Automatica 98 (2018) 1–13

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

A distributed approach to robust control of multi-robot systems[☆]

Yuan Zhou[a], Hesuan Hu[a,b,*], Yang Liu[a], Shang-Wei Lin[a], Zuohua Ding[c]

[a] School of Computer Science and Engineering, College of Engineering, Nanyang Technological University, Singapore 639798, Singapore
[b] School of Electro-Mechanical Engineering, Xidian University, Xi'an, Shaanxi 710071, PR China
[c] School of Information Sciences, Zhejiang Sci-Tech University, Hangzhou, Zhejiang 310018, PR China

ARTICLE INFO

Article history:
Received 9 December 2016
Received in revised form 19 May 2018
Accepted 2 August 2018

Keywords:
Autonomous robots
Discrete-event systems
Motion control
Robust control
Distributed control

ABSTRACT

Motion planning of multi-robot systems has been extensively investigated. Many proposed approaches assume that all robots are reliable. However, robots with priori known levels of reliability may be used in applications to account for: (1) the cost in terms of unit price per robot type, and (2) the cost in terms of robot wear in long term deployment. In the former case, higher reliability comes at a higher price, while in the latter replacement may cost more than periodic repairs, e.g., buses, trams, and subways. In this study, we investigate robust control of multi-robot systems, such that the number of robots affected by the failed ones is minimized. It should mandate that the failure of a robot can only affect the motion of robots that collide directly with the failed one. We assume that the robots in a system are divided into reliable and unreliable ones, and each robot has a predetermined and closed path to execute persistent tasks. By modeling each robot's motion as a labeled transition system, we propose two distributed robust control algorithms: one for reliable robots and the other for unreliable ones. The algorithms guarantee that wherever an unreliable robot fails, only the robots whose state spaces contain the failed state are blocked. Theoretical analysis shows that the proposed algorithms are practically operative. Simulations with seven robots are carried out and the results show the effectiveness of our algorithms.

1. Introduction

A multi-robot system is a system where multiple mobile robots work together to finish some given tasks by moving around in a given environment. Multi-robot systems have shown several advantages, such as increased spatial coverage and temporal throughput (Kitts & Egerstedt, 2008) and dramatic capability to resolve task complexity and efficiency (Khamis, Hussein, & Elmogy, 2015), and have been applied in many fields (Dias, Zinck, Zlot, & Stentz, 2004; Khamis et al., 2015; Smith, Schwager, & Rus, 2012). Motion planning is one of the most important issues in multi-robot systems. Researchers have proposed many approaches to collision-free path planning, such as formal methods (Egerstedt & Hu, 2002; Guo & Dimarogonas, 2015; Guo, Johansson, & Dimarogonas, 2013; Kloetzer & Belta, 2007), mathematical programming (Blackmore, Ono, & Williams, 2011; Gan, Fitch, & Sukkarieh, 2012), potential fields (Dimarogonas, Loizou, Kyriakopoulos, & Zavlanos, 2006; Li, Tamura, Yamashita, & Asama, 2013), reciprocal velocity obstacles (Van den Berg, Lin, & Manocha, 2008), state lattices (Pivtoraiko, Knepper, & Kelly, 2009), and sampling-based methods (Arslan, Berntorp, & Tsiotras, 2017).

If robots can replan their paths freely, robustness against robot failures can be obtained easily via real-time planning methods (Guo & Dimarogonas, 2015; Guo et al., 2013; Zhou, Hu, Liu, Lin, & Ding, 2017b) by regarding failed robots as obstacles. However, in many scenarios, especially in transportation systems, warehouses, tourist areas, and public parks, due to infrastructure limitations, task requirements, etc., a robot may have to move along a predetermined circular path. For example, different autonomous vehicles may be required to move along different circular lines to monitor the traffic conditions persistently; robots in warehouses are required to continuously load and unload materials or products in the given circular lines; and cars in tourist areas run in circles to carry tourists. In these examples, robots are required to move along predetermined and closed paths to perform some persistent tasks. Sometimes, with the state-of-the-art path planning algorithms, the predetermined paths can be obtained to accommodate infrastructure limitation (Paden, Čáp, Yong, Yershov, & Frazzoli, 2016) and special task requirements (Kress-Gazit, Lahijanian, & Raman, 2018; Smith et al., 2011; Tumova & Dimarogonas, 2016). For such systems, robust control is significant but not easy to achieve if robot failures are considered.

