Multi-Robot Coordination Workflows for Assistive Healthcare

Addison Clark[1], Ishfaq Ahmad[1], Divya Prakash[1], and Manfred Huber[1]

[1] University of Texas at Arlington, Arlington TX 76017, USA
addison.clark@mavs.uta.edu, iahmad@cse.uta.edu,
kishori.divya@gmail.com, huber@cse.uta.edu

Abstract. Assistive robots are robots that are designed to help people with disabilities in their daily lives. These robots often are either too simple, and thus cannot meet the needs of the user, or are too complex, and thus become overly expensive or unreliable. To mitigate this problem, we propose a system of multi-robot coordination that allows several heterogeneous robots to work cooperatively to assist an individual. We design care task workflows that cater to the needs of a specific patient, outlining the tasks needed for the user as well as the robots that can complete each task. We then build a ROS simulation to test our coordination methods on two example workflows for PTSD care routines. Our experiments show that our coordination successfully navigates each robot to its task location at the proper time, thus showing the feasibility of these methods for multi-assistive-robot coordination.

Keywords: Assistive Technologies, Robots, Multi-Robot Coordination

1 Introduction

Assistive robotics is the intersection between robotics and assistive technologies. There are many varieties of assistive robots, including those providing physical, therapeutic, or social services [1]. According to the World Health Organization, as of 2023 an estimated 16% of the world population has some form of disability [2]. These individuals can often benefit from assistive devices, specialized medical care, or caretakers. Assistive robotics must be designed based on the patient or user needs and must be capable of working alongside people.

One major challenge of assistive robotics is designing or programming a robot in a way that is truly useful for people with disabilities (PwD). This can be a very difficult task due to the varying requirements of each specific person. Individuals with the same diagnosis can experience the same condition in vastly different ways, and thus need assistance with different tasks. When designing assistive robots, it can be very easy to overlook certain aspects of a disability and end up with a robot that is only useful to a subset of the target audience. Conversely, trying to cover all possible needs in a single robot can result in a robot that is large, complicated, and very expensive. In both cases, a significant portion of people will be unable to benefit from the robot.

One possible solution to this problem in assistive robotics is to design simple robots that are suited for a more specific set of tasks, and then create a routine that will coordinate several of these robots. In this way, we can combine only the robots that are needed to meet the needs of one specific individual. Thus, the healthcare needs of each person can be catered to on an individual basis.

This paper proposes a multirobot coordination scheme that is customizable and scalable to user needs. The proposed method involves the design of workflows that specify the tasks that need to be completed and the various entities that can complete them. The tasks and robots are then simulated in an at-home environment. To demonstrate the personalization and scalability, we design two example workflows for a fictional patient with PTSD. The framework presented here is adaptable to more robots, and each task is abstracted such that other tasks can easily be added or swapped in.

2 Background

2.1 Assistive Robotics

Assistive robots are defined as robots designed to help PwD, both with healthcare tasks and in their daily lives. These robots are further categorized into physically assistive robots and socially assistive robots [1]. Some examples of physically assistive robots are exoskeletons and smart wheelchairs [3], [4]. Other examples include any robot that the user touches, or that picks up, delivers, or moves objects for the user.

Socially assistive robots, on the other hand, do not physically interact with the user or objects. Instead, they can facilitate social interaction between people or provide social interaction to a user. Some examples are [5] and [6], which both use HRI and human-robot interfacing to provide social services and facilitate social interaction.

2.2 Methods of Robot Coordination

Multi-robot coordination methods are highly application dependent, as the coordination itself changes based on the types of robots and the tasks they are completing. These coordination schemes can be categorized as homogeneous and heterogeneous, based on the composition of the robots used. In homogeneous coordination, every robot is the same, whereas heterogeneous coordination involves robots with different capabilities.

Two examples of homogeneous coordination are presented in [11] and [12]. In [11], an impedance-based control scheme is used to allow feedback between robots as they carry an object together. In [12], a leader-follower algorithm is used to guide mobile robots to their goal. This exemplifies the robot- and task-dependent nature of coordination algorithms.

