related to goal reachability and obstacle avoidance, while portions that require inter-robot collision avoidance are addressed through a protocol that relies on local communication and control to modify plans as needed during execution. Trust affects this decomposition adaptively, with more trusted robots assigned more destinations, and with trust decreasing as robots generate faults by coming too close to obstacles. In addition to implementing methods for obstacle and collision avoidance, safety versus efficiency of planning is addressed by switching between a safe but conservative robot motion planning mode that avoids obstacles based on an overapproximation of the environment and a riskier but more efficient human planning mode that allows paths between obstacles, with switching mediated by human trust. Finally, deadlock- and livelock-free algorithms are proposed to guarantee the reachability of all goal destinations with a human-in-the-loop.

In the future, there are several other areas that could be explored. The computational trust model we use here assumes trust in a specific robot evolves independently of all other robots, when in actuality, evolution of trust for each robot might be interdependent [38]. Other human-to-robot trust models could be used in the same framework. Moreover, we have assumed that obstacles are placed such that all destinations are reachable by all robots, which would not be true in all environments and would require more careful assignment of destinations. In some cases, certain destinations might not be reachable by any robot, which would require revising the specification. The simulations in Section 7 present a simple environment to demonstrate the overall strategy. However, in real-world applications (e.g. automated storage and retrieval robotic systems) the number of robots, goals and obstacles can be much larger. Correspondingly, human's cognitive workload will increase as well

and overload will negatively affect the user performance. To further quantify the workload and the corresponding effects on user performance, a proper workload model will be adopted in our future work.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Anqi Xu of McGill University for the discussion of the OPTIMO trust model.

REFERENCES

- [1] J. A. Adams. 2002. *Critical Consideration for Human-Robot Interface Development*. Technical Report. AAAI.
- [2] D. Aksaray, K. Leahy, and C. Belta. 2015. Distributed multi-agent persistent surveillance under temporal logic constraints. *IFAC-PapersOnLine* 48, 22 (2015), 174–179.
- [3] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. 2000. Discrete abstractions of hybrid systems. *Proc. IEEE* 88, 7 (2000), 971–984.
- [4] R. Alur, S. Moarref, and U. Topcu. 2016. Compositional Synthesis of Reactive Controllers for Multi-agent Systems. In *International Conference on Computer Aided Verification*. Springer, 251–269.
- [5] C. Baier and JP Katoen. 2008. *Principles of model checking*. MIT press Cambridge.
- [6] W. A. Bainbridge, J. Hart, E. S. Kim, and B. Scassellati. 2008. Fan-out: Measuring Human Control of Multiple Robots. *Proceedings of 17th IEEE International Symposium on Robot and Human Interactive Communication RO-MAN* (August 2008), 701–706.
- [7] S. Bak, D. K. Chivukula, O. Adegunle, M. Sun, M. Caccamo, and L. Sha. 2009. The system-level simplex architecture for improved real-time embedded system safety. In *Real-Time and Embedded Technology and Applications Symposium, 2009. RTAS 2009. 15th IEEE*. IEEE, 99–107.
- [8] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas. 2007. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14, 1 (2007), 61–70.
- [9] C. Belta, V. Isler, and G. J. Pappas. 2005. Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics* 21, 5 (2005), 864–874.
- [10] M. Bolton, E. Bass, and R. Siminiceanu. 2013. Using Formal Verification to Evaluate Human-Automation Interaction: A Review. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43, 3 (2013).
- [11] E. M. Clarke, O. Grumberg, and D. Peled. 2000. *Model Checking*. MIT (2000).
- [12] M. S. Cohen, R. Parasuraman, and J. T. Freeman. 1998. Trust in decision aids: A model and its training implications. In *Proceedings of Command and Control Research and Technology Symposium*.
- [13] J. W. Crandall, M. A. Goodrich, D. R. Olsen Jr, and C. W. Nielsen. 2005. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 35, 4 (2005), 438–449.
- [14] J. W. Crandall, C. W. Nielsen, and M. Goodrich. 2003. Towards predicting robot team performance. In *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1. IEEE, 906–911.
- [15] P. de Vries, C. Midden, and D. Bouwhuis. 2003. The effects of errors on system trust, self-confidence, and the allocation of control in route planning. *International Journal of Human-Computer Studies* 58, 6 (2003), 719–735.
- [16] Munjal Desai. 2012. *Modeling trust to improve human-robot interaction*. Ph.D. Dissertation. University of Massachusetts Lowell.
- [17] B. S. Dhillon and A. R. M. Fashandi. 1997. Safety and reliability assessment techniques in robotics. *Robotica* 15, 06 (1997), 701–708.
- [18] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas. 2005. Temporal logic motion planning for mobile robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020–2025.
- [19] S. Farrell and S. Lewandowsky. 2000. A connectionist model of complacency and adaptive recovery under automation. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 26, 2 (2000), 395.
- [20] L. Feng, C. Wiltse, L. R. Humphrey, and U. Topcu. 2015. Controller Synthesis for Autonomous Systems Interacting with Human Operators. In *Proceedings of ACM/IEEE 6th International Conference on Cyber-Physical Systems*.
- [21] I. Filippidis, D. V. Dimarogonas, and K. J. Kyriakopoulos. 2012. Decentralized multi-agent control from local LTL specifications. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 6235–6240.
- [22] M W Floyd, M Drinkwater, and D W Aha. 2015. Improving trust-guided behavior adaptation using operator feedback. In *International Conference on Case-Based Reasoning*. Springer, 134–148.
- [23] A. Freedy, E. DeVisser, G. Weltman, and N. Coeyman. 2007. Measurement of trust in human-robot collaboration. In *International Symposium on Collaborative Technologies and Systems, 2007. CTS 2007*. IEEE, 106–114.

