12-2015

CO-DESIGN OF DYNAMIC REAL-TIME SCHEDULING AND COOPERATIVE CONTROL FOR HUMAN-AGENT COLLABORATION SYSTEMS BASED ON MUTUAL TRUST

Xiaotian Wang
Clemson University, xiaotiw@g.clemson.edu

CO-DESIGN OF DYNAMIC REAL-TIME SCHEDULING AND COOPERATIVE CONTROL FOR HUMAN-AGENT COLLABORATION SYSTEMS BASED ON MUTUAL TRUST

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Xiaotian Wang
December 2015

Accepted by:
Dr. Yue(Sophie)Wang, Committee Chair
Dr. Kalyan Katuri
Dr. Yongqiang Wang

Abstract

Mutual trust is a key factor in human-human collaboration. Inspired by this social interaction, we analyze human-agent mutual trust in the collaboration of one human and (semi)autonomous multi-agent systems. In the thesis, we derive time-series human-agent mutual trust models based on results from human factors engineering. To avoid both over-trust and under-trust, we set up dynamic timing models for the multi-agent scheduling problem and develop necessary and sufficient conditions to test the schedulability of the human multi-agent collaborative task.

Furthermore, we extend the collaboration between one human and multiple agents into the collaboration between multi-human network and swarm-based agents network. To measure the collaboration between these two networks, we propose a novel measurement, called fitness. By fitness, we can simplify multi-human and swarms collaboration into one-human and swarms collaboration. Cooperative control is incorporated into the swarm systems to enable several large-scale agent teams to simultaneously reach navigational goals and avoid collisions.

Our simulation results show that the proposed algorithm can be applied to human-agent collaboration systems and guarantee effective real-time scheduling of collaboration systems while ensuring a proper level of human-agent mutual trust.

To my family,

Thanks for your support and love.

Acknowledgments

First and foremost, I would like to thank my advisor, Dr. Yue (Sophie) Wang, for giving me the opportunities to join in Interdisciplinary Intelligence Research (I2R) Laboratory, learn systems controls, and finish related projects. Dr. Wang not only gave me much help and advices on research, but also guided me for my life with her knowledge and experience. I am grateful to my advisor for all of her support, and hope to keep in touch with her in the future. I would like to thank my advisory committee members Dr. Kalyan Katuri and Dr. Yongqiang Wang for their technical expertise and advice.

I also want to thank my lab mates in the I2R lab, namely, Hamed Saeidi, Behzad Sadrfaridpour, David Adam Spencer. They gave me strong support in conducting experiments and simulations, as well as their generous help. Thank you all for bringing fun and laugh to our laboratory life.

I would like to thank my parents Jing Zhang and Pingze Wang, for their constant love and support.

Last but not the least, I would like to thank my fiancee, Xinqi (Mandy) Zhang. She was always there no matter the best of times or the worst.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

As the labor cost increases and the autonomy technology advances, the number of human operators per agent has been reduced to a large extent. In future operations, it is envisioned that one human operator can work with multiple agents [7, 69]. In this pursuit, building mutual trust between human operators and (semi)autonomous agents is of particular importance since mutual trust is the basis of collaboration, which may improve task efficiency and reduce risks and errors. Similar to human-human trust, human-agent mutual trust includes both human-to-agent trust and agent-to-human trust. On one hand, if a human operator trusts agents in a task, he/she will delegate the task to these agents and hence reduce workload. On the other hand, if an agent trusts a human operator, that is, the agent believes in the commands from the human operator, it will finish the task based on these input commands.

Human-to-agent trust is a significant factor to guarantee successful human-agent collaboration (HAC) [20]. The recent meta-analysis [28] studies the factors involved in trust for human-robot interaction (HRI), which includes robot-related, human-related, and

environmental-related factors. In [21], Freedy et al. study the critical performance attributes of trust in HRI and develop a collaborative performance model. In [17], Yanco and Desai investigate the HRI problems involved in remote robot teleoperation (RRT) and summarize five categories of trust models. Apart from the above qualitative works, the dynamic aspect of trust under different operating conditions of an automated system is studied in [32]. Itoh and Tanaka propose a mathematical model of trust in automation based on the expectation of humans from the automation, dependability of the automation, and predictability of automation behaviour. Most of the existing literature has been focused on the unilateral human-to-agent trust. Nevertheless, since there exists interaction in the HAC system, trust between humans and agents should be bilateral, including both human-to-agent trust and agent-to-human trust. Here, the agent-to-human trust is similar to the human-to-agent trust, which inversely depends on the human performance. Based on different levels of agent-to-human trust, an agent will select different modes, such as "DECLINE" or "ACCEPT", to response [39, 46]. Therefore, inspired by the time-series trust model proposed in [39, 46], and the theoretical framework for trust in [28], we derive time-series dynamic models for the mutual trust between a human operator and a (semi)autonomous agent. The human-to-agent trust model is a function of agent performance and agent fault rates. The agent-to-human trust model is a function of human performance and human fault rates.

A human operator needs to distribute attention to each agent when collaborating with a multi-agent system. This case is analogous to the classic research topic in real-time scheduling. The collaboration with each agent can be understood as the execution of an individual task and the human operator can be viewed as a single processor when he/she needs to share attention with one agent at a time [69]. In this case, it is necessary to develop a dynamic scheduling algorithm so that human can allocate his/her attention for each agent in real time.

In the literature [43, 58], there are three classical scheduling algorithms: fixed pri-

ority scheduling, dynamic priority scheduling, and mixed priority scheduling, respectively. The paper [43] also discusses three corresponding boundary conditions for the schedulability test of these algorithms. Murray et al. in [48] develop a mathematical model for simultaneously routing multiple unmanned aerial vehicles (UAVs) and scheduling human operators, subject to operator workload considerations. However, they only consider the problem as constrained mathematical programming for task allocation and neglect the dynamic interactions between the human operators and the UAVs. Nevertheless, due to digital control intermediate between human and agents, we consider discrete-time scheduling in our paper. In [24], Gooding et al. develop a generalized discrete-time scheduling model based on the Embedded-Time Graph (ETG). Another general scheduling framework is developed by Pantelides in [51] based on the Resource-Task Network (RNT) process representation. In [71], Yee et al. present two methods to improve the efficiency of discrete-time scheduling algorithms. However, these continuous and discrete-time algorithms for processor scheduling are not applicable in our dynamic systems where the ultimate goal is not just to meet deadlines but also to avoid both "over-trust" and "under-trust". Therefore, in [59, 60, 74], we propose a dynamic timing model and necessary and sufficient conditions for schedulability test for dynamic systems. Specifically, for the human multi-agent collaboration system, we introduce a novel scheduling algorithm called "highest-trust-first" scheduling, which can guarantee effective real-time scheduling of manual and autonomous control of agents [69]. Furthermore, in the paper [67], based on the bilateral trust dynamic models and extending the "highest-trust-first" scheduling [69], we propose a rigorous schedulability test algorithm using the dynamic timing model to avoid both "over-trust" and "under-trust". In this part, we give a more detailed discussion of the dynamic timing model, proof of (the necessary and) sufficient conditions of schedulability, explanation of the schedulability test algorithm, as well as provide more detailed analysis of the simulation results. Note that the above part has been composed into two papers and accepted

by the American Control Conference (ACC) [67] and the Cyber Physical Systems (CPS) journal [66].

It is further envisioned that multiple agents are expanded into a swarm of agents in the human-agent collaboration systems. Swarm-agents is a new approach to coordinate large numbers of basic agents, which takes its inspiration from social insects [54]. A swarm system typically consists of a big number of relatively simple agents that act independently and in parallel on the various tasks that must be completed in order for the swarm to achieve its overall goal [5]. In the paper [34], Keller et al present a coupled reaction-diffusion partial derivative dynamic equations to model the motions of swarms. In the paper [8], Bonabeau et al provide a response threshold model to simulate the swarm-human dynamics, where uses the associated response threshold to combine the number of workers and the belonging task. Similarly, in [9], Bonabeau et al once again propose a new response threshold to simulate the succession of tasks. In the literature [40], Lerman et al uses a probability approach to describe the states of some group of robots based on a macroscopic quantity. In papers [3, 52, 53], Romanczuk et al propose a model of collective motion based on escape and pursuit responses. In the paper [73], Yu et al investigate swarming behaviors in multi-agent systems with nonlinear dynamics. In the paper [23], authors propose N-member "individual-based" continuous time swarm models by two fundamentally different approaches to analyze the swarm dynamics, which are spatial and non-spatial approaches. For the spatial approaches, the space (environment) is either explicitly or implicitly present in the model and the analysis. The environment framework is usually projected by the profile functions, which can represent the plane, cone, and concave-convex. In the individual-based swarm models, the space is related with the profiles of each (separate) individual while the profiles depict the clusters' shapes in the group-based systems [23, 73]. For the non-spatial approaches, the swarming dynamics are described in a non-spatial way in terms of frequency distributions of groups of various size. It is assumed that groups

of various population levels split or merge into other groups based on the inherent group dynamics, environmental conditions, and communications with other groups [26].

To improve the efficiency for a one-human and swarm-agents collaboration system, we extend unique human operator to multiple human. Hence, we need to focus on the multi-human system. In [42], Lieberman et al study evolutionary dynamics for homogeneous or spatially extended populations. They generalize population structure by arranging individuals on a graph. In [12], Champagnat et al unify basic evolutionary models to account for mutation bias and random drift between multiple evolutionary attractors and describe a population in which the adaptive traits of individuals influence their birth rate, the mutation process, their death rate, and how they interact with each other and their external environment. Replicator-Mutator dynamics are used to describe the dynamics of complex adaptive systems in population genetics, biochemistry and models of language learning [29, 37]. In the paper [31], Hussein uses the replicator-mutator dynamical equations to model the process of building individual behavioral inclinations. In [68], Wang et al extend the replicator-mutator dynamics into a distributed version for multi-agent networks with local interactions. In [64], Wang uses the replicator-mutator dynamics to model users and advertisers behaviors. In our case, we will use the replicator-mutator dynamical equations to model the individual human operator preference inclinations.

1.2 Contribution

The major contributions of this thesis are listed as follows: 1) We propose bilateral trust models, i.e., the human-to-agent trust model and agent-to-human trust model; 2) Based on these two trust models, we introduce a mutual trust-based scheduling method to determine the priority for the agent team; 3) Based on new trust models and the agent selection method, we provide a schedulability test to check the schedulability of the human

multi-agent system, combining with the dynamic timing model; 4) Furthermore, we extend the collaboration between one human and multiple agents into the collaboration between multi-human network and swarm-based agents network; 5) To measure the collaboration between these two networks, we propose a novel measurement, called 'fitness'.

1.3 Structure of Thesis

The organization of this thesis is as follows: In Chapter2, we first propose the t-wo unilateral trust models. In Chapter 3, we introduce dynamic real-time scheduling for human-agent collaboration systems based on mutual trust. Based on this result, we derive a necessary and sufficient schedulaiblity test algorithm. In Chapter 4, we extend the one human and multi-agent collaboration systems into the multi-human and swarm-agents collaboration systems. Besides, we co-design the scheduling and cooperative control for this network. Combining the replicator-mutator dynamics, we propose a novel measurement, called'fitness'. Discussion and future works are provided in Chapter 5.

Chapter 2

MUTUAL TRUST

2.1 Trust model

In this chapter, we start from the case where one human operator collaborates with one agent. Here, we introduce two dynamic trust models: human-to-agent trust model $T_{H\to A}$, and agent-to-human trust model $T_{A\to H}$, respectively.

First, it has been shown that human-to-agent trust is affected by three broad categories of influential variables, which are agent performance, human performance, and environmental factors [55]. It is further pointed out in the meta-analysis [28] that agent performance is strongly related to the trust level, the environmental factors are moderately associated with the trust level, and the human performance has the least relationship with the trust evolution. Therefore, combining the qualitative trust model [55] and time-series trust model proposed in [39,41,46], we have the following human-to-agent trust model:

$$T_{H\to A}(k) = A_1 T_{H\to A}(k-1) + B_1 P_A(k) - B_2 P_A(k-1) + D_1 F_A(k) - D_2 F_A(k-1), \tag{2.1}$$

where k denotes the discrete time step, $P_A(k)$ denotes the agent performance, $F_A(k)$ de-

notes the agent fault rate under the autonomous/manual control mode, and A_1, B_1, B_2, D_1, and D_2 are constant coefficients whose values depend on the human operator, the agent, and the collaborative task. As the above equation shows, the current trust level $T_{H\to A}(k)$ is determined by the prior trust level $T_{H\to A}(k-1)$, change of agent performance, and change of agent fault rate.

Next, we consider the agent-to-human trust model. Similar to human-to-agent trust, the agent-to-human trust model, $T_{A\to H}(k)$, will depend on the change of performance of the human collaborator, P_H, and the human fault rate, F_H. Analogous to Equation (2.1), we propose the following agent-to-human trust model:

$$T_{A\to H}(k) = A_2 T_{A\to H}(k-1) + C_1 P_H(k) - C_2 P_H(k-1) + E_1 F_H(k) - E_2 F_H(k-1), \tag{2.2}$$

where $A_2, C_i, E_i, i = 1, 2$ are constant coefficients. The human-agent mutual trust models apply to each agent in the multi-agent system with varying coefficients specific to agent capabilities.

2.2 Trust & Use of automation

To avoid both "over-trust" and "under-trust", we assume that both $T_{H\to A}(k)$ and $T_{A\to H}(k)$ must fall within the desired regions [1] denoted as, $T_{H\to A}(k) \in [T_{H\to A,l}, T_{H\to A,u}]$ and $T_{A\to H}(k) \in [T_{A\to H,l}, T_{A\to H,u}]$ for $k \geq 0$. If the human-to-agent trust $T_{H\to A}(k)$ exceeds its corresponding upper limit ($T_{H\to A}(k) \geq T_{H\to A,u}$), this indicates that the human operator has too much trust on the autonomous operation of the agents, i.e., "over-trust". To overcome this problem, the human operator should start to control the agent manually. On the other hand, if the human-to-agent trust $T_{H\to A}(k)$ goes below the lower limit ($T_{H\to A}(k) \leq T_{H\to A,l}$), it means that the human operator has too little trust on the au-

tonomous operation of the agents and puts too much manual control, i.e., "under-trust". In this case, the human operator should allow the autonomous operation of agent itself rather than control it manually.

Similarly, if the agent-to-human trust level $T_{A\to H}(k)$ exceeds its corresponding upper limit ($T_{A\to H}(k) \geq T_{A\to H,u}$), this indicates that the agent relies too much on the human operator's manual control, i.e., "over-trust". To avoid this situation, the agent should be controlled autonomously without human intervention. On the other hand, if the agent-to-human trust $T_{A\to H}(k)$ goes below the lower limit ($T_{A\to H}(k) \leq T_{A\to H,l}$), this indicates that the agent has too little trust on the human operator, i.e., "under-trust". In this case, the agent should be operated manually.

2.3 Agent performance model

We consider two modes when a human collaborates with an agent: the autonomous mode and the manual mode, with different performance models given by the following two different equations [15]

$$P_{n,A}(k) = \begin{cases} (1-k_{n,A})P_{n,A}(k-1) + k_{n,A}P_{n,A,\text{min}}, & \text{(autonomous mode)} \\ (1-k_{n,H})P_{n,A}(k-1) + k_{n,H}P_{n,A,\text{max}}, & \text{(manual mode)} \end{cases} \tag{2.3}$$

where $P_{n,A,\text{max}}, P_{n,A,\text{min}} \in [0,1]$ stand for the maximum and minimum performance of the agent A_n, and $k_{n,A}, k_{n,H} \in (0,1)$ are the performance coefficients for autonomous mode and manual mode, respectively. The agent performance model (2.3) guarantees that $P_{n,A}$ of each agent A_n is bounded between $[P_{n,A,\text{min}}, P_{n,A,\text{max}}]$, given that their initial performance falls within $[P_{n,A,\text{min}}, P_{n,A,\text{max}}]$. From Equation (2.3), the performance of an agent will decrease under the autonomous mode and increase under the manual mode.

2.4 Human performance model

The Yerkes-Dodson law [72] describes human performance as an empirical model with respect to human arousal and task difficulty. In our paper, human performance means the capability and efficiency of the human operator collaborating with an agent. We represent the following performance model, as [6]

$$P_H(k) = (P_{H,\max} - P_{H,\min}) \left(\frac{r(k)}{\beta}\right)^{\beta} \left(\frac{1-r(k)}{1-\beta}\right)^{1-\beta} + P_{H,\min}, \tag{2.4}$$

where $\beta \in (0,1)$ represents the difficulty of a task for a human (a smaller value of β represents a more difficult task [45]), $r(k)$ represents the utilization, and $P_{H,\max}$ and $P_{H,\min}$ represent the maximum and minimum human performance value, respectively. Note that the human performance model (2.4) guarantees that P_H is bounded between $[P_{H,\min}, P_{H,\max}]$.

Inspired by the single-server queue model [56, 69], we introduce the following utilization model for the case when one human operator collaborates with multiple agents

$$\begin{aligned} r(k) &= (1-\frac{1}{\tau})r(k-1) + \frac{u(k)}{\tau}, \\ u(k) &= \begin{cases} 1 & \text{manual mode} \\ 0 & \text{autonomous mode} \end{cases}, \end{aligned} \tag{2.5}$$

where $u(k)$ denotes the control mode of an agent and $\tau > 0$ is a time constant that determines the extent to which past utilization affects the current state. The time constant τ here represents the inverse of the sensitivity of the operator to its recent utilization history, which means a larger τ corresponds to lower sensitivity, and vice versa. As shown in Eq. (2.5), the utilization ratio $r(k)$ is determined by the control modes of all agents. $r(k)$ increases in the manual mode ($u(k) = 1$) and decreases in the autonomous mode ($u(k) = 0$). The utilization ratio $r(k)$ is bounded between 0 and 1.