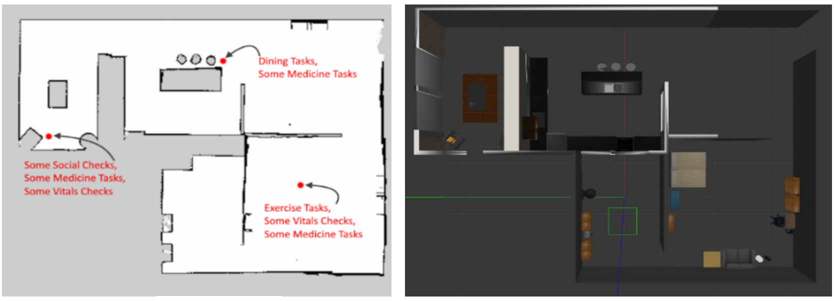
Heterogeneous multi-robot coordination, conversely, is used for scenarios in which different kinds of tasks are required, and therefore robots with different capabilities are needed. In [13], three robot modules work cooperatively to sew personalized stent grafts. [14] uses an unmanned ground vehicle (UGV) and three unmanned arial vehicles (UAV) to provide information on fields and move to specified locations. Once the task is determined, requirements for robot capabilities, types of robots, movement, and information sharing can be used to define the control scheme and coordination method.

2.3 AI in Robotics

CARE is an assistive multi-robot system that uses a system of heterogeneous assistive robots to assist elderly patients within a nursing home [7]. The system uses a centralized AI controller to learn user preferences with regards to different tasks. It also relies on many sensors for data gathering, which means that it requires a specific environment and setup. AI is also necessary for many robotic interfacing methods, such as EEG and computer vision [15]. The processing of complex signals such as video or brain waves is a complicated task that requires sophisticated machine learning algorithms, and there is much ongoing work in improving the reliability and accuracy of those systems. The notable differences in the system proposed in this paper compared to past work are that it is designed for a generic home environment. Integration of ambient sensors or other streams of data is possible, but they are not assumed to be present in all scenarios. Our system also generalizes the robots used, so that different patients can receive exactly the care that is necessary. Additionally, we assume that our system is set up in part with input from the patient's doctor or care provider, so preferences and care needs do not need to be learned.

3 Methodology

A central controller is used to gather all relevant data from each robot and use that to track the environment state. Factors such as robot location will inform the decisions with regards to task assignment.



(a) (b)

Fig. 1. (a) A map of the apartment environment with the locations of tasks marked, and (b) a top-down view of the apartment model.

Each robot will send their own sensor data to the central controller, which will help determine the environment state, which in turn informs the assignment of tasks. The internal data includes the overall workflow data. The workflow is the description of all tasks that must be performed, which robots can perform those tasks, and the timing and dependencies of certain tasks. The workflows will be outlined in Section 4.

The simulation environment is set up using ROS (robot operating system) melodic on Ubuntu 18.04 and Gazebo for dynamic simulations. The older version of ROS was chosen due to its compatibility with more robot simulation packages.

The goal of this simulation is to demonstrate the functionality of a multi-robot coordination system for scalable, at home healthcare. For this purpose, we have designed a model apartment using Gazebo assets that will give the robots a physical environment in which to move. A map and picture of the environment are shown in Fig. 1. The robots themselves are three TurtleBot3s from Robotis [16].

4 Experimental Setup

4.1 Experiments

We have created two task graphs for patients with PTSD to demonstrate the efficacy of our system. Each workflow describes different assistive tasks in a household environment. The goal of the simulation is to test coordination and navigation methods in order to improve the service of the robots in each scenario. If a robot fails, it is assumed there will be humans involved that can pick up the slack, so the simulation will note the failure time and then move on to the next task, assuming that the previous task has been completed by a human.

Each workflow has the same set of inputs. First, is the task graph that shows each task, the robots capable of each task, the control/ data signals, and the dependencies between tasks. Next, each task has a service time input, which is used to simulate the time that a robot will have to remain at the task location to complete the task. The next input for each task is the set of robots capable of completing the task, which are summarized in Table 1. Each robot used in the simulation is a TurtleBot3, or TB3. The next input for each task is the navigation time limit tolerance constant c. This parameter determines the deadline for when a robot must arrive at the task. Robot navigation is sometimes unreliable, and the navigation time limit input parameter allows the system to abandon tasks when a robot becomes stuck or is oscillating, regardless of the reason or navigation method. The final input for each task is the robot idle time; this is the time that the robots remain idle after completing each task. It can be used to set the timing between tasks, such as mealtimes, exercises, and when medicine must be taken.