- [24] J. Fu and U. Topcu. 2015. Pareto efficiency in synthesizing shared autonomy policies with temporal logic constraints. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 361–368.
- [25] J. Gao and J. D. Lee. 2006. Extending the decision field theory to model operators' reliance on automation in supervisory control situations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 36, 5 (2006), 943–959.
- [26] M. A. Goodrich and A. C. Schultz. 2007. Human-Robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction* 1, 3 (2007), 203–275.
- [27] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. 2011. Safe physical human-robot interaction: measurements, analysis and new insights. In *Robotics Research*. Springer, 395–407.
- [28] R. J. Hall. 1996. Trusting your assistant. In *Proceedings of the 11th Knowledge-Based Software Engineering Conference*. IEEE, 42–51.
- [29] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. YC Chen, E. J. De Visser, and R. Parasuraman. 2011. A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 53, 5 (2011).
- [30] B. Hayes and B. Scassellati. 2013. Challenges in Shared-Environment Human-Robot Collaboration. *Learning* 8 (2013), 9.
- [31] K. A. Hoff and M. Bashir. 2015. Trust in Automation Integrating Empirical Evidence on Factors That Influence Trust. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 57, 3 (2015), 407–434.
- [32] L. R. Humphrey, E. M. Wolff, and U. Topcu. 2014. Formal Specification and Synthesis of Mission Plans for Unmanned Aerial Vehicles. In *2014 AAAI Spring Symp. Series*.
- [33] M. Huth and M. Ryan. 2012. *Logic in Computer Science: Modeling and Reasoning about Systems* (2nd edition ed.). Cambridge.
- [34] S. Ikemoto, H. B. Amor, T. Minato, B. Jung, and H. Ishiguro. 2012. Physical human-robot interaction: Mutual learning and adaptation. *IEEE Robotics & Automation Magazine* 19, 4 (2012), 24–35.
- [35] J. Jian, A. M. Bisantz, and C. G. Drury. 2000. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics* 4, 1 (2000), 53–71.
- [36] S. Karaman and E. Frazzoli. 2009. Sampling-based motion planning with deterministic μ -calculus specifications. In *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009*. IEEE, 2222–2229.
- [37] M. Karimadini and H. Lin. 2011. Guaranteed global performance through local coordinations. *Automatica* 47 (2011), 890–898.
- [38] D. Keller and S. Rice. 2009. System-Wide versus Component-Specific Trust Using Multiple Aids. *The Journal of General Psychology* 137, 1 (2009), 114–128.
- [39] M. Kim, M. Viswanathan, H. Ben-Abdallah, S. Kannan, I. Lee, and O. Sokolsky. 1998. *A framework for run-time correctness assurance of real-time systems*. Technical Report. Technical Report MS-CIS-98-37, University of Pennsylvania.
- [40] D. E. Kirk. 2012. *Optimal control theory: an introduction*. Courier Corporation.
- [41] M. Kloetzer and C. Belta. 2010. Automatic Deployment of Distributed Teams of Robot From Temporal Logic Motion Specifications. *IEEE Transactions on Robotics* 26, 1 (February 2010), 48–61.
- [42] M. Kloetzer, S. Itani, S. Birch, and C. Belta. 2010. On the Need for Communication in Distributed Implementations of LTL Motion Specifications. *IEEE International Conference on Robotics and Automation* (2010), 4451–4456.
- [43] Steven M. LaValle. 2006. *Algorithms Planning*. (2006).
- [44] J. D. Lee and N. Moray. 1992. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* 35, 10 (1992), 1243–1270.
- [45] J. D. Lee and K. A. See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 46, 1 (2004), 50–80.
- [46] D. Z. Levin, E. M. Whitener, and R. Cross. 2006. Perceived trustworthiness of knowledge sources: The moderating impact of relationship length. *Journal of Applied Psychology* 91, 5 (2006), 1163–1171.
- [47] W. Li, D. Sadigh, S. S. Sastry, and S. A. Seshia. 2014. Synthesis for human-in-the-loop control systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 470–484.
- [48] M. F. Mahani and Y. Wang. 2016. Run-time Verification for Symbolic Robot Motion Planning with human-in-the-loop. *The ASME Dynamic Systems and Control Conference (DSCC)* (2016).
- [49] D. G. Mikulski. 2013. *Trust-Based Cooperative Games and Control Strategies for Autonomous Military Convoys*. Ph.D. Dissertation. Okaland University.
- [50] T. K. Moon. 1996. The expectation-maximization algorithm. *Signal processing magazine, IEEE* 13, 6 (1996), 47–60.
- [51] N. Moray and T. Inagaki. 1999. Laboratory studies of trust between humans and machines in automated systems. *Transactions of the Institute of Measurement and Control* 21, 4-5 (1999), 203–211.