Chapter 3

DYNAMIC REAL-TIME SCHEDULING FOR HUMAN-AGENT COLLABORATION SYSTEMS BASED ON MUTUAL TRUST

3.1 Multi-Agent Schedulability

Consider the case where a human operator collaborates with N (semi)autonomous agents and denote the agents as $\{A_1, ..., A_N\}$. Based on the above analysis, we define the schedulability of a human multi-agent collaborative team in terms of mutual trust as follows. We use the subscript n to represent each agent A_n in the following notation.

Definition 3.1.1. *Consider an arbitrary time period starting from k_a and ending at k_b, denoted as $k \in [k_a, k_b]$. For any agent A_n ($1 \leq n \leq N$) that is collaborating with a human operator, if both the human-to-agent and agent-to-human trust level fall within the*

limits of the desired trust region, i.e., $T_{n,H-A}(k) \in [T_{n,H-A,l}, T_{n,H-A,u}]$ *and* $T_{n,A-H}(k) \in [T_{n,A-H,l}, T_{n,A-H,u}]$*, the human multi-agent collaboration system is said to be* schedulable *within* $[k_a, k_b]$.

As we can see from the above definition, the most simple and straightforward way of schedulability test is to (1) compute mutual trust $T_{n,H-A}(k)$ and $T_{n,A-H}(k)$ at each time step $k \in [k_a, k_b]$; and (2) then check whether both trust values fall within the desired trust region at each time step. However, this method requires a lot of computation resources, which makes it impractical for online computing. In the following part of this paper, we will show that the schedulability test can be performed at only a small set of critical time points instead of every time step. Based on this observation, we will develop necessary and sufficient conditions to check the schedulability of such human multi-agent system.

3.2 Dynamic Timing Model

In this paper, we develop trust based scheduling algorithm to allocate human-attention to each agent so that the mutual trust level of each human-agent pair falls within the desired trust region. Existing schedulability tests may not be directly applicable to the problem of trust based scheduling. The fixed priority scheduling algorithms, such as the rate-monotonic scheduling (R.M.S.) [43], can only be used in systems where the scheduling parameters are constant over each period, which is not the case for our dynamic systems. On the other hand, the dynamic or mixed priority scheduling algorithms, such as the earliest deadline first (E.D.F.) scheduling [43], guarantee the tasks to finish before deadline while our goal is to maintain the trust level within the desired region in addition to meet deadlines which increases the complexity of the schedulability test.

We define two parameters, collaboration time $I_n(k)$ and period L_n for each agent. The choice of $I_n(k)$ will dynamically change according to the bilateral trust levels,

i.e., $T_{n,H-A}$ and $T_{n,A-H}$ within the previous period L_n. Coordinating a set of agents $\{A_1, \cdots, A_n\}$ corresponds to executing a set of tasks $\Gamma = \{\tau_1, ..., \tau_N\}$ on a single core processor. Based on the above analysis, we apply the "highest-trust-first" scheduling method in the multi-agent systems [43,59]. When the human-to-agent trust approaches to the upper limit, the human operator tends to over trust. When the agent-to-human trust goes below the lower limit, the agent tends to under trust the human operator. In both cases, manual control is required and hence the agent should have priority to be chosen to collaborate with the human operator. Therefore, the "highest-trust-first" scheduling method will select the agent with trust values approaching to the corresponding limits and give it the highest priority to be manually operated. As we discussed in the previous section, we will not use Definition 3.1.1 to test if a given human multi-agent collaboration system is schedulable. Instead, we develop necessary and sufficient conditions to achieve this goal by adopting a dynamic timing model to save computational efforts [61].

To describe the current status of agents at any time step k and build up the dynamic timing model, we introduce an evolution model of a state vector $Z(k) = [\mathbf{Q}(k), \mathbf{S}(k), \mathbf{O}(k)]$ within any sub-interval $[k_w, k_{w+1}]$, where $P_w = k_{w+1} - k_w$ is the length of each sub-interval. Note that our study is under the discrete-time setting. More details about the definition of sub-intervals and division procedure from one given period into a series of sub-intervals can be found in [59].

The state $\mathbf{Q}(k) = [q_1(k), ..., q_N(k)]$ indicates how long after time step, k, the next collaboration request from A_n will launch, and it satisfies the following update equations:

$$\begin{cases} q_n(k_w) = q_n(k_w - 1) - 1, & \text{if } q_n(k_w - 1) > 1 \quad (3.1a) \\ q_n(k_w) = L_n, & \text{if } q_n(k_w - 1) = 1 \quad (3.1b) \\ q_n(k_w + \epsilon_w) = q_n(k_w) - \epsilon_w, & \forall\, \epsilon_w \in [1, P_w - 1] \quad (3.1c) \end{cases}$$

where ϵ_w is an integer representing time step. Equations (3.1a) and (3.1b) show the evo-

lution of the state vector $q_n(k)$ from the end of the last sub-interval to the beginning of the current sub-interval. As we can see, this evolution may jump, depending the value of $q_n(k_w - 1)$. On the other hand, Equation (3.1c) shows that the evolution of $q_n(k)$ within the sub-interval $[k_w + 1, k_w + P_w - 1]$, which is consistent. The initial condition during the whole time interval is set as $q_n(k_a) = L_n$.

We choose $P_w \leq \min\{q_1(k_w), ..., q_N(k_w)\}$ so that the requests from agents only arrive at k_w, but not at any other time step within $[k_w + 1, k_w + P_w - 1]$. Hence, the largest possible window length P_w can be expressed as: $P_w = \min\{q_1(k_w), ..., q_N(k_w), k_b - k_w\}$ [59].

The state $\mathbf{S}(k) = [s_1(k), ..., s_N(k)]$ indicates the remaining collaboration time required after time step, k, by agent A_n, and it satisfies the following update equations:

$$\begin{cases} s_n(k_w) = \max\{0, s_n(k_w - 1) - \max\{0, 1 - \sum_{i \in HP} u_i(k_w - 1)\}\}, & \text{(3.2a)} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } q_n(k_w - 1) > 1 & \\ s_n(k_w) = I_n(k_w), \qquad\qquad \text{if } q_n(k_w - 1) = 1 & \text{(3.2b)} \\ s_n(k_w + \epsilon_w) = \max\{0, s_n(k_w) - \max\{0, \epsilon_w - \sum_{i \in HP} \sum_{k=k_w}^{k_w+\epsilon_w} u_i(k)\}\}, & \text{(3.2c)} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall\, \epsilon_w \in [1, P_w - 1] & \end{cases}$$

where the term HP represents a set containing all the agents which have higher priority than A_n. Equations (3.2a) and (3.2b) show the evolution of the state vector $s_n(k)$ from the end of the last sub-interval to the beginning of the current sub-interval. Equation (3.2d) shows that the evolution of $s_n(k)$ within the sub-interval $[k_w + 1, k_w + P_w - 1]$. The initial condition for $\mathbf{S}(k)$ is $s_n(k_a) = I_n(k_a)$. In addition, if $q_n(k_w - 1) = 1$ and $s_n(k_w - 1) > 1$, the system will be unschedulable as the collaboration time for one agent will exceed its period.

The state $\mathbf{O}(k) = [o_1(k), ..., o_N(k)]$ indicates two situations: (1) When the human

still collaborates with the agent A_n, $o_n(k)$ denotes the length of time from the initial request of A_n to the current time step, k; (2) When the collaboration between the human and the agent A_n is completed, $o_n(k)$ denotes the length of time from the initial request of A_n to the collaboration completion time, and it satisfies the following update equations:

$$\begin{cases} o_n(k_w) = o_n(k_w-1) + \mathrm{sgn}(s_n(k_w-1)), & \text{if } q_n(k_w-1) > 1 \quad (3.3a) \\ o_n(k_w) = 0, & \text{if } q_n(k_w-1) = 1 \quad (3.3b) \\ o_n(k_w+\epsilon_w) = o_n(k_w) + \mathrm{sgn}(s_n(k_w)) \min\{s_n(k_w) + \sum_{i\in HP} s_i(k_w), \epsilon_w\}, & (3.3c) \\ \qquad\qquad \forall\, \epsilon_w \in [1, P_w - 1] \end{cases}$$

where $s_n(k_w) + \sum_{i\in HP} s_i(k_w)$ denotes the total time steps propagated before the agent A_n finishes collaboration with the human operator. Equations (3.3a) and (3.3b) show the evolution of the state vector $o_n(k)$ from the end of the last sub-interval to the beginning of the current sub-interval. Equation (3.3d) shows that the evolution of $o_n(k)$ within the sub-interval $[k_w + 1, k_w + P_w - 1]$. The initial condition is $o_n(k_a) = 0$.

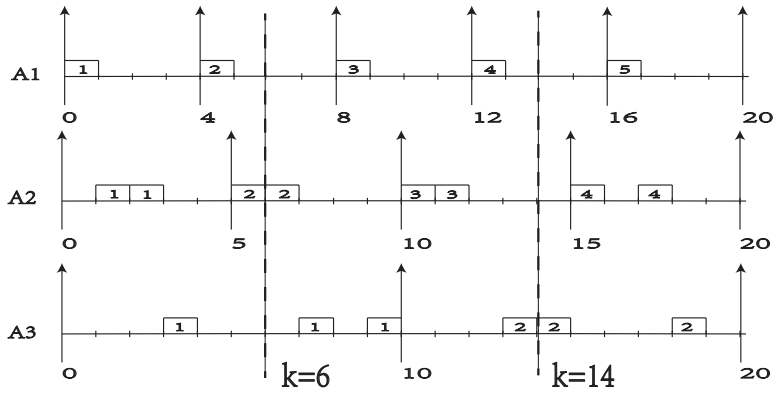


Figure 3.1: Three acyclic agents collaborating with one operator

Example 3.2.1. *Consider agents* $\{A_1, A_2, A_3\}$ *with* $[I_1(k), I_2(k), I_3(k)] = [1, 2, 3]$ *and* $[L_1, L_2, L_3] = [4, 5, 10]$. *The three acyclic agents are scheduled under a fixed priority*

preemptive scheduling algorithm such that the agent with the smallest interaction time gets the highest priority.

Fig. 3.1 demonstrates the scheduled behavior of $\{A_1, A_2, A_3\}$. The upper arrows indicate the time steps when the collaboration request from agents launch.

We can observe that at time step $k = 6$, the next collaboration request from agents $\{A_1, A_2, A_3\}$ will launch at 8, 10, and 10 respectively. Thus, we have

$$Q(6) = [q_1(6), q_2(6), q_3(6)] = [8 - 6, 10 - 6, 10 - 6] = [2, 4, 4].$$

After time step $k = 6$, A_2 and A_3 have NOT finished collaboration with human operator in their corresponding periods. Hence, we have

$$S(6) = [s_1(6), s_2(6), s_3(6)] = [0, 1, 2].$$

For A_1, it launches collaboration request at time step 4 and finished collaboration with human operator at current time step in its second period. Hence, $o_1(6) = 1$. For A_2, it launches collaboration request at time step 5 and has NOT finished collaboration with human operator at time step 6 in its second period. Hence, $o_2(6) = 1$. For A_3, it launches collaboration request at time step 1 and has NOT finished collaboration with human operator at time step 6 in its first period. Hence, $o_3(6) = 6$. Therefore, we have

$$O(6) = [o_1(6), o_2(6), o_3(6)] = [1, 1, 6].$$

Similarly, at time step $k = 14$, we have state vectors as

$$Q(14) = [2, 1, 6], S(14) = [0, 0, 2], O(14) = [1, 2, 4].$$

Based on the evolutions of state variables and Definition 3.1.1, the schedulability of the human-agent collaboration system can now be redefined as follows.

Definition 3.2.1. *A human multi-agent collaboration system is schedulable within time interval* $[k_a, k_b]$ *if and only if the system is schedulable within each sub-interval* $[k_w, k_{w+1}] \in [k_a, k_b]$. *The system is schedulable within a sub-interval* $[k_w, k_{w+1}]$ *if and only if each individual agent* A_n *is schedulable within* $[k_w, k_{w+1}]$.

The following theorems state the necessary and sufficient conditions for the schedulability of an agent A_n within a sub-interval $[k_w, k_{w+1}]$.

Theorem 3.2.1. (Sufficient Condition) *An agent* A_n *is schedulable within* $[k_w, k_{w+1}]$ *if it satisfies one of the following conditions:*

$$1.\ o_n(k_{w+1} - 1) = L_n - 1 \text{ and } s_n(k_{w+1} - 1) = 0;$$

$$2.\ o_n(k_{w+1} - 1) < L_n - 1.$$

The proof can be found in [59]. The following corollary gives a less conservative sufficient condition.

Corollary 3.2.1. (Sufficient and necessary Condition) *An agent* A_n *is schedulable if and only if it satisfies one of the following conditions within* $[k_w, k_{w+1}]$*:*

$$1.\ o_n(k_{w+1} - 1) = L_n - 1 \text{ and } s_n(k_{w+1} - 1) = 0 \text{ or } 1;$$

$$2.\ o_n(k_{w+1} - 1) < L_n - 1.$$

The extension made by Corollary 3.2.1 is the addition of condition, $s_n(k_{w+1} - 1) = 1$.

Proof. For the first condition, we have the following proof. The schedulability of A_n within $[k_w, k_{w+1}]$ is satisfied if the dynamic response time of A_n is equal to $L_n - 1$ at time step $k_{w+1} - 1$, and the effective collaboration request from A_n has been completed by the time step $k_{w+1} - 1$, i.e., $s_n(k_{w+1} - 1) = 0$. Consider the case when the dynamic response time of A_n is equal to $L_n - 1$ at time step $k_{w+1} - 1$, and the effective collaboration request from A_n has not been completed by the time step $k_{w+1} - 1$. If the task remains active for only one more time step, i.e., $s_n(k_{w+1} - 1) = 1$, the schedulability of A_n within $[k_w, k_{w+1}]$ is still satisfied.

For the second condition, if the dynamic response time of A_n is smaller than $L_n - 1$ at time step $k_{w+1} - 1$, the schedulability of A_n within $[k_w, k_{w+1}]$ is automatically satisfied. □

Utilizing Corollary 3.2.1, the schedulability of the human multi-agent collaboration system does not need to be checked at every time step according to Definition 3.1.1. Instead, the entire time interval $[k_a, k_b]$ is decomposed into several sub-intervals $[k_w, k_{w+1}]$ and the system schedulability only needs to be checked at the beginning of each sub-interval. Clearly, this method requires less computation resources.

3.3 Schedulability Test Algorithm

We can now perform the dynamic schedulability test over the time interval $[k_a, k_b]$ using an algorithm based on Theorem 3.2.1. This algorithm is composed of 6 parts, as shown below. It iteratively checks the schedulability of each agent, A_n.

Description of Algorithm 3.1: First, the human performance model, $P_H(k_a)$, is measured based on the human utilization and task difficulty, by Equation (2.4). Meanwhile, we can obtain the agent performance model, $P_{n,A}(k_a)$, based on Equation (2.3). Next, $T_{n,H-A}(k_a)$ and $T_{n,A-H}(k_a)$ can be calculated, based on Equations (2.1) and (2.2). The

Algorithm 3.1: Main Algorithm

Data: $r(k_a)$, $P_H(k_a)$, $\{u_n(k_a)\}_{n=1}^N$, $\{L_n, I_n(k_a)\}_{n=1}^N$, $\{P_{n,A}(k_a), T_{n,H-A}(k_a), T_{n,A-H}(k_a), q_n(k_a), s_n(k_a), o_n(k_a)\}_{n=1}^N$

Result: $\{DS_n\}_{n=1}^N$

1 **for** *each* $A_n \in \Gamma$ **do**
2 $\quad DS_n = [\,]$;
3 $\quad ds_n = 1$;
4 $k = 1$;
5 $w = 1$;
6 $k_w = k_a$;
7 Algorithm 3.2;
8 **return** $\{DS_n\}_{n=1}^N$;

state vectors [**Q(k), S(k), O(k)**] are achieved from Equations (3.1), (3.2), and (3.3). Finally, we choose the initial collaboration time $I_n(k_a)$ within the period of L_n. For each agent, we initiate the schedulability result, ds_n, as 1, which means it is schedulable at the beginning. We use the variable ds_n to represent the schedulability result with $ds_n = 1$ representing schedulable and $ds_n = 0$ otherwise. The set DS_n contains the schedulability results during the time interval $[k_a, k_b]$.

Description of Algorithm 3.2: We choose $k = k_a$ as the beginning time of the first fixed priority window, $k_{w=1}$, and then determine whether k_w is currently within the time range $[k_a, k_b]$. As Lines 3-7 show, we check each fixed priority window according to state vector, **Q(k)**.

Description of Algorithm 3.3: If the system is schedulable, i.e., $ds_n = 1$, we first update the performance $\{P_{n,A}, P_H\}$, human-to-agent trust level, $T_{n,H-A}(k)$, and agent-to-human trust level, $T_{n,A-H}(k)$. At the beginning of a new sub-interval, the corresponding ϵ_w will be reset and start to count from 1. State vectors [**Q(k), S(k), O(k)**] can be then updated based on Equations (3.1), (3.2), and (3.3). k shown in Lines 8 and 10 is within the time interval $[k_w, k_w + \epsilon_w]$.