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Dimos V. Dimarogonas under the direction of Editor Christos G. Cassandras.

[*] Corresponding author at: School of Computer Science and Engineering, College of Engineering, Nanyang Technological University, Singapore 639798, Singapore.

E-mail addresses: y.zhou@ntu.edu.sg (Y. Zhou), hshu@mail.xidian.edu.cn (H. Hu), yangliu@ntu.edu.sg (Y. Liu), shang-wei.lin@ntu.edu.sg (S.-W. Lin), zuohuading@zstu.edu.cn (Z. Ding).

https://doi.org/10.1016/j.automatica.2018.08.022

In Zhou, Hu, Liu, and Ding (2017a), we studied collision and deadlock avoidance in such multi-robot systems. In that study, we assumed that robots can always work well without failures. However, in practice, a system is usually configured with robots of different levels of reliability since: (1) Robots of higher reliability are more expensive. Sometimes, it is not cost-efficient to use robots of higher reliability. For example, for non-critical tasks like warehouse operations, it is more cost-efficient to repair the failed robots, rather than deploying higher-reliability robots; for dangerous environments like mining, we prefer cheap robots since we can replace the failed robots once they are broken and cannot be recovered. (2) For long-term robot deployment, the hardware wear of robots determines the robot reliability. Performance of robots can degrade gradually, and manufacturers oftentimes provide performance degradation information in the robots' technical manuals. In this paper, we label robots of higher reliability as reliable and those of lower reliability as unreliable. We assume that reliable robots can always work well, while the unreliable ones may fail unexpectedly. We study robust control of multi-robot systems containing reliable and unreliable robots. The control target is to guarantee that the failure of an unreliable robot blocks the minimum number of robots.

Here we describe the reliability of robots in a non-stochastic manner. A probabilistic analysis of robustness with respect to stochastic models of failures is left for future work. In this paper, we assume that a classifier is available that can *a priori* label robots as reliable and unreliable. Such a classifier might be provided by the robot manufacturer in terms of wear, and robot models. The focus of this paper is on robustness in multi-robot systems modeled as discrete-event systems (DESs).

This paper proposes two distributed algorithms for robust control: one is for reliable robots, while the other is for unreliable ones. Our approach relates to the control of DESs. First, the motion of each robot is modeled as a labeled transition system (LTS). Second, a distributed strategy is briefly described to avoid collisions and deadlocks. Third, once deadlock avoidance is ensured, two distributed algorithms are proposed for reliable and unreliable robots to ensure robustness.

The main contributions of this study are that we (1) investigate robust control, which aims to minimize the number of stopped robots because of robot failures in multi-robot systems, and (2) propose a distributed robust control approach, with which a robot only needs some local information to perform its motion via detecting its own path and communicating with its neighbors.

The paper is organized as follows. Section 2 gives a brief literature review. Section 3 gives the LTS model of a multi-robot system and the problem statement. The collision and deadlock avoidance strategy is reviewed in Section 4. In Section 5, detailed algorithms for robust control are described. Simulation results are given in Section 6. Finally, discussion and conclusion are provided in Sections 7 and 8.