Table 1. Robot to Task Mapping

		Tasks	
		PTSD 1 Workflow	PTSD 2 Workflow
Robot Label	TB3 (0)	Food Delivery	Food Delivery
	TB3 (1)	Medicine Delivery	Medicine Delivery
	TB3 (2)	—	Social Check-ups

The outputs of each task graph are various timing metrics. The first is the coordination time, which is the time between receiving the command to perform a task and sending that command on to a robot. The next output is the navigation time limit; this is the time calculated with *Equation 1*, and determines the maximum time allowed for navigating to a task. The navigation time is the next output, which shows the time the robot spent moving to the goal location. The next output is the total completion time. This is the total time from sending the signal that a task must be completed until it has either been completed or abandoned. Next, we mark if the task has been completed, and at the end, if the entire workflow was completed successfully. With these outputs, we can measure the success of our system. The goal of the system is to maximize the completion rate and minimize the navigation and coordination time.

4.2 User-Centered Workflow Design

The first task graph is shown in Fig. 2. This is a simple care routine for a patient with PTSD, depression, or other mental disorders. Two robots are used to deliver meals and medicine respectively. In this care routine, the patient's medicine must be taken after breakfast. At the end of all care scenarios, we make a report on the task completions and user data for a care provider to ensure that the system is properly aiding the patient. The legend at the bottom of the figure defines the different entities and symbols in the diagram, and the robot labels correspond to the actual robots used in the simulation. Each robot in the simulation is the same, but they represent different functionalities needed for each task graph. For Fig. 2-3, the control signals are generated by the central controller, which is not shown for brevity.

The inputs for the first experiment are shown in Table 2. The service time values are stand-ins for actual tasks being completed and are chosen to be fairly realistic based on the nature of the task.

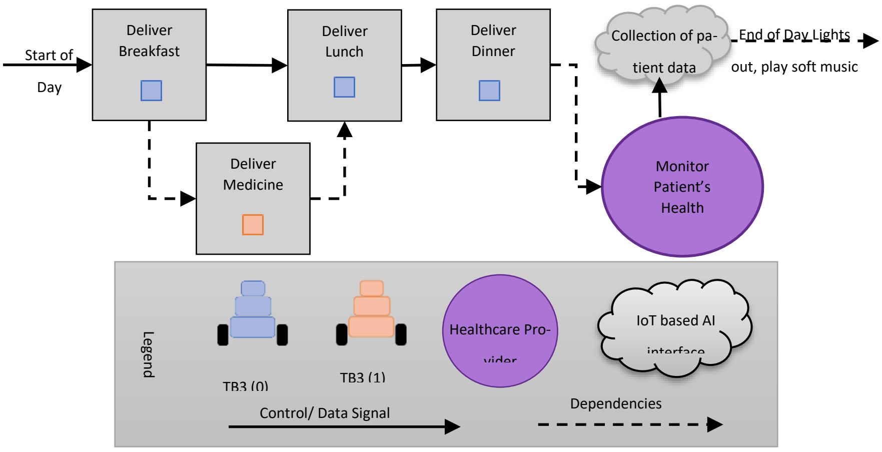


Fig. 2. The task graph for the PTSD patient in scenario 1.

The navigation time limits are set based on the distance that the robot has to travel to get from its starting location to the task location based on the following formula:

$$T(i) = (path\ length/maximum\ translational\ velocity) \times c \quad (1)$$

where c is the tolerance constant that allows extra time for correcting navigation mistakes and leniency for navigating obstacles. The *path length* is the distance determined dynamically once a robot receives a navigation goal and the initial path is set. The *maximum translational velocity* is how fast the robot can move in a straight line, which is provided by the robot manufacturer.

The second task graph is shown in Fig. 3. This is a scaled-up version of the first task graph. The same delivery tasks are included, but we now have a Socially Assistive

Robot (SAR) that completes wellness checks on the patient. The inputs for experiments using this scenario are found in Table 3.