Algorithm 3.2: Determination of the fixed priority window

Data: $r(k_a)$, $P_H(k_a)$, $\{u_n(k_a)\}_{n=1}^N$, $\{L_n, I_n(k_a)\}_{n=1}^N$, $\{P_{n,A}(k_a), T_{n,H-A}(k_a), T_{n,A-H}(k_a), q_n(k_a), s_n(k_a), o_n(k_a)\}_{n=1}^N$

Result: P_w

while $k_w < k_b$ **do**
  for *each* A_n **do**
    if $q_n(k_w - 1) == 1$ **then**
      $q_n(k_w) = L_n$;
    else
      $q_n(k_w) = q_n(k_w - 1) - 1$;
  /* The length of the current fixed priority window P_w */
  $P_w = \min\{q_1(k_w), ..., q_N(k_w), k_b - k_w\}$;
  $k_{w+1} = k_w + P_w$;
  Algorithm 3.3;
  $w = w + 1$;
return P_w ;

Description of Algorithm 3.4: We first calculate the error vector for next time step, $k+1$, based on the difference between the upper limit and the current trust level in the human-to-agent case and the difference between the current trust level and the lower limit in the agent-to-human case. The term $e_{n,H-A}(k+1) = T_{n,H-A,u} - T_{n,H-A}(k)$ represents the deviation of the current human-agent trust level away with the upper limit. When the human-to-agent trust level approaches to the upper limit, the system is then close to the "over-trust" situation, which means that the manual should be activated. Similarly, we define another term $e_{n,A-H}(k+1) = T_{n,A-H}(k) - T_{n,A-H,l}$. When the agent-to-human trust level approaches to the lower limit, it represents the "under-trust" situation, which also suggests the activation of manual mode. The agent with the minimum error will be chosen to collaborate with.

Description of Algorithm 3.5: We update the collaboration time of each agent $I_n(k)$ by calculating the maximum and minimum mutual trust levels. When the human-to-agent trust level becomes too high or the agent-to-human trust level becomes too low,

Algorithm 3.3: Update of the trust level and state variables for dynamic timing model

Data: $r(k_a)$, $P_H(k_a)$, $\{u_n(k_a)\}_{n=1}^N$, $\{L_n, I_n(k_a)\}_{n=1}^N$, P_w
$\{P_{n,A}(k_a), T_{n,H-A}(k_a), T_{n,A-H}(k_a), q_n(k_a), s_n(k_a), o_n(k_a)\}_{n=1}^N$

Result: $\{T_{n,H-A}(k), T_{n,A-H}(k), q_n(k), s_n(k), o_n(k)\}_{n=1}^N$

if $ds_n == 1$ **then**
  $\epsilon_w = 1$;
  $q_n(k_w) \xleftarrow{\text{Eq.}(3.1a)\&(3.1b)} \{q_n(k_w - 1), L_n\}$;
  $s_n(k_w) \xleftarrow{\text{Eq.}(3.2a)\&(3.2b)} \{s_n(k_w - 1), \{u_n(k_w)\}_{n=1}^N, q_n(k_w - 1), I_n(k_w)\}$;
  $o_n(k_w) \xleftarrow{\text{Eq.}(3.3d)\&(3.3d)} \{s_n(k_w - 1), \{u_n(k_w)\}_{n=1}^N, q_n(k_w - 1)\}$;
  while *within the subinterval* $[k_w, k_{w+1}]$ **do**
    for $each\ A_n$ **do**
      $P_{n,A}(k+1) \xleftarrow{\text{Eq.}(2.3)} \{P_{n,A}(k), u_n(k)\}$;
      $P_H(k+1) \xleftarrow{\text{Eq.}(2.4)} r(k)$;
      $T_{n,H-A}(k+1) \xleftarrow{\text{Eq.}(2.1)}$
      $\{T_{n,H-A}(k), P_{n,A}(k+1), P_{n,A}(k), F_A(k+1), F_A(k)\}$;
      $T_{n,A-H}(k+1) \xleftarrow{\text{Eq.}(2.2)}$
      $\{T_{n,A-H}(k), P_H(k+1), P_H(k), F_H(k+1), F_H(k)\}$;
      $q_n(k_w + \epsilon_w) \xleftarrow{\text{Eq.}(3.1c)} \{q_n(k_w)\}$;
      $s_n(k_w + \epsilon_w) \xleftarrow{\text{Eq.}(3.2d)} \{s_n(k_w), \{u_n(k_w)\}_{n=1}^N, q_n(k_w)\}$;
      $o_n(k_w + \epsilon_w) \xleftarrow{\text{Eq.}(3.3d)} \{s_n(k_w), \{u_n(k_w)\}_{n=1}^N, q_n(k_w)\}$;
    Algorithm 3.4;
    Algorithm 3.5;
    $\epsilon_w = \epsilon_w + 1$;
    $k = k + 1$ **;**
for $each\ A_n$ **do**
  Algorithm 6;
return $\{T_{n,H-A}(k), T_{n,A-H}(k), q_n(k), s_n(k), o_n(k)\}_{n=1}^N$;

Algorithm 3.4: Agent selection

Data: $\{T_{n,H-A}(k), T_{n,A-H}(k), s_n(k)\}_{n=1}^{N}$

Result: i

1 **for** *each* A_n **do**

  /* G is a set of agents with the non-zero remaining collaboration time*/

2 $e_{n,H-A}(k+1) = T_{n,H-A,u} - T_{n,H-A}(k)$;

3 $e_{n,A-H}(k+1) = T_{n,A-H}(k) - T_{n,A-H,l}$;

4 $G = [\,]$;

5 **if** $s_n(k_w + \epsilon_w) > 0$ **then**

6 $G = [\,G,\ A_n\,]$;

7 **if** G *is not empty* **then**

8 $i = \min_{A_n \in G}([e_{n,H-A}(k+1), e_{n,A-H}(k+1)])$;

9 **if** $n == i$ **then**

10 $u_n(k+1) = 1$;

11 **else if** $n \neq i$ **then**

12 $u_n(k+1) = 0$;

13 **return** i ;

we increase the amount of collaboration time, as shown in Lines 6-11. Note that the value of $I_n(k)$ cannot go beyond L_n. On the other hand, when the human-to-agent trust level becomes too low or the agent-to human trust level becomes too high, we decrease the amount of collaboration time, as shown in Lines 12-17. Note that the value of $I_n(k)$ cannot be smaller than zero. Besides, $\delta_1, \delta_2, \delta_3, \delta_4 > 0$ are constant small values guaranteeing that $I_n(k)$ is adjusted before the mutual trust level goes beyond the upper and lower limits.

Description of Algorithm 3.6: We first determine whether the time instance is a new launched time point of the request from A_n, as shown in Line 1. If so, based on the updated state variables and sufficient and necessary conditions as Corollary 3.2.1 shows, we will perform dynamic schedulability test. Next, the above process repeats for $n = 1, \cdots, N$. Finally, we will end the entire loop and output the schedulability results.

Algorithm 3.5: Update of the collaboration time

Data: $\{T_{n,H-A}(k), T_{n,A-H}(k), L_n, I_n(k_a)\}_{n=1}^N$

Result: $\{I_n(k)\}_{n=1}^N$

for *each* A_n **do**

  $T_{n,H-A,\max} = \max_{k-L_n \le \tau \le k} T_{n,H-A}(\tau)$;

  $T_{n,A-H,\max} = \max_{k-L_n \le \tau \le k} T_{n,A-H}(\tau)$;

  $T_{n,H-A,\min} = \min_{k-L_n \le \tau \le k} T_{n,H-A}(\tau)$;

  $T_{n,A-H,\min} = \min_{k-L_n \le \tau \le k} T_{n,A-H}(\tau)$;

  if $T_{n,H-A,\max} > T_{n,H-A,u} - \delta_1$ and $T_{n,H-A,\min} > T_{n,H-A,d}$ **then**

    $\phi_1 = 1$;

  if $T_{n,A-H,\min} < T_{n,A-H,l} + \delta_2$ and $T_{n,A-H,\max} < T_{n,A-H,d}$ **then**

    $\phi_2 = 1$;

  if $\phi_1 \cup \phi_2 = 1$ **then**

    $I_n(k) = \min\{I_n(k) + 1, L_n\}$;

  if $T_{n,H-A,\min} < T_{n,H-A,l} + \delta_3$ and $T_{n,H-A,\max} < T_{n,H-A,d}$ **then**

    $\phi_3 = 1$;

  if $T_{n,A-H,\max} > T_{n,A-H,u} - \delta_4$ and $T_{n,A-H,\min} > T_{n,A-H,d}$ **then**

    $\phi_4 = 1$;

  if $\phi_3 \cup \phi_4 = 1$ **then**

    $I_n(k) = \max\{I_n(k) - 1, 0\}$;

return $\{I_n(k)\}_{n=1}^N$;

Algorithm 3.6: Schedulability test

Data: L_n, $\{q_n(k), s_n(k), o_n(k)\}_{n=1}^{N}$

Result: $\{DS_n\}_{n=1}^{N}$

if $q_n(k_{w+1}-1) == 1$ **then**
  if $o_n(k_{w+1}-1) < L_n - 1$ **then**
    $ds_n = 1$ within the subinterval $[k_w, k_{w+1}]$;
  else if $o_n(k_{w+1}-1) = L_n - 1$ and $s_n(k_{w+1}-1) = 0$ or 1 **then**
    $ds_n = 1$ within the subinterval $[k_w, k_{w+1}]$;
  else
    $ds_n = 0$ within the subinterval $[k_w, k_{w+1}]$;
else
  $ds_n = ds_n$ within the subinterval $[k_w, k_{w+1}]$;
$DS_n = \{DS_n, ds_n\}$;
return $\{DS_n\}_{n=1}^{N}$;

3.4 Simulation Results

3.4.1 Parameter setup

We simulate the scenario when a human operator collaborates with three heterogeneous agents: $\{A_1, A_2, A_3\}$. The agent performance can be updated by Equation (2.3) and the choice of parameters for each agent is listed in Table 3.1. Each agent has its initial performance as $[P_{1,A}(0), P_{2,A}(0), P_{3,A}(0)] = [0.08, 0.15, 0.11]$. The human operator has his/her performance as described by Equation (2.4). We assume that the task difficulty for the human operator is $\beta = 0.8$ and the maximum human performance, $P_{H,\max}$, and minimum human performance, $P_{H,\min}$, are 1 and 0, respectively. The initial human performance is $P_H(0) = 0.25$ and the initial utilization ratio is $r(0) = 0.1$.

The mutual trust levels between the human operator and each agent A_n follow the dynamic models discussed in Equations (2.1) and (2.2). The constant coefficients in these equations are chosen as $A_{n,1} = 1$, $A_{n,2} = 1$, $B_{n,1} = -0.5$, $B_{n,2} = 0.5$, $C_{n,1} = -0.5$, $C_{n,2} = 0.5$, $D_{n,1} = 0.005$, $D_{n,2} = 0.005$, $E_{n,1} = 0.005$, $E_{n,2} = 0.005$ and the fault rates

Table 3.1: Coefficients in Agent Performance Model

	k_R	k_H	$P_{A,\min}$	$P_{A,\max}$
A_1	0.17	0.25	0.02	0.85
A_2	0.15	0.25	0.05	0.96
A_3	0.25	0.17	0.04	0.9

follow the standard normal distribution $N(0,1)$. The initial mutual trust values between the human operator and three agents are assumed to be $[T_{H-A,1}(0), T_{H-A,2}(0), T_{H-A,3}(0)] = [1.93, 1.9, 1.98]$ and $[T_{A-H,1}(0), T_{A-H,2}(0), T_{A-H,3}(0)] = [1.93, 1.9, 1.98]$. The goal is to make sure that mutual trust level $T_n(k) = [T_{n,H-A}(k); T_{n,A-H}(k)]$ stays within a desired trust region as time propagates. In this simulation, we choose the desired trust regions with the lower bounds $T_{1,l} = 1.45, T_{2,l} = 1.35, T_{3,l} = 1.25$, the upper bounds $T_{1,u} = 2.15, T_{2,u} = 2.35, T_{3,u} = 2.25$, and the ideal expert level $T_{n,d} = \frac{T_{n,u}+T_{n,l}}{2}$ for each agent. Note that we set the same bound for both human-to-agent and agent-to-human trust regions.

As discussed in Section 3.2, we choose the initial parameters in the periodic strategy as

$$[I_1, L_1] = [2, 10]\text{s} \quad [I_2, L_2] = [3, 10]\text{s} \quad [I_3, L_3] = [4, 10]\text{s} \tag{3.4}$$

where each pair $[I_n, L_n]$ for $n = 1, 2, 3$ denotes that the human operator must collaborate with the agent A_n for I_n seconds within every L_n seconds. Note that the value of $I_n(k)$ will dynamically change according to Algorithm 5.

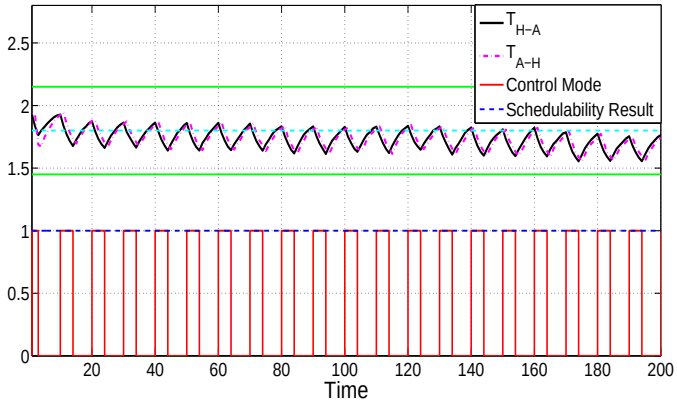
3.4.2 Results and discussions

Fig. 3.2 shows the evolution of the human-to-agent trust level and agent-to-human trust level for all three agents within the time interval $[1, 200]$. The green lines represent the upper bound and lower bound of the desired trust regions. The cyan dashed lines represent

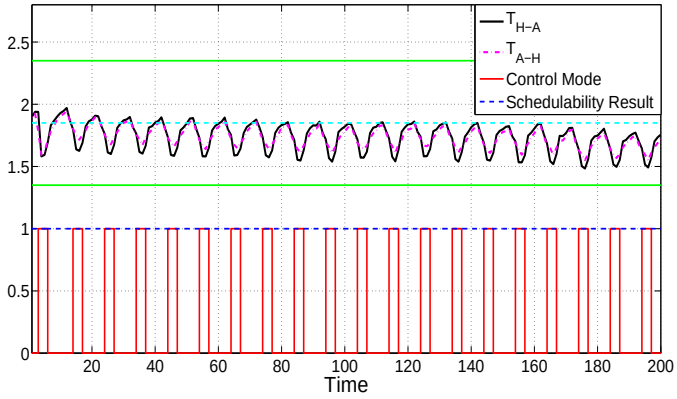
the ideal expert level. The black lines represent the human-to-agent trust level and the pink dotted lines represent the agent-to-human trust level. The red squares represent the (de)activation of manual control. We can observe that the human operator interacts with one agent at a time. Here, "1" means that the human operator is collaborating with the agent; and "0" means no collaboration. Finally, the blue dotted lines represent the schedulability results, where "1" means that the system is schedulable and "0" means NOT schedulable. From the figure, it can be seen that the mutual trust level in each human-agent pair is consistently bounded within the desired regions and the human-agent system is always schedulable, which indicates that the proposed scheduling algorithm could guarantee the ultimate goal. Besides, the plenty amount of time between one control mode and another suggests that the human has enough time to respond, decide and take an action. Hence, this scheduling scheme can be adopted into the more practical human-in-the-loop applications.

As mentioned above, in order to save computational efforts, we set up a dynamic timing model and develop necessary and sufficient conditions to check the schedulability. To verify this computational efficiency improving, we calculate the computational time of the schedulability test by using two methods and compare results. The first one is that we compute the mutual trust values and check whether they fall within the desired trust regions at each time step, as Definition 3.1.1 shows. Another one is that we test the schedulability by necessary and sufficient conditions based on the dynamic timing model. The comparison experiment is performed on a Dell laptop with Processor 2.4 GHz, Intel Core i7-4500U, and Memory 12GB by using Matlab version 2015Ra. For the convenience of comparing, each simulation window length is extended tenfold, i.e., $[0, 2000]$. We run both methods 50 times and then calculate the averaged computational time of the schedulability test for each method. After calculating, we can see that the runtime of the second method is approximately 6 times faster than the first one. Hence, we can conclude that there is an obvious computational efficiency improvement for the schedulability test based on the

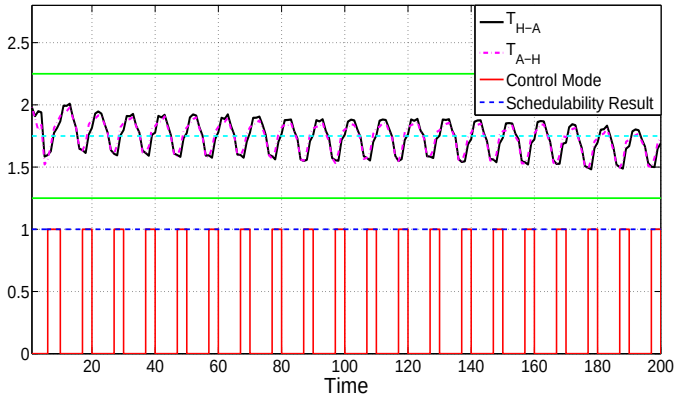
dynamic timing model.