2. Related work

Researchers have made great efforts on robust control in multi-robot systems (Blackmore et al., 2011; Dias et al., 2004; Dogar et al., 2015; Goldberg & Chen, 2001; Hofbaur, Köb, Steinbauer, & Wotawa, 2007; Li, Li, & Kang, 2010; Liemhetcharat & Veloso, 2013; Parker, 1998; Preisler & Renz, 2012; Ulusoy, Smith, Ding, Belta, & Rus, 2013; Wu & Zhang, 2012). Most of the current work focuses on the improvement in the capability to adapt to failures, changes, and disturbance. These approaches can be roughly divided into three categories.

The first one is to obtain robustness by giving a system some degrees of redundancy, so that the tasks could still be completed by others even when some robots failed (Dias et al., 2004; Goldberg & Chen, 2001; Parker, 1998; Wu & Zhang, 2012). For example, Dias et al. (2004) studied the means to ensure robustness in a robot team facing with malfunctions. A set of redundant strategies are proposed, such that the tasks bestowed to the failed robots can still be finished by other correctly running robots. Thus, the team can still complete the given tasks in case of robot failures. In Goldberg and Chen (2001), collaborative control, i.e., multiple sources sharing the control of a single robot, is used to guarantee motion robustness against malfunctions of some resources. The main challenge in such methods is a proper selection of spare components or robots since a full backup is time- and cost-consuming.

The second one is to add detection mechanisms to systems to detect failures so as to recover/reconfigure the robots (Dogar et al., 2015; Hofbaur et al., 2007; Preisler & Renz, 2012; Ulusoy et al., 2013). For example, Dogar et al. (2015) proposed a hierarchical planning approach to accomplish some multi-scale assembly operations. In this method, robustness was achieved by the process of failure detection and recovery: Once a scanner loses the track of a target object, the system reverts to an earlier stage in order to re-localize by using wider field-of-view sensors. Hofbaur et al. (2007) proposed a generalized framework to improve robustness of the motion of mobile robots. The proposed framework can automatically monitor the driving device of a mobile robot and reconfigure the robot in cases of failures. Thus, high-level control, such as a path planner, is only to change its behavior in case of a serious damage.

The last one is based on probabilistic methods. To obtain robustness against uncertainties of robots and the environment, a system is designed to be endowed with flexibility in terms of probability (Blackmore et al., 2011; Chiu, Lian, & Wu, 2004; Liemhetcharat & Veloso, 2013). For example, Blackmore et al. (2011) used a probabilistic approach to planning vehicles' flexible trajectories. Each trajectory is described by the probabilistic distribution of a vehicle's states. The probabilities of collisions along these trajectories are designed to be below a given threshold. Thus, each vehicle can deal with uncertainties, such as indefinite localizations, erroneous modelings, and unexpected disturbances. Hence, the whole system can perform robustly. Liemhetcharat and Veloso (2013) studied the method to select a team of robots, each of which has a failure probability, to construct a robust system. The robustness they considered is the probability of the performance exceeding a threshold. The algorithms they proposed are to maximize the robustness of the system.

All above studies have focused on resilience to failure. However, sometimes, some failures are difficult to predict in time and some serious failures cannot be fixed in a short period. Thus, the resilience of the system cannot be achieved. Therefore, we consider robustness as a phenomenon in which robots' failures have the least detrimental effect on the system. When a robot fails at a position unpredictably, the robots that may collide with it are blocked inevitably. Other robots may also be blocked by the blocked ones, which in turn can block more robots. Thus, the whole system may stagnate. A well-designed system is expected to avoid such situations. In this study, we focus on robust control, expecting to minimize the number of robots that are blocked due to the failed robots.

3. System modeling and problem statement

In this section, we model robot motion using LTS models, based on which we present our problem statement. Let N and M be numbers of all robots and unreliable robots, respectively. Their indexes are denoted as $\mathbb{I}_N = \{1, 2, \ldots, N\}$ and $\mathbb{U}_N = \{u_1, \ldots, u_M\} \subseteq \mathbb{I}_N$, respectively. A robot is denoted as r_i, $i \in \mathbb{I}_N$.