Table 2. PTSD Task Graph 1 Experimental Inputs

	Inputs			
	Service Time (s) $t_s(i)$	Robot Set R = {r_1, r_2, ...r_m}	Navigation Tolerance Constant *c*	Robot Idle Time (hr) $t_d(i)$
Task 1 – Deliver Breakfast	45	TB3 (0)	2	½
Task 2 – Deliver Medicine	30	TB3 (1)	2	3 ½
Task 3 – Deliver Lunch	45	TB3 (0)	2	6
Task 4 – Deliver Dinner	60	TB3 (0)	2	0

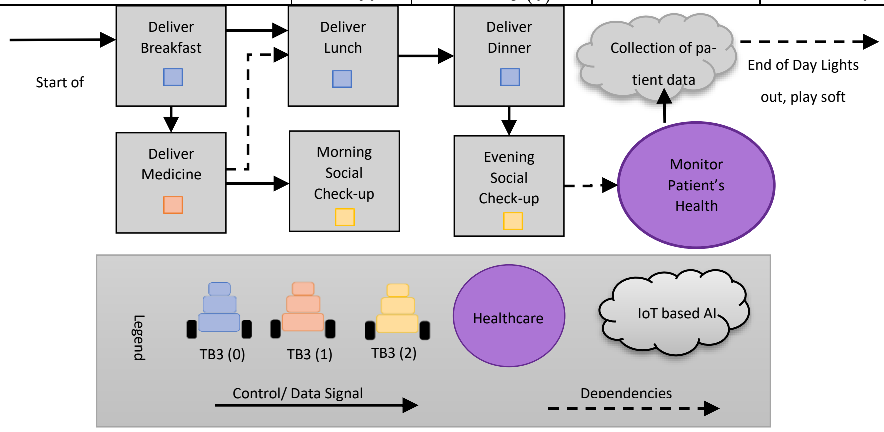


Fig. 3. The scaled-up task graph for the PTSD patient in scenario 2.

Table 3. PTSD Task Graph 2 Experimental Inputs

	Inputs			
	Service Time (s) $t_s(i)$	Robot Set R = {r_1, r_2, ...r_m}	Navigation Tolerance Constant *c*	Robot Idle Time (hr) $t_d(i)$
Task 1 – Deliver Breakfast	45	TB3 (0)	2	½
Task 2 – Deliver Medicine	30	TB3 (1)	2	½
Task 3 – Morning Social Check-up	120	TB3 (2)	2	3
Task 4 – Deliver Lunch	45	TB3 (0)	2	6
Task 5 – Deliver Dinner	60	TB3 (0)	2	2
Task 6 – Evening Social Check-up	180	TB3 (2)	2	0

5 Results and Discussion

The results of the experiments are presented below. The total completion time includes the coordination time, navigation time, task service time (if the task is successful), and any overheads such as communication time.

The first set of experiments follow the PTSD Task Graph 1 inputs. As shown in Table 4, all tasks in are completed successfully within the determined time limits. The navigation time limits are quite consistent across similar tasks. The navigation times are less consistent, showing that the robot navigation is sometimes suboptimal and loses time to obstacle avoidance and navigation recovery behaviors. .

Table 4. PTSD Task Graph 1 Experimental Outputs

	Coordination Time (ms) $t_c(i)$	Navigation Time Limit (s) $T(i)$	Navigation Time (s) $t_n(i)$	Total Execution Time (s) $t_e(i)$	Task Completed	Task Graph Completed
Task 1	5	121.96	65.20	110.41	Yes	Yes
Task 2	4	134.31	76.39	107.24	Yes	
Task 3	14	121.51	63.49	110.23	Yes	
Task 4	5	121.47	65.10	125.40	Yes	

The second set of experiments, implementing the PTSD Task Graph 2 inputs, are shown in Table 5. The robots are still able to successfully complete all tasks, despite the interference. However, this navigational failure is still worth noting, as it could cause task failures in the future.

Table 5. PTSD 2 Task Graph 2 Experimental Outputs

	Coordination Time (ms) $t_c(i)$	Navigation Time Limit (s) $T(i)$	Navigation Time (s) $t_n(i)$	Total Execution Time (s) $t_e(i)$	Task Completed	Task Graph Completed
Task 1	8	121.47	63.60	108.95	Yes	Yes
Task 2	8	133.99	71.60	102.92	Yes	
Task 3	4	247.47	130.30	250.60	Yes	
Task 4	6	121.93	67.50	112.72	Yes	
Task 5	5	121.47	65.30	125.67	Yes	
Task 6	6	247.81	132.49	312.69	Yes	

The robot coordination for these experiments is fairly trivial, as the current task graphs call for a specific robot for each task. However, future experiments will include more complex workflows and more robot capability overlap, and thus non-trivial coordination methods.