(a) $T_1(k)$ between the human and agent 1



(b) $T_2(k)$ between the human and agent 2



(c) $T_3(k)$ between the human and agent 3

Figure 3.2: Mutual trust within the time interval $[1, 200]$.

Chapter 4

FORMATION CONTROL FOR LARGE-SCALE HUMAN-SWARMS COLLABORATION SYSTEMS BASED ON MUTUAL TRUST

4.1 Human-swarms collaboration systems

Swarm systems are distributed in nature and well-suited for tasks that are concerned with the space [19, 33, 44, 54]. Environmental monitoring and freightage are examples of swarms application [38]. Take freightage as an example, which has a wide range of civilian and military applications. When heavy objects are needed to move in constrained terrains, we can deploy a swarm of agents to save labor cost and satisfy space requirements. Although an agent, like UGV, is small enough to meet the demand of terrain, it can not afford the weight. Therefore, a swarm of agents will be used to support this kind of objects. Based on the object's size and shape, several swarms will be employed. Hence, control should be

considered for swarms to satisfy the requirements of the space. The swarm coordination and control problems have attracted considerable attention from researchers recently. Several control methods have been proposed. In the paper [35], Kim et al present a framework for decentralized control of self-organizing swarm systems based on the artificial potential functions. Agents in a swarm self-organize to flock and achieve formation through attractive and repulsive forces among themselves using the artificial potential functions. Gazi et al, in [22], design continuous-time control schemes for swarms via a constructive analysis based on artificial potential functions and sliding mode control techniques. In the paper [47], Morgan et al develop a model predictive control implementation, which provides fuel-optimal collision-free motion for the reconfiguration of swarms of spacecrafts. Sayama, in [57], presents new methods of decentralized control with which swarms of agents can spontaneously organize and maintain non-trivial heterogeneous formations. The swarm formation control method proposed in [2] based on potential fields satisfies scalability (to varying numbers of swarm members) and supports multiple formations. In the paper [13], Cheng et al describe a decentralized algorithm for coordinating a swarm of identically-programmed mobile agents to spatially self-aggregate into arbitrary shapes using only local interactions. In [14], Coppin et al analyze the way representational gaps between man and swarms-agents can impact global performances. In [36], Kolling et al investigate principles of control for large swarms and to determine how human perform in controlling swarms using implementations of these principles for a complex foraging task in a variety of challenging environments.

To construct a swarm system, we imitate the dynamic model from biological systems [49]. Wild geese in biology usually arrange in different formations when they migrate seasonally. There is generally a leading wild goose in a flock. Similarly, for land animals, such as zebra and horses, when they move to other territories, there always exists one leader in the herd [70]. Besides, fish is always divided into numerous schools based on d-

ifferent destinations, during mass migration [10]. Inspired by these phenomena in biology, we imagine a leading agent in a swarm as a leading animal in a flock [11, 30]. That is to say, each leading agent is responsible for its belonging swarm. Furthermore, other agents in the swarm will follow the motion of this leading agent. Hence, a leading agent and its followers constitute a swarm system.

In a network consisted of several swarm systems, communication among leading agents is not easily available due to limitation of bandwidth. This is because as the size of a swarm system increases, the available bandwidth to the leader decreases [27]. Furthermore, the leading agent will always keep in touch with its followers to direct them, occupying for its own limited communication bandwidth. Besides, agents have strict communication constraints asking team members to be in close physical proximity to communicate [4]. Hence, a leading agent prefers to communicate with nearby agents. This is also often observed in biology swarms [62, 63]. However, when we employ such a network in the practical application, communication within the network is necessary. To address this issue, we are used to adopting a centralized control system to distribute information [18]. Nevertheless, for a large-scaled network, i.e., a network composed of a large number of swarm systems, human operators will be adopted. It will be critical to include the human at some level of decision making within these swarming networks both as a safety check and also to ensure that the automation is truly supporting overall mission goals [16]. Since the number of complex problems and the space of possibilities which swarms will assist in are vast, enabling the control of swarms by a human operator who could interact with the autonomy and adapt to specific challenges in a variety of conditions is still crucial [36]. Therefore, it is necessary for a human team to join in the swarms' loop.

Although some researchers have talked about the collaboration between human and swarms [14, 16, 36, 36], human operators are just on the swarms' loop, which means human only supervise the swarms' work. Nevertheless, in our case, we incorporate the human fac-

tors into the swarms' loop, which means human will truly cooperate with swarms. Besides, to measure the collaboration between human network and swarm network, we propose a novel measurement, called 'fitness'. Furthermore, using this novel measurement, we propose a scheduling algorithm to guarantee a successful human-swarm collaboration system.

4.2 Swarm setup

For a human-swarm collaboration system, each swarm has its own task. In this part, we focus on the motion of a swarm of mobile agents in the planar space, which means each swarm has its own coordinate. We consider the motion of a swarm under two modes: manual model and autonomous mode. The manual mode means a human operator inputs the neighbouring swarms' information, such as position and velocity, to the leader of the current collaborating swarm. If this operator provides the neighbouring information at every time instant during a time interval, we say the human controls a swarm perfectly or the swarm is under pure manual mode. On the other hand, the autonomous means the leading agent and its followers run automatically based on the initial given neighbours' information. That is, the neighbours' information will not be updated. Here, we relate the swarm performance with the formation shape constructed by the leading agent and the diversity of the following agents in this swarm. Hence, under the manual mode, the more information a leader knows, the better its belonging swarm performance. On the contrary, the less information a leader knows, the worse the swarm performance. To save human resources, we wish less human operators collaborate with more swarms, which leads to the time delay of information input. This time delay results from the allocation of human attention since one human can only collaborate with one swam at one time. We define the

time delay as τ, which satisfies the following update equation.

$$\tau_m = o_m(t) - \max\{0, I_m - s_m(t)\}, \text{if } s_m(t) = s_m(t-1) \text{ and } o_m(t) > 0 \tag{4.1}$$

where $o_m(t)$ is the dynamic response time for swarm m, I_m is the collaboration time with swarm m, and $s_m(t)$ is the residue time. The update equations of the dynamic response time and the residue time can be found in Chapter 3.2. $I_m - s_m(t)$ denotes the loss of collaboration time. $s_m(t) = s_m(t-1)$ means the human stopping collaboration with the swam m. When the human stops collaborating with the swarm, the propagation of dynamic response time removing the loss of collaboration time represents the delay time.

Assume that the horizontal and vertical coordinates of a position, $\mathbf{q} = (x, y)^T$, are independent. The dynamics of the leading agent in the swarm m is given by [25, 50]

$$\dot{\mathbf{Q}}_m(t) = \begin{cases} \Phi\ \mathbf{Q}_m(t) + \Gamma\ \mathbf{u_m}(t), & \text{(manual mode)} \\ \Phi\ \mathbf{Q}_m(t) + \Gamma\ \mathbf{u_m}(t - \tau_m), & \text{(autonomous mode)} \end{cases} \tag{4.2}$$

$$\Phi = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} 0 & 0 \\ -k_f & 0 \\ 0 & 0 \\ 0 & -k_f \end{bmatrix}$$

$$\mathbf{Q}_m(t) = [x_m(t), \dot{x}_m(t), y_m(t), \dot{y}_m(t)]^T$$

$$\mathbf{u_m}(t) = [u_{mx}(t), u_{my}(t)]^T$$

where k_f is the positive feedback gain.

The $\mathbf{u}_m$ works as formation control input to shape several clusters, aiming to shape a regular N-polygon. Hence, the formulation of $\mathbf{u}_m$ is given by

$$\mathbf{u_m}(t) = \begin{bmatrix} \sum_{n=1,n\neq m}^{N} \frac{x_m(t)-x_n(t)-Rdx_{m,n}}{\sqrt{(x_m(t)-x_n(t)-Rdx_{m,n})^2+(y_m(t)-y_n(t)-Rdy_{m,n})^2}} \\ \sum_{n=1,n\neq m}^{N} \frac{y_m(t)-y_n(t)-Rdy_{m,n}}{\sqrt{(x_m(t)-x_n(t)-Rdx_{m,n})^2+(y_m(t)-y_n(t)-Rdy_{m,n})^2}} \end{bmatrix} \tag{4.3}$$

where N indicates the number of clusters in the space, $Rdx_{m,n}$ and $Rdy_{m,n}$ represent the respective desired distance between the center of a cluster m and neighbour clusters, $(x_m, y_m)^T$ marks the position of the leader of swarm m in the Cartesian coordinate system, and $(x_n, y_n)^T$ stands for the position of the leader of neighbouring swarm n.

When the leading agent lies under autonomous mode, we put the control in the form, developed by

$$\mathbf{u_m}(t-\tau) = \begin{bmatrix} \sum_{n=1,n\neq m}^{N} \frac{x_m(t)-x_n(t-\tau)-Rdx_{m,n}}{\sqrt{(x_m(t)-x_n(t-\tau)-Rdx_{m,n})^2+(y_m(t)-y_n(t-\tau)-Rdy_{m,n})^2}} \\ \sum_{n=1,n\neq m}^{N} \frac{y_m(t)-y_n(t-\tau)-Rdy_{m,n}}{\sqrt{(x_m(t)-x_n(t-\tau)-Rdx_{m,n})^2+(y_m(t)-y_n(t-\tau)-Rdy_{m,n})^2}} \end{bmatrix} \tag{4.4}$$

Based on the system shown in Equation (4.2), we first define a Lyapunov function as [25],

$$V(t) = \tfrac{1}{2} \sum_{m=1}^{N} \left\{ \dot{x}_m(t)^2 + \dot{y}_m(t)^2 + \sum_{n=1, n\neq m}^{N} 2k_f \sqrt{(x_m(t) - x_n(t) - Rdx_{m,n})^2 + (y_m(t) - y_n(t) - Rdy_{m,n})^2} \right\},$$

further obtaining $\dot{V} = \nabla V \cdot \dot{\mathbf{Q}}$.

Theorem 4.2.1. *If a leading agent in the swarm m is under pure manual mode, the control law $\mathbf{u_m}(t)$ given by (4.3) drives $\dot{x}_m(t) \to 0$, $\dot{y}_m(t) \to 0$, $x_m(t) - x_n(t) \to Rdx_{m,n}$ $(n \neq m)$, and $y_m(t) - y_n(t) \to Rdy_{m,n}$ $(n \neq m)$*

Proof. Computing $\dot{V} = \nabla V \cdot \dot{\mathbf{Q}}$ gives

$$\begin{aligned} \dot{V} = \sum_{m=1}^{N} & \begin{bmatrix} \sum_{n=1, n\neq m}^{N} k_f \frac{x_m - x_n - Rdx_{m,n}}{\sqrt{(x_m - x_n - Rdx_{m,n})^2 + (y_m - y_n - Rdy_{m,n})^2}} \\ \dot{x}_m \\ \sum_{n=1, n\neq m}^{N} k_f \frac{y_m - y_n - Rdy_{m,n}}{\sqrt{(x_m - x_n - Rdx_{m,n})^2 + (y_m - y_n - Rdy_{m,n})^2}} \\ \dot{y}_m \end{bmatrix}^T \\ & \cdot \begin{bmatrix} \dot{x}_m \\ -\dot{x}_m - \sum_{n=1, n\neq m}^{N} k_f \frac{x_m - x_n - Rdx_{m,n}}{\sqrt{(x_m - x_n - Rdx_{m,n})^2 + (y_m - y_n - Rdy_{m,n})^2}} \\ \dot{y}_m \\ -\dot{y}_m - \sum_{n=1, n\neq m}^{N} k_f \frac{y_m - y_n - Rdy_{m,n}}{\sqrt{(x_m - x_n - Rdx_{m,n})^2 + (y_m - y_n - Rdy_{m,n})^2}} \end{bmatrix} \\ & = \sum_{m=1}^{N} -(\dot{x}_m^2 + \dot{y}_m^2) \leq 0 \end{aligned} \tag{4.5}$$

Therefore, we can conclude that the system constructed by leading agents under manual mode is stable holding for any N. □

Remark 4.2.1. *When a leading agent in the swarm m is under pure autonomous mode, the control law $\mathbf{u_m}(t-\tau)$ is given by (4.4). We cannot guarantee the system constructed by leading agents is stable because the term $\mathbf{u_m}(t-\tau)$ in the Lyapunov function 4.5 cannot be cancelled out and further $\dot{V} \leq 0$ cannot be confirmed.*

Inspired by the swarm model in [23], we propose the equation of motion for each individual agent i in a swarm as following

$$\dot{\mathbf{Q}}_m^i(t) = -A_m^i \left(\mathbf{Q}_m^i(t) - \mathbf{Q}_m(t)\right) + \Psi \, \mathbf{u}_m^i(t), \tag{4.6}$$

$$\Psi = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{Q}_m^i(t) = [x_m^i(t), \dot{x}_m^i(t), y_m^i(t), \dot{y}_m^i(t)]^T$$

$$\mathbf{u}_m^i(t) = [u_{mx}^i(t), u_{my}^i(t)]^T$$

where A_m^i is a positive coefficient for agent i in the swarm m.

Self-organization refers to the ability of a swarm to form an appropriate division of labor given current task demands or stimuli. When a swarm moves, the agents in the swarm should satisfy the properties of self-organization which have linear attraction and bounded repulsion. In our case, since every agent in a swarm will follow the leader, we

only consider the bounded repulsion property. Hence, the formula of $\mathbf{u}_m^i$ is given by

$$\mathbf{u_m^i} = \begin{bmatrix} \sum_{j=1,j\neq i}^{M} g(x_m^i - x_m^j) \\ \sum_{j=1,j\neq i}^{M} g(y_m^i - y_m^j) \end{bmatrix}$$

where M indicates the number of agents in a cluster and $\mathbf{q}_m^i = (x_m^i, y_m^i)^T$ marks the position of the agent i in the swarm m.

The function $g(\cdot)$ represents the function of mutual repulsion between the individuals. The explicit formulae are given by

$$g(x_m^i - x_m^j) = \begin{cases} b\,(x_m^i - x_m^j - d_{mx})\; exp(-\frac{(x_m^i - x_m^j - d_{mx})^2}{c}), & x_m^i - x_m^j \geq d_{mx} \\ 0, & |x_m^i - x_m^j| < d_{mx} \\ b\,(x_m^i - x_m^j + d_{mx})\; exp(-\frac{(x_m^i - x_m^j + d_{mx})^2}{c}), & x_m^i - x_m^j \leq -d_{mx} \end{cases} \tag{4.7}$$

$$g(y_m^i - y_m^j) = \begin{cases} b\,(y_m^i - y_m^j - d_{my})\; exp(-\frac{(y_m^i - y_m^j - d_{my})^2}{c}), & y_m^i - y_m^j \geq d_{my} \\ 0, & |y_m^i - y_m^j| < d_{my} \\ b\,(y_m^i - y_m^j + d_{my})\; exp(-\frac{(y_m^i - y_m^j + d_{my})^2}{c}), & y_m^i - y_m^j \leq -d_{my} \end{cases} \tag{4.8}$$

where b and c are positive constants and $\mathbf{d}_m = [d_{mx}, d_{my}]$ denotes the minimum safety distance for collision avoidance. Note that $g(z) \leq b^* = b\sqrt{\frac{c}{2}}\; exp(-\frac{1}{2})$. The proof can be found in Appendix A.

Furthermore, we will analyze the cohesion properties for agents in a swarm. We first define the differences between an individual agent i and the leader of the swarm , as $\mathbf{E}^i = \mathbf{Q}_m^i - \mathbf{Q}_m$.