- [52] N. Moray, T. Inagaki, and M. Itoh. 2000. Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied* 6, 1 (2000), 44.
- [53] NuSMV. 2015. NuSMV: a new symbolic model checker. (2015). <http://nusmv.fbk.eu/>.
- [54] I. Papusha, J. Fu, U. Topcu, and R. M. Murray. 2016. Automata theory meets approximate dynamic programming: Optimal control with temporal logic constraints. In *IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 434–440.
- [55] V. Raman and H. Kress-Gazit. 2013. Towards minimal explanations of unsynthesizability for high-level robot behaviors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 757–762.
- [56] W. B. Rouse and S. H. Rouse. 1983. Analysis and Classification of Human Error. *IEEE Transactions on Systems, Man, and Cybernetics* 13, 4 (1983), 539–549.
- [57] H. A. Ruff, S. Narayanan, and M. H. Draper. 2002. Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. *Presence: Teleoperators and virtual environments* 11, 4 (2002), 335–351.
- [58] B. Sadrfaridpour, H. Saeidi, J. Burke, K. Madathil, and Y. Wang. 2015. Modeling and Control of Trust in Human-Robot Collaborative Manufacturing. In *The Intersection of Robust Intelligence (RI) and Trust in Autonomous Systems*. Springer.
- [59] B. Sadrfaridpour, H. Saeidi, and Y. Wang. 2016. An Integrated Framework for Human-Robot Collaborative Assembly in Hybrid Manufacturing Cells. *IEEE International Conference on Automation Science and Engineering (CASE)* (2016). Best Student Paper Winner.
- [60] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. Seshia. 2014. Automated composition of motion primitives for multi-robot systems from safe LTL specifications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1525–1532.
- [61] L. Sha. 2001. Using simplicity to control complexity. *IEEE Software* 4 (2001), 20–28.
- [62] D. A. Spencer and Y. Wang. 2015. SLQR suboptimal human-robot collaborative guidance and navigation for Autonomous Underwater Vehicles. In *American Control Conference (ACC), 2015*. IEEE, 2131–2136.
- [63] D. A. Spencer, Y. Wang, and L. Humphrey. 2016. Trust-Based Human-Robot Interaction for Multi-Robot Symbolic Motion Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
- [64] J. Tumova and D. V. Dimarogonas. 2016. Multi-agent planning under local LTL specifications and event-based synchronization. *Automatica* 70 (2016), 239–248.
- [65] A. R. Wagner. 2009. *The role of trust and relationships in human-robot social interaction*. Ph.D. Dissertation. Georgia Institute of Technology.
- [66] X. Wang, Z. Shi, F. Zhang, and Y. Wang. 2015. Dynamic Real-Time Scheduling for Human-Agent Collaboration Systems Based on Mutual Trust. *Cyber-Physical Systems (CPS)*, Taylor & Francis (2015).
- [67] T. Wongpiromsarn, U. Topcu, and R. M. Murray. 2012. Receding Horizon Temporal Logic Planning. *IEEE Transaction on Automatic Control* 57, 11 (2012), 2817–2830.
- [68] A. Xu and G. Dudek. 2015. OPTIMO: Online probabilistic trust inference model for asymmetric human-robot collaborations. In *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction*. ACM.

Received December 2016; revised May 2017; accepted May 2017