6 Conclusions

Assistive robotics is the subfield of robotics that aims to help people with disabilities by designing and developing assistive robots. Due to the varying needs of individuals, even those with the same diagnosis, it is imperative that these robots be designed to be customizable and scalable. For this purpose, we have designed a framework for assistive robot coordination in a home environment.

Acknowledgments. This project was funded by the National Science Foundation under Award Number:1757641.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. D. Feil-Seifer and M. J. Matarić, "Defining Socially Assistive Robotics," *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics*, 2005, pp. 465-468.
2. "Disability," World Health Organization. Accessed: Jan. 22, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/disability-and-health
3. S. R. Soekadar, M. Witkowski, C. Gómez, E. Opisso, J. Medina, M. Cortese, M. Cempini, M. C. Carrozza, L. G. Cohen, N. Birbaumer, and N. Vitiello, "Hybrid EEG/EOG-Based Brain/Neural Hand Exoskeleton Restores Fully Independent Daily Living Activities after Quadriplegia," *Science Robotics*, vol. 1, no. 1, Dec. 2016.
4. S. D. T. Olesen, R. Das, M. D. Olsson, M. A. Khan, and S. Puthusserypady, "Hybrid EEG-EOG-Based BCI System for Vehicle Control," *9th IEEE International Winter Conference on Brain-Computer Interface, BCI 2021*, 2021.
5. L. Chen, M. Wu, M. Zhou, J. She, F. Dong, and K. Hirota, "Information-Driven Multirobot Behavior Adaptation to Emotional Intention in Human-Robot Interaction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 647–658, 2018.
6. G. Zhang and J. P. Hansen, "Accessible Control of Telepresence Robots based on Eye Tracking," in *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications (ETRA)*, Denver, 2019, pp. 1–3.
7. A. A. Ravankar, S. A. Tafrishi, J. V. Salazar Luces, F. Seto, and Y. Hirata, "CARE: Cooperation of AI Robot Enablers to Create a Vibrant Society," *IEEE Robotics and Automation Magazine*, vol. 30, no. 1, pp. 8–23, Mar. 2023.
8. M. Luperto *et al.*, "Integrating Social Assistive Robots, IoT, Virtual Communities and Smart Objects to Assist at-Home Independently Living Elders: The MoveCare Project," *International Journal of Social Robotics*, 2022.
9. J. Vogel *et al.*, "An Ecosystem for Heterogeneous Robotic Assistants in Caregiving: Core Functionalities and Use Cases," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, pp. 12–28, Sep. 2021.
10. R. Doriya, S. Mishra, and S. Gupta, "A Brief Survey and Analysis of Multi-Robot Communication and Coordination," in *International Conference on Computing, Communication & Automation*, IEEE, May 2015, pp. 1014–1021.
11. D. Sieber and S. Hirche, "Human-Guided Multirobot Cooperative Manipulation," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1492–1509, 2019.
12. A. Soni and H. Hu, "A Multi-Robot Simulator for the Evaluation of Formation Control Algorithms," *2019 11th Computer Science and Electronic Engineering Conference, CEEC 2019 - Proceedings*, pp. 79–84, 2019.
13. B. Huang, M. Ye, Y. Hu, A. Vandini, S. L. Lee, and G. Z. Yang, "A Multirobot Cooperation Framework for Sewing Personalized Stent Grafts," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1776–1785, 2018.
14. C. Ju and H. Il Son, "Modeling and Control of Heterogeneous Agricultural Field Robots Based on Ramadge-Wonham Theory," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 48–55, 2020.
15. A. Clark and I. Ahmad, "Touchless and nonverbal human-robot interfaces: An Overview of the State-of-the-Art," *Smart Health*, vol. 27, p. 100365, 2023.
16. "TurtleBot3: Overview," Robotis. Accessed: Jan. 23, 2024. [Online]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/