Note that $\dot{\mathbf{E}}^i = \dot{\mathbf{Q}}^i_m - \dot{\mathbf{Q}}_m$, which is

$$\dot{\mathbf{E}}^i = -A^i_m \mathbf{E}^i + \Psi \mathbf{u}^i_m - \Phi \mathbf{Q}_m - \Gamma \mathbf{u}_m$$

Defining the Lyapunov function as $V_i = \frac{1}{2}||\mathbf{E}^i||^2 = \frac{1}{2}\mathbf{E}^{iT}\mathbf{E}^i$. $\dot{V}_i$ is further given by $\dot{V}_i = \dot{\mathbf{E}}^T_i \mathbf{E}_i = -A^i_m (\mathbf{E}^i)^T \mathbf{E}^i + (\Psi \mathbf{u}^i_m)^T \mathbf{E}^i - (\Phi \mathbf{Q}_m)^T \mathbf{E}^i - (\Gamma \mathbf{u}_m)^T \mathbf{E}^i$. Expanding $\dot{V}_i$, we can obtain

$$\begin{aligned}
\dot{V}_i = & -A^i_m ||\mathbf{E}^i||^2 + u^i_{mx}(x^i_m - x_m) + u^i_{my}(y^i_m - y_m) \\
& -\dot{x}_m(x^i_m - x_m) + \dot{x}_m(\dot{x}^i_m - \dot{x}_m) - \dot{y}_m(y^i_m - y_m) + \dot{y}_m(\dot{y}^i_m - \dot{y}_m) \\
& +k_f u_{mx}(\dot{x}^i_m - \dot{x}_m) + k_f u_{my}(\dot{y}^i_m - \dot{y}_m)
\end{aligned}$$

$$\begin{aligned}
\dot{V}_i = & -A^i_m ||x^i_m - x_m||^2 + \left\{ \sum_{j=1, j\neq i}^{M} g(x^i_m - x^j_m) \right\} (x^i_m - x_m) \\
& -\dot{x}_m(x^i_m - x_m) + \dot{x}_m(\dot{x}^i_m - \dot{x}_m) - A^i_m ||\dot{x}^i_m - \dot{x}_m||^2 + k_f u_{mx}(\dot{x}^i_m - \dot{x}_m) \\
& -A^i_m ||y^i_m - y_m||^2 + \left\{ \sum_{j=1, j\neq i}^{M} g(y^i_m - y^j_m) \right\} (y^i_m - y_m) \\
& -\dot{y}_m(y^i_m - y_m) + \dot{y}_m(\dot{y}^i_m - \dot{y}_m) - A^i_m ||\dot{y}^i_m - \dot{y}_m||^2 + k_f u_{my}(\dot{y}^i_m - \dot{y}_m)
\end{aligned}$$

From the swarm systems in Equation (4.2) and the Lyapunov function shown in Equation (4.5), we can see the velocities of the leading agent in two directions will converge to constants. Hence, we set $\dot{x}_m < a_x$ and $\dot{y}_m < a_y$. Besides, we can obtain u_{mx} and u_{my} are bounded within $N-1$ from the formula of the formation control $\mathbf{u_m}$. Combining with $g(z) < b^*$, we have

$$
\begin{aligned}
\dot{V}_i \leq & -A_m^i||x_m^i - x_m||^2 + (M-1)b^* \, ||x_m^i - x_m|| + a_x \, ||x_m^i - x_m|| \\
& -A_m^i||\dot{x}_m^i - \dot{x}_m||^2 + a_x \, ||\dot{x}_m^i - \dot{x}_m|| + k_f(N-1) \, ||\dot{x}_m^i - \dot{x}_m|| \\
& -A_m^i||y_m^i - y_m||^2 + (M-1)b^* \, ||y_m^i - y_m|| + a_y \, ||y_m^i - y_m|| \\
& -A_m^i||\dot{y}_m^i - \dot{y}_m||^2 + a_y \, ||\dot{y}_m^i - \dot{y}_m|| + k_f(N-1) \, ||\dot{y}_m^i - \dot{y}_m|| \\
\dot{V}_i \leq & -A_m^i||x_m^i - x_m|| \left\{ ||x_m^i - x_m|| - \tfrac{(M-1)b^* + a_x}{A_m^i} \right\} \\
& -A_m^i||\dot{x}_m^i - \dot{x}_m|| \left\{ ||\dot{x}_m^i - \dot{x}_m|| - \tfrac{a_x + k_f(N-1)}{A_m^i} \right\} \\
& -A_m^i||y_m^i - y_m|| \left\{ ||y_m^i - y_m|| - \tfrac{(M-1)b^* + a_y}{A_m^i} \right\} \\
& -A_m^i||\dot{y}_m^i - \dot{y}_m|| \left\{ ||\dot{y}_m^i - \dot{y}_m|| - \tfrac{a_y + k_f(N-1)}{A_m^i} \right\}
\end{aligned}
$$

Therefore, we can conclude that as long as $||x_m^i - x_m|| > \frac{(M-1)b^* + a_x}{A_m^i}$, $||y_m^i - y_m|| > \frac{(M-1)b^* + a_y}{A_m^i}$, $||\dot{x}_m^i - \dot{x}_m|| > \frac{a_x + k_f(N-1)}{A_m^i}$, and $||\dot{y}_m^i - \dot{y}_m|| > \frac{a_y + k_f(N-1)}{A_m^i}$, we have $\dot{V}_i < 0$. Hence we will guarantee that as these differences decrease, eventually the bounds for position and velocity will be achieved.

Lemma 4.2.1. *Consider the swarm described by the model in Equation (4.6). Then as $t \to \infty$, we have $\mathbf{q}_m^i \to B_\epsilon(\mathbf{q}_m)$, where*

$$B_\epsilon(\mathbf{q}_m) = \{(x_m^i, y_m^i) : \sqrt{(x_m^i - x_m)^2 + (y_m^i - y_m)^2} \leq \epsilon_m\}$$

$$\epsilon_m = \sqrt{\left\{ \frac{(M-1)b^* + a_x}{\min\{A_m^i\}} \right\}^2 + \left\{ \frac{(M-1)b^* + a_y}{\min\{A_m^i\}} \right\}^2}$$

In addition, we have

$$-\frac{a_x + k_f(N-1)}{A_m^i} \leq (\dot{x}_m^i - \dot{x}_m) \leq \frac{a_x + k_f(N-1)}{A_m^i}$$

$$-\frac{a_y + k_f(N-1)}{A_m^i} \leq (\dot{y}_m^i - \dot{y}_m) \leq \frac{a_y + k_f(N-1)}{A_m^i}$$

These results are important because they prove the cohesiveness of the swarm and provides bounds for the swarm size and the following agents' velocities in this swarm.

Remark 4.2.2. *Notice that no matter whether a leading agent in the swarm m is under the autonomous mode or under the manual mode, both of these two formation control laws are bounded with $N-1$. It indicates that the control modes of the leading agent in a swarm will not affect the cohesiveness of the following agents in this swarm.*

4.3 Collaboration framework

For large-scale human-swarms collaboration systems, how to connect human operators and swarms is a big challenge for us. In this part, we will propose a new measurement to pair human operators and swarms and guarantee successful human-swarms collaborations.

4.3.1 Trust model

In Chapter 2, we introduce two unilateral trust models. Considering the swarm systems we have set up, we extend the human-agent trust model to human-swarm trust model, i.e., $T_{H\to S}(t)$ and $T_{S\to H}(t)$.

Combining the qualitative trust model [55] and time-series trust model proposed in [39], a similar human-to-swarm trust model is proposed as follows:

$$\begin{aligned} T_{m,H\to S}(t) = A_1 T_{m,H\to S}(t-1) + B_1 P_{m,S}(t) - B_2 P_{m,S}(t-1) \\ + D_1 F_{m,S}(t) - D_2 F_{m,S}(t-1), \end{aligned} \tag{4.9}$$

where $P_{m,S}(t)$ denotes the swarm performance and $F_{m,S}(t)$ denotes the swarm fault rate. A_1, B_1, B_2, D_1, and D_2 are constant coefficients whose values depend on the human oper-

ator, the swarm, and the collaborative task. As the above equation shows, the current trust level $T_{m,H\to S}(t)$ is determined by the prior trust level $T_{m,H\to S}(t-1)$, change of swarm performance, and change of swarm fault rate.

Next, we consider the unilateral swarm-to-human trust model $T_{m,S\to H}(t)$. It has been shown that a swarm is composed of a leading agent and its followers. Hence the swarm-to-human trust should consider every agent's trust in human within the swarm. We propose the following swarm-to-human trust model:

$$T_{m,S\to H}(t) = \frac{1}{M}\sum_{i=1}^{M} T_{A_m^i\to H}(t), \tag{4.10}$$

where M represents the number of agents in a swarm and A_m^i denotes the agent i in the swarm m. $T_{A_m^i\to H}(t)$ is defined in Chapter 2.2, which depends on the change of human performance P_H and human fault rate F_H.

4.3.2 Swarm performance model

We propose the swarm performance model based on the cohesiveness of the swarm and swarm systems formation situation. The cohesiveness is measures by the distances between the leader and each following agent in the swarm while the formation situation is measured by the relative distances between the current swarm and its neighbours. The performance model is updated in the following equation

$$P_{m,S}(t) = (1-k_m)\left\{1 - \frac{\sum\limits_{i\in M} e_m^i(t)}{\sum\limits_{i\in M}\epsilon_m}\right\} + k_m\frac{\sum\limits_{n\in N, n\neq m} ARd_{m,n}(t)}{\sum\limits_{n\in N, n\neq m} Rd_{m,n}} \tag{4.11}$$

where $e_m^i(t) = \sqrt{(x_m^i(t)-x_m(t))^2 + (y_m^i(t)-y_m(t))^2}$ is the distance between the position of an individual agent i and the leader in the swarm m at current time instant t,

M represents the number of agents in the swarm m, ϵ_m is the bounded size of the swarm m shown in Lemma (4.2.1), $ARd_{m,n}(t) = \sqrt{(x_m(t) - x_n(t))^2 + (y_m(t) - y_n(t))^2}$ denotes the instantaneous relative distance between the swarm m and its neighbour n, and $Rd_{m,n} = \sqrt{Rdx_{m,n}^2 + Rdy_{m,n}^2}$ represents the desired relative distance between the swarm m and its neighbouring swarm n.

Compared with the performance models shown in Chapter 2.3, the swarm performance model in this part is no more an update formula but an instantaneous equation based on the position of agents at every time instant.

4.3.3 Human attention preference

In the individual-based replicator-mutator (RM) dynamics, each agent in the network decides how to allocate its resources to the tasks based on its own tasking priorities and the local communications with its neighbouring agents as well [64]. In a similar fashion, we model the multi-human systems based on the individual-based replicator-mutator dynamics.

Consider a human network that forms a weighted graph $G_h = (V_h, E_h)$ with a set of vertices V_h of human h_k, $k = 1, 2, \ldots, n$. An edge $e_{jk} \in E_h$ connects h_j with h_k. Here, we consider the aforementioned graph as an agent relational one associated with human h_j with a set of nodes representing agent r_i, $i = 1, 2, \ldots, N$. Let $\mathbf{A}^j = [a_{ik}^j]$, $a_{ii}^j = 1$, denote the $rewards\ matrix$ of G_h, where a_{ik}^j describes the prioritization of agent r_i over r_k evaluated by human h_j under only local interactions with its neighbours, which mainly depends on the initial relative distance. The smaller the distance between r_k and r_i is, the larger a_{ik}^j becomes. Note that $a_{ik}^j < a_{ii}^j = 1$. It indicates only the agent r_i itself has the largest priority. Let $\mathbf{p}^j(t) = (p_1^j(t), p_2^j(t), \ldots, p_N^j(t))^T \in \mathbb{R}^N$ be a vector of allocated attentions of human h_j to agent $\mathbf{r} = (r_1, r_2, \ldots, r_N)$ at time t. That is, p_i^j represents the

percentage of the preferred attentions that human h_j allocates to agent r_i. Let the total attention of human h_j be 1, we have $c^T\mathbf{p}^j(t) = 1$, where $c = (1, 1, \ldots, 1)^T \in \mathbb{R}^N$.

A local-interaction version of the individual-based evolutionary dynamics with social interactions model developed in [31] is developed to model the evolution of $\mathbf{p}^j(t)$.

For each human h_j, define the expected rewards gained by preferring to agent r_i at time t as $f_i^j(t) = \sum_{k=1}^{N} a_{ik}^j(t)p_k^j(t)$, $i \in 1, 2, , N$. The expected rewards vector of agents $\mathbf{R}$ associated with human h_j at time t can be given in matrix form, as $\mathbf{f}^j(t) = \mathbf{A}^j(t)\mathbf{p}^j(t) \in \mathbb{R}$. Let $\mathbf{F}(t) = diag(\mathbf{f}(t))$. Define $\phi^j(t) = \mathbf{f}^j(t)^T\mathbf{p}^j(t)$ as the total rewards gained by human h_j via the attention's allocation of all the agents $\mathbf{R}$. Let $\mathbf{Q}^j(t) = [q_{ik}^j]$ be the mutation matrix associated with human h_j, which is a row stochastic matrix satisfying $\sum_{k=1}^{N} q_{ik}^j(t) = 1$. The component q_{ik}^j is the likelihood that human h_j reallocates its attention from agent r_i to agent r_k ($i \neq k$) at time t.

Now consider the effect of local interactions between human operators within the network. Let us first define two parameters. $f_h^j > 0$ describes the capability differences of a human team. a_{jn}^h describes the strength of the link or connectedness of human h_j to his/her neighbour h_n. Let the parameter ϕ_h^j be the total capability of human h_j's neighborhood h_n and it is defined as

$$\phi_h^j = \sum_{n=1}^{N} f_h^n a_{jn}^h$$

where N is the number of human operators.

Human h_j's individual evolutionary replicator-mutator equation can be written in the matrix form, as

$$\dot{\mathbf{p}}^j(t) = [\mathbf{Q}^j(t)]^T\mathbf{F}^j(t)\mathbf{p}^j(t) - \phi^j(t)\mathbf{p}^j(t) + \sum_{n=1}^{N} f_h^n a_{jn}^h \mathbf{p}^n(t) - \phi_h^j(t)\mathbf{p}^j(t) \tag{4.12}$$

where $\mathbf{p}^n(t)$ represents the neighbouring human h_n's attention allocation.

According to this model, human h_j's attention allocation vector $\mathbf{p}^j(t)$ updates according to its own valuation of different agents r_i (given by the first two terms) as well as the effects of the neighbouring humans allocation $\mathbf{p}^n(t)$ (given by the latter two terms).

4.3.4 Fitness

At first, we define a relation for the node and task information. $fitness(i, j)$ is the fitness value between a task and a node. W_i is the workload of the ith task in the Task Manager. S_j is the CPU speed of the jth node according to the Resource Information Service. Hence, W_i/S_j represents the actual execution time of the ith task with respect to the jth node. E_i is the expected execution time of the ith task. Users can set E_i of tasks. If users do not set the E_i, the Task Manager will set it automatically [65].

This method uses the difference of execution time of the ith task, i.e., $W_i/S_j - E_i$, to estimate how "fit" a node is for a task. The fitness value shows how suitable the task is to the node. The fitness value ranges from 0 to 100000. Larger fitness values will indicate greater suitability between a given task and a node. The fitness value is defined as follows:

$$fitness(i, j) = \frac{100000}{1 + |W_i/S_j - E_i|}$$

Inspired by the above relationship, we define a new 'fitness' between human teams and swarms. This fitness value is proposed as follows:

$$fitness(s_m, h_j) = \frac{1}{1 + |p_m^j(t)P_{m,S}(t) - P_{m,S}^*|} \tag{4.13}$$

where p_m^j represents the allocated attentions of human h_j to the leading agent in the swarm m, $P_{m,S}(t)$ represents the performance of the swarm m, and $P_{m,S}^*$ represents the desired performance of the swarm m. Furthermore, $p_m^j(t)P_{m,S}(t)$ presents the actual performance

with respect to human h_j. The desired performance denotes the performance when a swarm can maintain desired relative distances with its neighbours and each agent in this swarm approaches the leader agent as closely as possible but still keep safety distances with each other to avoid collisions.

This formula analogously uses the difference of performance of the mth swarm, i.e., $p_m^j(t)P_{m,S}(t) - P_{m,S}^*$, to estimate how "fit" a human is for a swarm. Note that $p_m^j(t)$, $P_{m,S}(t)$ and $P_{m,S}^*$ can be guaranteed within the open interval $(0, 1)$ based on Equation. (4.12) and (4.11). We scale the numerator from 10000 to 1 to make the fitness value range from 0 to 1. Larger fitness values will indicate greater suitability between a given swarm and a human.

4.4 Real-time scheduling

Based on the measurement " fitness", we can simplify a large-scale multi-human and swarms collaboration system into several one-human and swarms collaboration systems. Since the human operator only collaborates with leading agents, one-human and swarms collaboration systems can be approximately thought as one-human multi-agent collaboration systems. Hence, we propose a similar scheduling algorithm as shown in Chapter 3.3, called Minimum-Gap-First.

Consider, one gap as $E_{H\to S}(t) = [e_{1,H\to S}(t), \ldots, e_{N,H\to S}(t)]$ where any $e_{m,H\to S}(t) \in E_{H\to S}(t)$ represents the difference between the current human-to-swarm trust level $T_{m,H\to S}(t)$ with respect to the swarm m and its desired trust lower limit $T_{n,l}$, i.e., $e_{m,H\to S}(t+1) = T_{m,H\to S}(t) - T_{n,l}$. The other gap as $E_{S\to H}(t) = [e_{1,S\to H}(t), \ldots, e_{N,S\to H}(t)]$ where any $e_{m,S\to H}(t) \in E_{S\to H}(t)$ represents the difference between the current swarm-to-human trust level $T_{m,S\to H}(t)$ and its desired trust upper limit $T_{n,u}$, i.e., $e_{m,S\to H}(t+1) = T_{n,u} - T_{m,S\to H}(t)$. We define this scheduling as the swarm with the smallest gap gets the highest

priority from a human operator. Algorithms 4.1 and 4.2 discuss the detailed implementation.

Description of Algorithm 4.1: We first update the state vectors [**Q(t), S(t), O(t)**] based on the period $\{L_m\}_{m=1}^N$, collaboration time $\{I_m\}_{m=1}^N$, and control modes. Here we notate the control modes as $\{ctr_m(t)\}_{m=1}^N$. Note that in this scheduling algorithm these three state are updated in the continuous-time case while Equations (3.1), (3.2), and (3.3) update in the discrete-time case. When we switch between these two cases, the previous time step $k-1$ should be replaced by t^- to indicate a time point just before a given time instant. However, the core update procedures are same in these two cases. Furthermore, we update the following system states, i.e., $P_H(t)$, $\{P_{m,S}(t+1)\}_{m=1}^N$, $\{T_{m,H\to S}(t+1), T_{m,S\to H}(t+1), q_m(t+1), s_m(t+1), o_m(t+1)\}_{m=1}^N$. Note that $\{n \in \bar{N}\} = \{n \in N, n \neq m\}$.

Description of Algorithm 4.2: We first calculate the gap vectors for next time instant, $t+1$, based on the difference between the current trust level and the lower limit in the human-to-swarm case and the difference between the upper limit and the current trust level in the swarm-to-human case. As Lines 4 and 5 show, based on the fitness value and the number of human operator $\mathcal{H}$, we categorize each directional gap vector into $\mathcal{H}$ sub-vectors. For example, if the human-swarm collaboration systems include 2 human operators and 4 swarms of agents and the current fitness result is $fitness = [1, 1, 2, 2]$. This result means Swarms $1\&2$ are suitable for Human 1 and Swarms $3\&4$ are suitable for Human 2. Hence, we have $\mathcal{H} = 2$, $N_1 = 2$, and $N_2 = 2$. Furthermore, the trust gaps between Human 1 and Swarms $1\&2$ are grouped while the trust gaps between Human 2 and Swarm $3\&4$ are grouped. Each human will further decide which swarm to collaborate from the corresponding human-swarms group at the current time instant, as Lines 6-15 shows. When the human-to-swarm trust level approaches to the lower limit, an operator is scheduled to interact with a swarm. It indicates that this swarm performance degrades, which

means the formation deviates from the desired situation. Hence, in order to upgrade the swarm performance, the human operator should interact with this swarm and provide more neighbours' information for the swarm. When the agent-to-human trust level approaches to the upper limit, it represents the current swarm becomes " blind" due to lack of neighbours' information and further feels "anxious". Hence, this swarm desires to collaborate with a human operator and gets more information. The swarm with the minimum gap will be chosen to collaborate with.

4.5 Simulation Results

4.5.1 Parameter setup

We simulate the scenario such that two human operators collaborate with four swarms led by four heterogeneous agents: $\{S_1, S_2, S_3, S_4\}$. Each leading agent is followed by 25 agents. The states of each leading agent can be updated by Equation (4.2). The choice of initial states and desired distances between each leading agent and its neighbours are listed in Tables 4.1 and 4.2. The agent performance can be updated by Equation (4.6). The choice of parameters with respect to control inputs shown in Equations (4.7) and (4.8) for each following agent is listed in Table 4.3. Note that the initial condition for each agent in each swarm is set randomly. Each swarm has its initial performance as $[P_{1,S}(0), P_{2,S}(0), P_{3,S}(0), P_{4,S}(0)] = [0.23, 0.27, 0.25, 0.29]$.

The human-to-swarm trust levels follows the dynamic model discussed in Equation (4.9). The constant coefficients in this equation are chosen as $A_1 = 1$, $B_1 = 0.5$, $B_2 = 0.5$, $D_1 = 0.005$, $D_2 = 0.005$, and the fault rates follow the standard normal distribution $N(0, 1)$. The initial trust values between the human operator and four swarms are assumed to be $[T_{H_1-S_1}(0), T_{H_1-S_2}(0), T_{H_1-S_3}(0), T_{H_1-S_4}(0)] = [1.6, 1.6, 1.6, 1.6]$ and

Algorithm 4.1: Co-design of control and scheduling

Data: $r(t)$,$\{ctr_m(t)\}_{m=1}^{N}$,$\{\mathbf{Q}_m(t), \mathbf{u}_m(t)\}_{m=1}^{N}$, $\{L_m, I_m\}_{m=1}^{N}$, $fitness(t)$, $P_H(t)$, $\{P_{m,S}(t)\}_{m=1}^{N}$, $\{\{p_m^j\}_{m=1}^{N}\}_{j=1}^{\mathcal{H}}$, $\{\{\mathbf{Q_m}^i, \mathbf{u}_m^i\}_{i=1}^{M}\}_{m=1}^{N}$, $\{\{e_m^i(t)\}_{i=1}^{M}\}_{m=1}^{N}$, $\{\sum_{n\in\bar{N}} ARd_{m,n}(t)\}_{m=1}^{N}$, $\{T_{m,H\to S}(t), T_{m,S\to H}(t), q_m(t), s_m(t), o_m(t)\}_{m=1}^{N}$

Result: $r(t+1)$, $\{ctr_m(t+1)\}_{m=1}^{N}$, $fitness(t+1)$, $P_H(t+1)$, $\{P_{m,S}(t+1)\}_{m=1}^{N}$, $\{T_{m,H\to S}(t+1), T_{m,S\to H}(t+1)\}_{m=1}^{N}$, $\{q_m(t+1), s_m(t+1), o_m(t+1)\}_{m=1}^{N}$

for *each* S_m **do**

- $q_m(t+1) \xleftarrow{\text{Eq.(3.1)}} \{q_m(t), L_m\}$;
- $s_m(t+1) \xleftarrow{\text{Eq.(3.2)}} \{s_m(t), ctr_m(t), q_m(t), I_m\}$;
- $o_m(t+1) \xleftarrow{\text{Eq.(3.3)}} \{s_m(t), ctr_m(t), q_m(t)\}$;
- $\sum_{n\in\bar{N}} ARd_{m,n}(t+1) \xleftarrow{\text{Eq.(4.2)}} \{\mathbf{Q}_m(t), \mathbf{u}_m(t), o_m(t)\}$;
- $e_m^i(t+1) \xleftarrow{\text{Eq.(4.6)}} \{\mathbf{Q}_m^i(t), \mathbf{u}_m^i(t)\}$;
- $P_{m,S}(t+1) \xleftarrow{\text{Eq.(4.11)}} \{e_m^i(t+1), \sum_{n\in\bar{N}} ARd_{m,n}(t+1)\}$;
- $r(t+1) \xleftarrow{\text{Eq.(2.5)}} \{r(t), ctr_m(t)\}$;
- $P_H(t+1) \xleftarrow{\text{Eq.(2.4)}} r(t)$;
- $fitness(t+1) \xleftarrow{\text{Eq.(4.13)}} \{P_{m,S}(t), p_m^j\}$;
- $T_{m,H-S}(t+1) \xleftarrow{\text{Eq.(4.9)}} \{T_{m,H-S}(t), P_{m,S}(t+1), P_{m,S}(t), F_S(t+1), F_S(t)\}$;
- $T_{m,S-H}(t+1) \xleftarrow{\text{Eq.(4.10)}} \{T_{m,S-H}(t), P_H(t+1), P_H(t), F_H(t+1), F_H(t)\}$;

Algorithm 4.2;

return $r(t+1)$, $\{ctr_m(t+1)\}_{m=1}^{N}$, $P_H(t+1)$, $\{P_{m,S}(t+1)\}_{m=1}^{N}$, $\{T_{m,H\to S}(t+1), T_{m,S\to H}(t+1), q_m(t+1), s_m(t+1), o_m(t+1)\}_{m=1}^{N}$;

Algorithm 4.2: Minimum-Gap-First Scheduling

Data: $\{ctr_m(t)\}_{m=1}^{N}$, $\{P_{m,S}(t)\}_{m=1}^{N}$, $fitness(t)$,
$\{T_{m,H\to S}(t), T_{m,S\to H}(t), s_m(t+1)\}_{m=1}^{N}$

Result: $\{m_j\}_{j=1}^{\mathcal{H}}$

for *each* S_m **do**
    /* G is a set of agents with the non-zero remaining collaboration time*/
    $e_{m,H\to S}(t+1) = T_{m,H\to S}(t) - T_{n,l}$;
    $e_{m,S\to H}(t+1) = T_{n,u} - T_{m,S\to H}(t)$;
$\{\{e_{m_j,H\to S}\}_{m_j=1}^{N_j}(t+1)\}_{j=1}^{\mathcal{H}} \leftarrow \{fitness(t)\}$;
$\{\{e_{m_j,S\to H}\}_{m_j=1}^{N_j}(t+1)\}_{j=1}^{\mathcal{H}} \leftarrow \{fitness(t)\}$;
for *each* H_j **do**
    $G_j = [\,]$;
    if $s_{m_j}(t+1) > 0$ **then**
        $G_j = [\,G,\, S_{mj}\,]$;
    if G_j *is not empty* **then**
        $m_j = \min_{S_{m_j}\in G}([\{e_{m_j,H\to S}\}_{m_j=1}^{N_j}(t+1), \{e_{m_j,S\to H}\}_{m_j=1}^{N_j}(t+1)])$;
        if $m == m_j$ **then**
            $ctr_m(t+1) = 1$;
        else if $m \neq mj$ **then**
            $ctr_m(t+1) = 0$;
return $\{m_j\}_{j=1}^{\mathcal{H}}$;

Table 4.1: Initial Condition in Leading Agent Systems

	$x(0)$ [m]	$y(0)$ [m]	$v_x(0)$ [m/s]	$v_y(0)$ [m/s]
S_1	100	105	5	8
S_2	101	106	6	7
S_3	108	103	7	6
S_4	102	107	8	5

Table 4.2: Desired Relative Distances

Rd_x/Rd_y	S_1 [m]	S_2 [m]	S_3 [m]	S_4 [m]
S_1	0/0	100/0	100/-50	0/-50
S_2	-100/0	0/0	0/-50	-100/-50
S_3	-100/50	0/50	0/0	-100/0
S_4	0/50	100/50	100/0	0/0

$[T_{H_2-S_1}(0), T_{H_2-S_2}(0), T_{H_2-S_3}(0), T_{H_2-S_4}(0)] = [1.6, 1.6, 1.6, 1.6]$. Note that the swarm-to-human trust levels are combination of all agent-to-human trust levels. Hence, we set the same constant coefficients and initial trust values as those used in Chapter 3.4. In this simulation, we choose the trust regions with the lower bounds $T_{1,l} = 1.2, T_{2,l} = 1.2, T_{3,l} = 1.2, T_{4,l} = 1.2$, the upper bounds $T_{1,u} = 3.2, T_{2,u} = 3.2, T_{3,u} = 3.2, T_{4,u} = 3.2$.

In addition, the parameters in the periodic strategy are shown in Table 4.4

4.5.2 Results and discussions

Fig. 4.1 shows the shape formation in the coordinate plane of swarm systems under pure manual mode within time interval $[1, 100]$. We know each leading agent under manual mode can obtain whole neighbouring information during this time interval. Hence, from Fig. 4.1(a), we can get the desired formation based on the defined relative distances. Fig. 4.1(b) depicts the layout of Swarm 1 and shows that each following agent can avoid collision from other agents. This indicates that Equation (4.7) takes effect on mutual repul-

Table 4.3: Parameters in Following Agent Systems

	a	b	c
$\{Agent_1^i\}_{i=1}^{25}$	1	2	30
$\{Agent_2^i\}_{i=1}^{25}$	1	2	40
$\{Agent_3^i\}_{i=1}^{25}$	1	2	50
$\{Agent_4^i\}_{i=1}^{25}$	1	2	60

Table 4.4: Parameters in Periodic Strategy

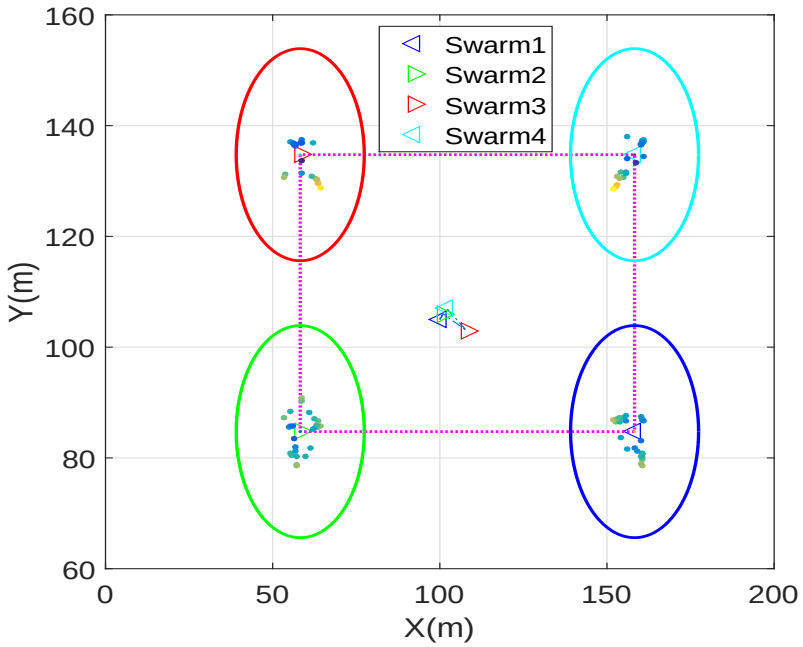
	S_1	S_2	S_3	S_4
H_1	$[I_1, L_1] = [15, 50]s$	$[I_1, L_1] = [18, 50]s$	$[I_1, L_1] = [19, 50]s$	$[I_1, L_1] = [16, 50]s$
H_2	$[I_2, L_2] = [16, 50]s$	$[I_2, L_2] = [17, 50]s$	$[I_2, L_2] = [18, 50]s$	$[I_2, L_2] = [15, 50]s$

sion between the individuals in one swarm.

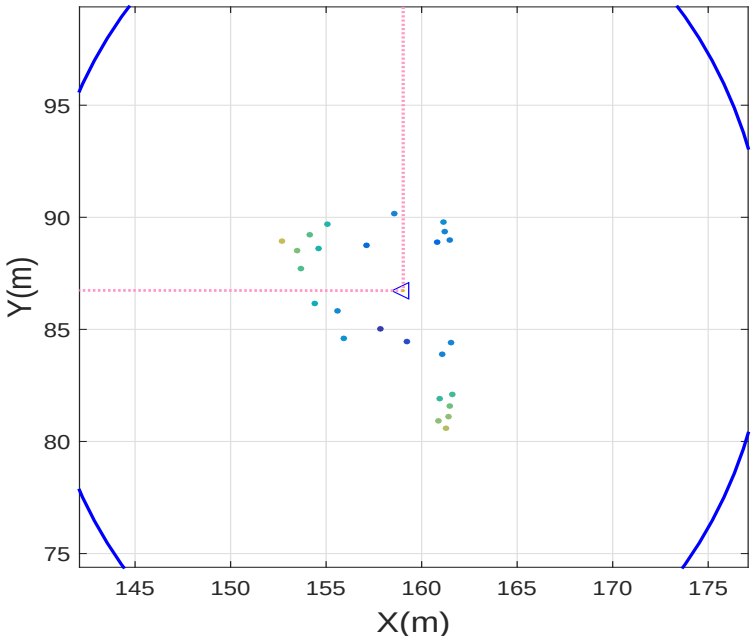
Fig. 4.2 shows the shape formation in the coordinate plane of swarm systems under pure autonomous mode within time interval $[1, 100]$. As we discuss, each leading agent under autonomous mode can only obtain neighbouring information at the initial time. Hence, from Fig. 4.2(a), we can see the shape formation deviates from the desired shape. Fig. 4.2(b) depicts the layout of Swarm 1 as well. When the leading agent lost the direction, the following agents also lost their directions.

In addition, we observe from Figures. 4.1 and 4.2 that the circle areas defined in Lemma 4.2.1 can cover all agents in each swarm. Note that we use same parameters in these two modes to obtain the same bounded area for each swarm. Here, we have $[b, c] = [2, 30]$ and $a_x = a_y = 1$. The radius of the area is $\epsilon_m = \sqrt{\left\{\frac{(M-1)b^*+a_x}{\min\{A_m^i\}}\right\}^2 + \left\{\frac{(M-1)b^*+a_y}{\min\{A_m^i\}}\right\}^2}$. Here, $\min\{A_m^i\}$ is assigned a random.

Fig. 4.3 shows the shape formation in the coordinate plane of swarm systems scheduled by the minimum-gap-first algorithm. Furthermore, we compare the shape formations under these three cases shown in Fig. 4.4. The area surrounded by blue line represents the

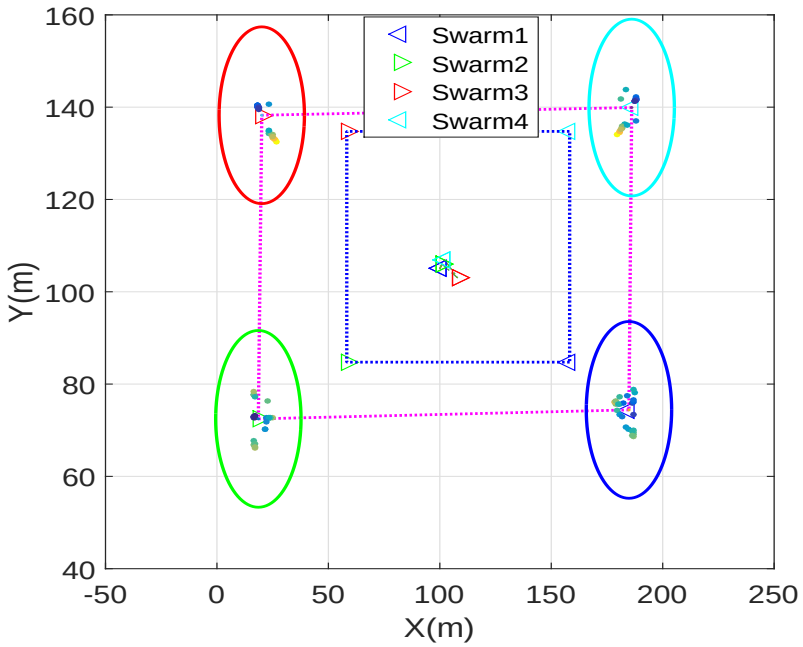


(a) global graph

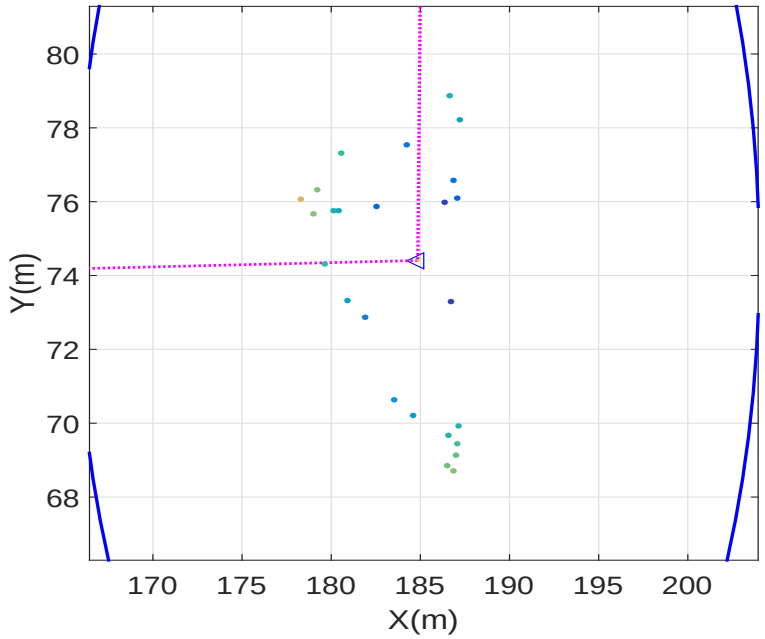


(b) local graph for Swarm 1

Figure 4.1: Formation under pure manual mode.



(a) global graph



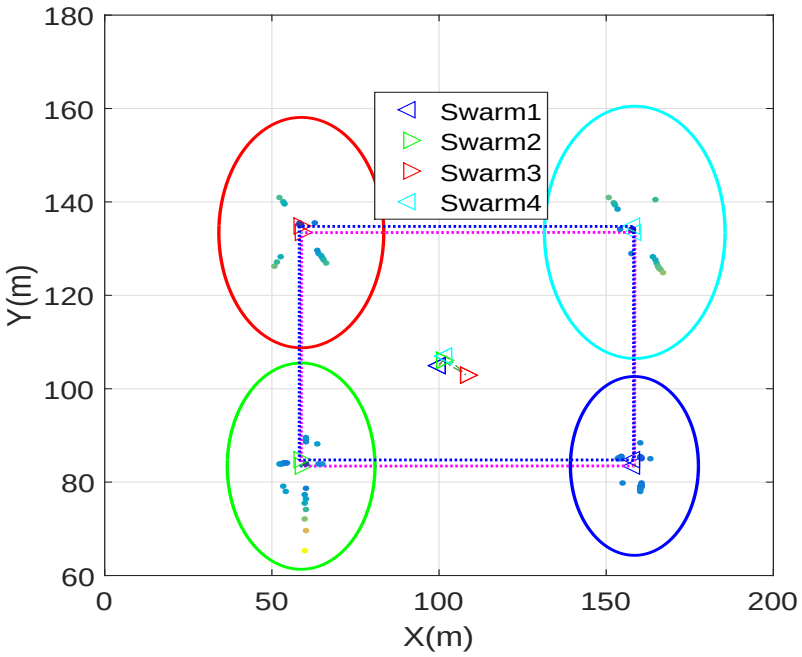
(b) local graph for Swarm 1

Figure 4.2: Formation under pure autonomous mode.

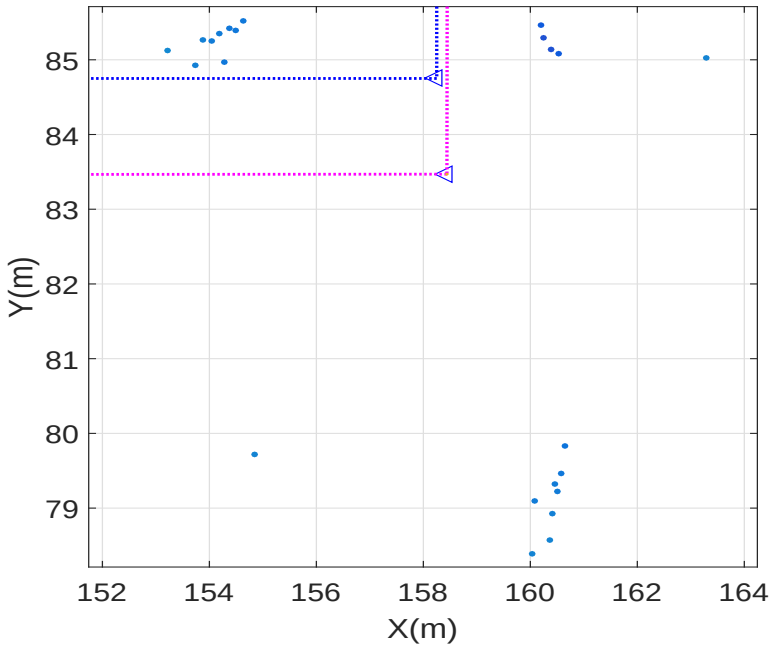
desired shape under pure manual model while the area surrounded by green line denotes the worst case under autonomous mode. Besides, the area enclosed by pink line is the formation scheduled by the minimum-gap-first algorithm. We can clearly see the area in this case approaches to the best formation shape.

Fig. 4.5 depicts the human-to-swarm trust levels. The red line shows 'fitness' results. Note that we scale the 'fitness' value to 0.5 and 1. When the red line goes to 0.5, it indicates Human 1 is collaborating with a swarm. When the red line reaches 1, it represents Human 2 is collaborating with a swarm. In addition, we can observe as time propagates, human-to-swarm trust levels will converge to steady. This observation coincides with the human-to-swarm-trust model in that the trust model depends on the states of swarm systems which will get stable based on Theorem 4.2.1 and Lemma 4.2.1. Fig. 4.6 shows the dynamic of leading agents in each swarm. Fig. 4.7 depicts the swarm-to-human trust levels. Here swarm-to-human trust is no more trust between a swarm and a fixed human operator. Based on 'fitness' results, swarm will produce corresponding trust levels towards different human operator.

Finally, we discuss the relation between human workload and the acceptable formation shape. From Fig. 4.4, we can observe that the proposed scheduling algorithm can guarantee the formation shape under this scheduling algorithm is located between the shape under pure autonomous mode and manual mode. In this part, we also utilize the trust levels to decide the control modes. However, the ultimate goal is no more to guarantee the trust levels within the trust bounds as Chapter 3 shows, but to determine the acceptable formation. From the minimum-gap-scheduling algorithm, we know the control mode depends on the trust gaps between trust levels and trust bounds. Furthermore, the control mode decides the formation shape. That is to say, we need to balance two trust levels to have an acceptable shape. Since the workload can be reflected by the delay time, we directly analyze the statistics of delay time.



(a) global graph



(b) local graph for Swarm 1

Figure 4.3: Formation scheduled by Minimum-Gap-First algorithm.

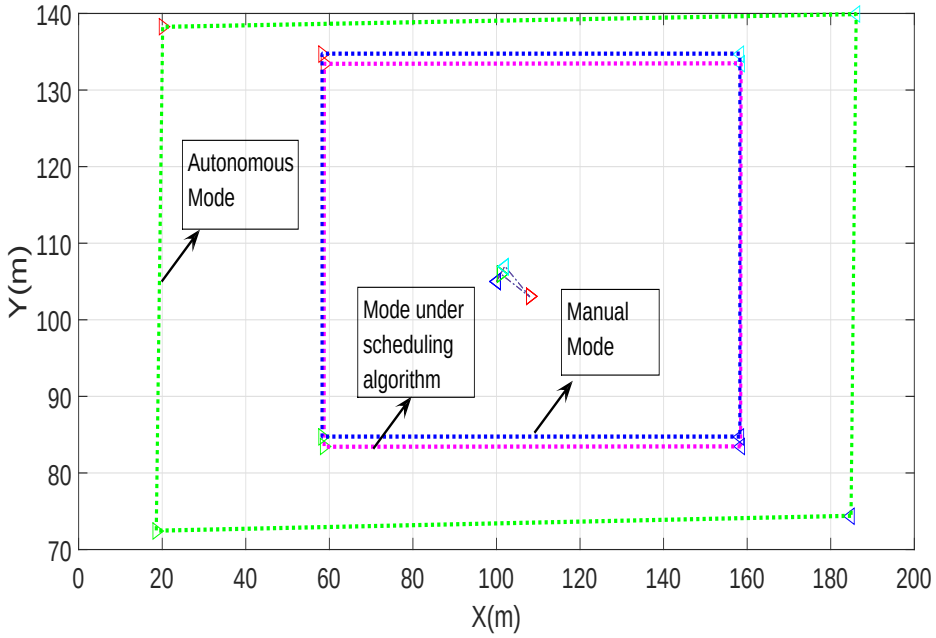
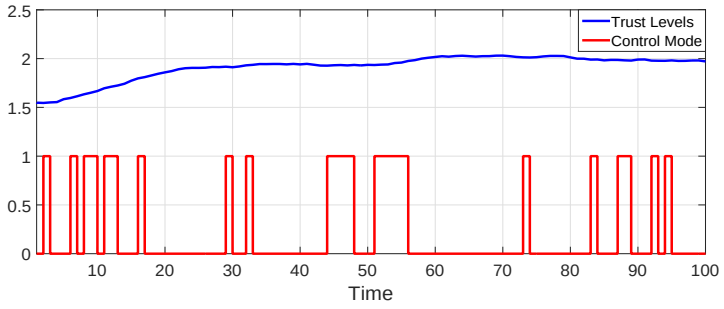
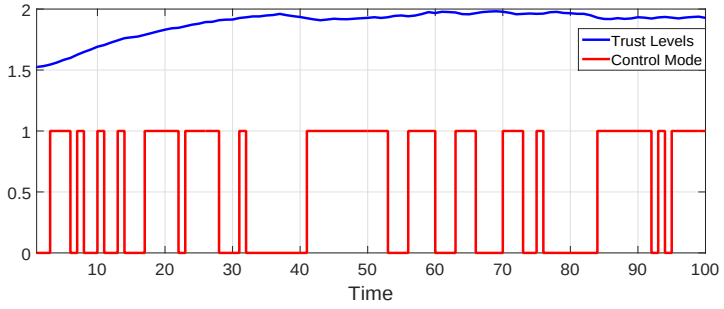


Figure 4.4: Comparison

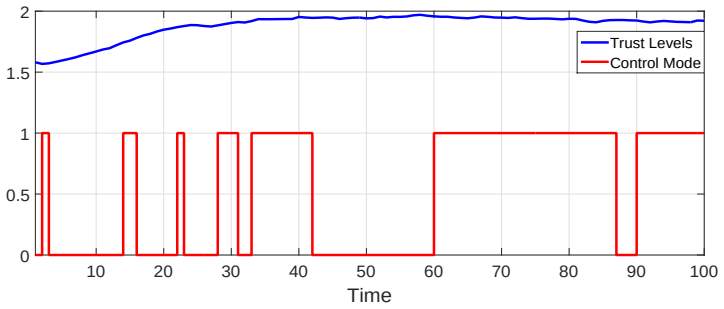
Table 4.5 and 4.6 show the formation shapes under different trust bounds. Note that we run the simulation under each trust bound 20 times and take the average. From these two tables, we can observe that the deviation from the desired shape is small if the standard variance of delay time, i.e., Std(τ), is relatively small. The standard variance of delay time reflects the distribution of human workload to each swarm. The smaller the standard variance of delay time is, the more equal the distribution of human workload to each swarm is. This indicates if each swarm can obtain the relatively more information from human operators compared with the case under the pure autonomous mode, the formation constructed by swarm systems is perfect. Besides, we can observe as the bound increases, the standard variance of delay time has the increment trend. Combining with the range of trust levels shown Figures. 4.5 and 4.7, we can say when human operator makes a decision, if he or she cares both swarm-to-human trust levels and human-to-swarm trust levels, that is, the human operator cares both his/her own opinions and swarm's willing, the decision



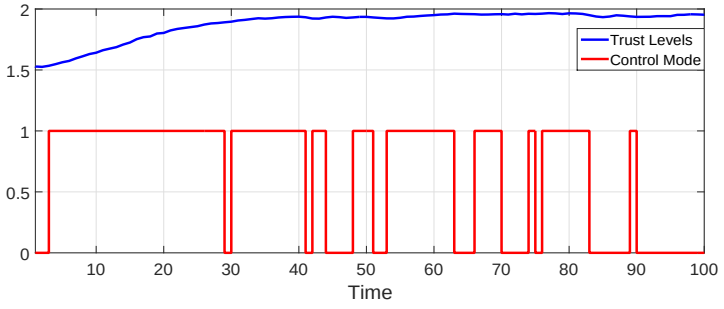
(a) Human-to-Swarm 1 Trust



(b) Human-to-Swarm 2 Trust

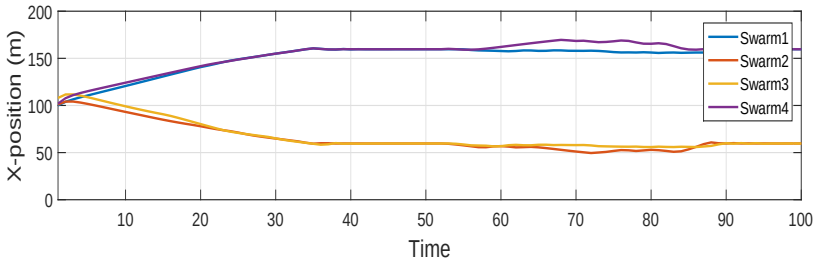


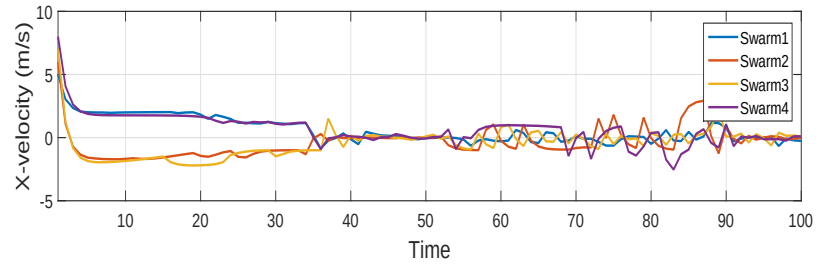
(c) Human-to-Swarm 3 Trust



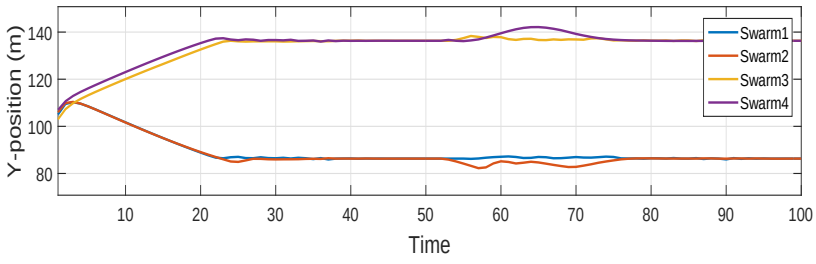
(d) Human-to-Swarm 4 Trust

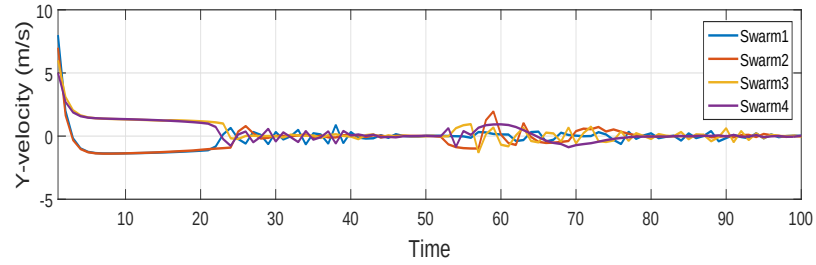
Figure 4.5: Human-to-swarm trust





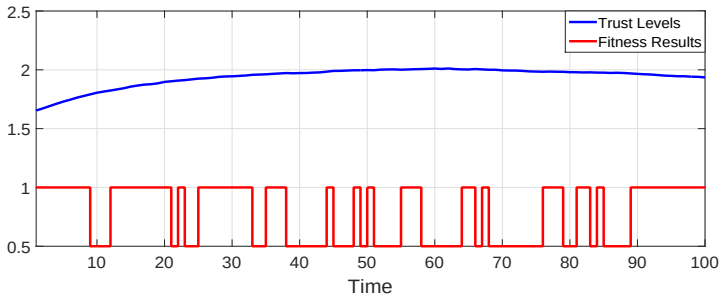
(a) X-direction



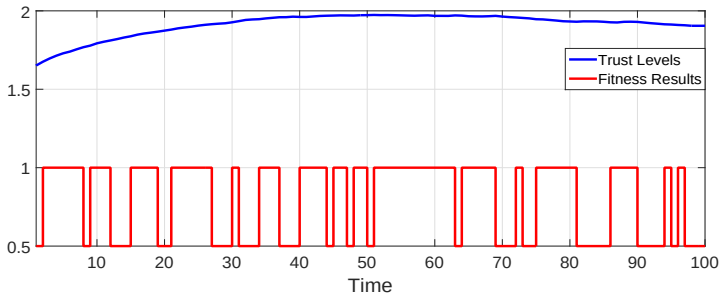


(b) Y-direction

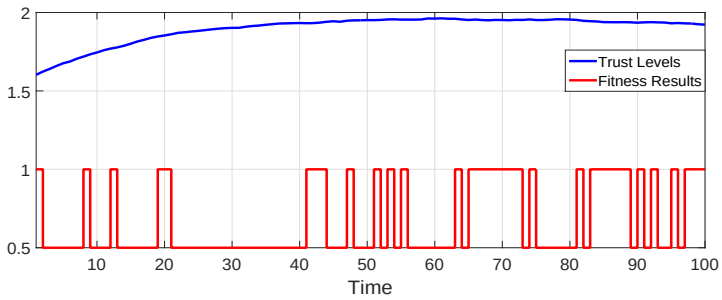
Figure 4.6: Leading Agents Dynamic.



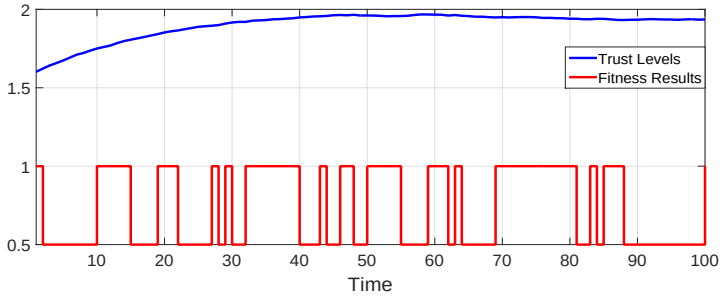
(a) Swarm 1-to-Human Trust



(b) Swarm 2-to-Human Trust



(c) Swarm 3-to-Human Trust



(d) Swarm 4-to-Human Trust

Figure 4.7: Swarm-to-human trust

Table 4.5: Relation between upper trust bounds and formation shape

Upper bound	τ_1	τ_2	τ_3	τ_4	Mean(τ)	Std(τ)	Scheduled shape	Desired shape	Worst shape
6.2	55	61	80	4	50	32.5	303.5	300	460
5.2	56	81	61	2	50	33.5	304.5	300	460
4.2	52	10	97	41	50	36	308.5	300	460
3.2	71	28	52	49	50	17.6	300	300	460

Table 4.6: Relation between lower trust bounds and formation shape

Lower bound	τ_1	τ_2	τ_3	τ_4	Mean(τ)	Std(τ)	Scheduled shape	Desired shape	Worst shape
2.2	50	66	80	4	50	33.1	303.5	300	460
1.2	81	18	49	52	50	25.7	304.5	300	460
0.2	52	20	87	41	50	28.1	308.5	300	460

for control mode is more reasonable. However, even if the distribution of human workload is not equal, the formation shape deviates slightly from the perfect formation shape and is still acceptable. Hence, we can conclude that the scheduling algorithm can guarantee successful human-swarm collaboration.

Chapter 5

Conclusions and Discussion

In this paper, we propose bilateral trust models for the collaborations between human and (semi)autonomous agents. Specifically, we offer a human-to-agent trust model based on results in human factors and a novel agent-to-human trust model based on human-human collaboration. We develop a dynamic timing model to describe the status of different state variables, and use it to derive necessary and sufficient conditions for schedulability test to save computational efforts.

Furthermore, we extend the collaboration between one human and multiple agents into the collaboration between multi-human network and swarm-based agents' network. A corresponding swarm system is set up. Cooperative controls are incorporated into the swarm systems to enable several swarms to simultaneously reach navigational goals and avoid collisions between each agent in the swarm. To measure the collaboration between human systems and swarm systems, we propose a novel measurement, called 'fitness'.

Our simulation results show that the proposed algorithm can also be applied to this large-scaled collaboration system and guarantees effective real-time scheduling of the human multi-agent collaboration system while ensuring a proper level of human-agent mutual trust.

Appendices

Appendix A The maximum value of $g(\cdot)$ function

According to the explicit formulae of the function $g(\cdot)$ shown in Equations. (4.7)and (4.8), we can summarize the $g(\cdot)$ function in one formula, as

$$g(z) = \begin{cases} b\,(z-d)\,exp(-\frac{(z-d)^2}{c}), & z \geq d \\ 0, & |z| < d \\ b\,(z+d)\,exp(-\frac{(z+d)^2}{c}), & z \leq -d \end{cases}$$

where b, c, and d are positive constants. Note that $g(z)$ is a continuous function in the entire value range.

I. When $z \geq d$, we have $\frac{dg}{dz} = b\,exp(-\frac{(z-d)^2}{c})[1 - \frac{2(z-d)^2}{c}]$.

Let $\frac{dg}{dz} = 0$. We obtain

$$\begin{aligned} b\,exp(-\tfrac{(z-d)^2}{c})[1 - \tfrac{2(z-d)^2}{c}] &= 0 \\ \Rightarrow (z-d)^2 &= \tfrac{c}{2} \\ \Rightarrow z &= \sqrt{\tfrac{c}{2}} + d \end{aligned}$$

When $(z-d)^2 < \frac{c}{2} \Rightarrow d < z < \sqrt{\frac{c}{2}} + d$, we have $\frac{dg}{dz} > 0$.

When $(z-d)^2 > \frac{c}{2} \Rightarrow z > \sqrt{\frac{c}{2}} + d$, we have $\frac{dg}{dz} < 0$.

When $(z-d)^2 = 0 \Rightarrow z = d$, we have $g(d) = 0$.

Hence, we can conclude that when $z \geq d$, the maximum value of function $g(z)$ is $b\,\sqrt{\frac{c}{2}}\,exp(-\frac{1}{2})$ at $z = \sqrt{\frac{c}{2}} + d$.

II. When $|z| < d$, we have $g(z) = 0$.

III. When $|z| \leq -d$, we have $\frac{dg}{dz} = b\,exp(-\frac{(z+d)^2}{c})[1 - \frac{2(z+d)^2}{c}]$.

Let $\frac{dg}{dz} = 0$. We obtain

$$b\ exp(-\tfrac{(z+d)^2}{c})[1 - \tfrac{2(z+d)^2}{c}] = 0$$
$$\Rightarrow (z+d)^2 = \tfrac{c}{2}$$
$$\Rightarrow z = -\sqrt{\tfrac{c}{2}} - d$$

When $(z+d)^2 < \frac{c}{2} \Rightarrow -\sqrt{\frac{c}{2}} - d < z < -d$, we have $\frac{dg}{dz} > 0$.

When $(z+d)^2 > \frac{c}{2} \Rightarrow z < -\sqrt{\frac{c}{2}} - d$, we have $\frac{dg}{dz} < 0$.

When $(z+d)^2 = 0 \Rightarrow z = -d$, we have $g(-d) = 0$.

Hence, we can conclude that when $z \leq -d$, the maximum value of function $g(z)$ is 0 at $z = -d$.

Therefore, we can conclude that $g(z) \leq b\ \sqrt{\frac{c}{2}}\ exp(-\frac{1}{2})$ for $z \in \mathbb{R}^1$.

For the $z \in \mathbb{R}^1$ case with $b = 5$, $c = 40$, and $d = 1$, this function is shown in Fig. 1.

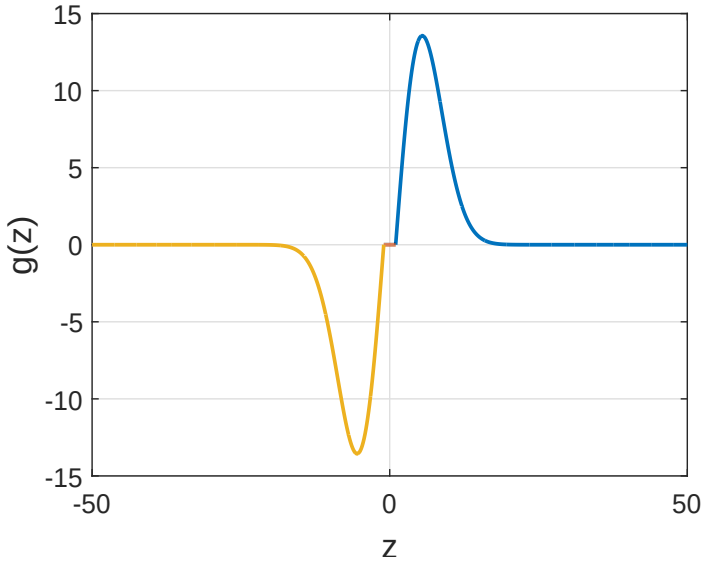


Figure 1: g(z) Function

Bibliography

[1] Natalia M Alexandrov, John E Dennis Jr, Robert Michael Lewis, and Virginia Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural Optimization*, 15(1):16–23, 1998.

[2] Laura Barnes, Wendy Alvis, MaryAnne Fields, Kimon Valavanis, and Wilfrido Moreno. Swarm formation control with potential fields formed by bivariate normal functions. In *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*, pages 1–7. IEEE, 2006.

[3] Sepideh Bazazi, Pawel Romanczuk, Sian Thomas, Lutz Schimansky-Geier, Joseph J Hale, Gabriel A Miller, Gregory A Sword, Stephen J Simpson, and Iain D Couzin. Nutritional state and collective motion: from individuals to mass migration. *Proceedings of the Royal Society of London B: Biological Sciences*, 278(1704):356–363, 2011.

[4] By Randal W Beard, Timothy W McLain, Derek B Nelson, Derek Kingston, and David Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *Proceedings of the IEEE*, 94(7):1306–1324, 2006.

[5] Gerardo Beni. From swarm intelligence to swarm robotics. In *Swarm Robotics*, pages 1–9. Springer, 2005.

[6] JW M Bertrand and HPG Van Ooijen. Workload based order release and productivity: a missing link. *Production Planning & Control*, 13(7):665–678, 2002.

[7] Naval Studies Board. *Autonomous vehicles in support of naval operations*. National Academies Press, 2005.

[8] Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 263(1376):1565–1569, 1996.

[9] Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60(4):753–807, 1998.

[10] CM Breder. Equations descriptive of fish schools and other animal aggregations. *Ecology*, pages 361–370, 1954.

[11] Stefano Carpin and Lynne E Parker. Cooperative leader following in a distributed multi-robot system. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2994–3001. IEEE, 2002.

[12] Nicolas Champagnat, Régis Ferrière, and Sylvie Méléard. Unifying evolutionary dynamics: from individual stochastic processes to macroscopic models. *Theoretical population biology*, 69(3):297–321, 2006.

[13] Jimming Cheng, Winston Cheng, and Radhika Nagpal. Robust and self-repairing formation control for swarms of mobile agents. In *AAAI*, volume 5, pages 59–64, 2005.

[14] Gilles Coppin and François Legras. Autonomy spectrum and performance perception issues in swarm supervisory control. *Proceedings of the IEEE*, 100(3):590–603, 2012.

[15] Jacob W Crandall, Michael A Goodrich, Dan R Olsen Jr, and Curtis W Nielsen. Validating human-robot interaction schemes in multitasking environments. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(4):438–449, 2005.

[16] Missy L Cummings. Human supervisory control of swarming networks. In *2nd Annual Swarming: Autonomous Intelligent Networked Systems Conference*, pages 1–9, 2004.

[17] Munjal Desai and Holly Adviser-Yanco. Modeling trust to improve human-robot interaction. 2012.

[18] Aris L Dimeas and Nikos D Hatziargyriou. Operation of a multiagent system for microgrid control. *Power Systems, IEEE Transactions on*, 20(3):1447–1455, 2005.

[19] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. A taxonomy for swarm robots. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 1, pages 441–447. IEEE, 1993.

[20] Xiaocong Fan, Sooyoung Oh, Michael McNeese, John Yen, Haydee Cuevas, Laura Strater, and Mica R Endsley. The influence of agent reliability on trust in human-agent collaboration. In *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction*, page 7. ACM, 2008.

[21] AMOS Freedy, Ewart DeVisser, Gershon Weltman, and Nicole Coeyman. Measurement of trust in human-robot collaboration. In *Collaborative Technologies and Systems, 2007. CTS 2007. International Symposium on*, pages 106–114. IEEE, 2007.

[22] Veysel Gazi, Barıs Fidan, Y Sinan Hanay, and M Ilter Köksal. Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques. *Turk J Elec Engin*, 15(2):149–168, 2007.

[23] Veysel Gazi and Kevin M Passino. Stability analysis of social foraging swarms. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(1):539–557, 2004.

[24] William Bryan Gooding. *Specially structured formulations and solution methods for optimisation problems important to process scheduling*. PhD thesis, Purdue University, 1994.

[25] Bill Goodwine and Panos Antsaklis. Multiagent coordination exploiting system symmetries. In *American Control Conference (ACC), 2010*, pages 830–835. IEEE, 2010.

[26] Daniel Grünbaum and Akira Okubo. Modelling social animal aggregations. In *Frontiers in mathematical biology*, pages 296–325. Springer, 1994.

[27] Daniel Lihui Gu, Guangyu Pei, Henry Ly, Mario Gerla, Beichuan Zhang, and Xiaoyan Hong. Uav aided intelligent routing for ad-hoc wireless network in single-area theater. In *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE*, volume 3, pages 1220–1225. IEEE, 2000.

[28] Peter A Hancock, Deborah R Billings, Kristin E Schaefer, Jessie YC Chen, Ewart J De Visser, and Raja Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(5):517–527, 2011.

[29] Josef Hofbauer and Karl Sigmund. Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 40(4):479–519, 2003.

[30] Yiguang Hong, Guanrong Chen, and Linda Bushnell. Distributed observers design for leader-following control of multi-agent networks. *Automatica*, 44(3):846–850, 2008.

[31] Islam Hussein et al. An individual-based evolutionary dynamics model for networked social behaviors. In *American Control Conference, 2009. ACC'09.*, pages 5789–5796. IEEE, 2009.

[32] Makoto Itoh and Kenji Tanaka. Mathematical modeling of trust in automation: Trust, distrust, and mistrust. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 44, pages 9–12. SAGE Publications, 2000.

[33] Sanza T Kazadi. *Swarm engineering*. PhD thesis, California Institute of Technology, 2000.

[34] Evelyn F Keller and Lee A Segel. Model for chemotaxis. *Journal of Theoretical Biology*, 30(2):225–234, 1971.

[35] Dong Hun Kim, Hua Wang, and Seiichi Shin. Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines. *Journal of Intelligent and Robotic Systems*, 45(4):369–394, 2006.

[36] Andreas Kolling, Steven Nunnally, and Michael Lewis. Towards human control of robot swarms. In *Proceedings of the seventh annual ACM/IEEE international conference on human-robot interaction*, pages 89–96. ACM, 2012.

[37] Natalia L Komarova. Replicator–mutator equation, universality property and population dynamics of learning. *Journal of Theoretical Biology*, 230(2):227–239, 2004.

[38] Vignesh Kumar and Ferat Sahin. Cognitive maps in swarm robots for the mine detection application. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, pages 3364–3369. IEEE, 2003.

[39] John Lee and Neville Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35(10):1243–1270, 1992.

[40] Kristina Lerman, Alcherio Martinoli, and Aram Galstyan. A review of probabilistic macroscopic models for swarm robotic systems. In *Swarm robotics*, pages 143–152. Springer, 2005.

[41] Stephan Lewandowsky, Michael Mundy, and Gerard Tan. The dynamics of trust: comparing humans to automation. *Journal of Experimental Psychology: Applied*, 6(2):104, 2000.

[42] Erez Lieberman, Christoph Hauert, and Martin A Nowak. Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316, 2005.

[43] Chung Laung Liu and James W Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.

[44] Yang Liu and Kevin M Passino. Swarm intelligence: Literature overview. *Department of Electrical Engineering, the Ohio State University*, 2000.

[45] Michael Mendl. Performing under pressure: stress and cognitive function. *Applied Animal Behaviour Science*, 65(3):221–244, 1999.

[46] Neville Moray, Toshiyuki Inagaki, and Makoto Itoh. Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44, 2000.

[47] Daniel Morgan, Soon-Jo Chung, and Fred Y Hadaegh. Model predictive control of swarms of spacecraft using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 37(6):1725–1740, 2014.

[48] Chase C Murray and Woojin Park. Incorporating human factor considerations in unmanned aerial vehicle routing. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(4):860–874, 2013.

[49] Akira Okubo. Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. *Advances in biophysics*, 22:1–94, 1986.

[50] Reza Olfati-Saber and Richard M Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, pages 346–352. Citeseer, 2002.

[51] Constantinos C Pantelides. Unified frameworks for optimal process planning and scheduling. In *Proceedings on the second conference on foundations of computer aided operations*, pages 253–274. Cache Publications New York, 1994.

[52] Pawel Romanczuk, Iain D Couzin, and Lutz Schimansky-Geier. Collective motion due to individual escape and pursuit response. *Physical Review Letters*, 102(1):010602, 2009.

[53] Pawel Romanczuk and Lutz Schimansky-Geier. Swarming and pattern formation due to selective attraction and repulsion. *Interface focus*, 2(6):746–756, 2012.

[54] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Swarm robotics*, pages 10–20. Springer, 2005.

[55] Tracy Sanders, Kristin E Oleson, DR Billings, Jessie YC Chen, and PA Hancock. A model of human-robot trust theoretical model development. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 55, pages 1432–1436. SAGE Publications, 2011.

[56] Ketan Savla and Emilio Frazzoli. A dynamical queue approach to intelligent task management for human operators. *Proceedings of the IEEE*, 100(3):672–686, 2012.

[57] Hiroki Sayama. Decentralized control and interactive design methods for large-scale heterogeneous self-organizing swarms. In *Advances in Artificial Life*, pages 675–684. Springer, 2007.

[58] Lui Sha, Tarek Abdelzaher, Karl-Erik Årzén, Anton Cervin, Theodore Baker, Alan Burns, Giorgio Buttazzo, Marco Caccamo, John Lehoczky, and Aloysius K Mok. Real time scheduling theory: A historical perspective. *Real-time systems*, 28(2-3):101–155, 2004.

[59] Zhenwu Shi. *Non-worst-case Response Time Analysis for Real-time Systems Design*. PhD thesis, Georgia Institute of Technology, April 2014.

[60] Zhenwu Shi and Fumin Zhang. Predicting time-delays under real-time scheduling for linear model predictive control. In *Proceedings of International Conference on Computing, Networking and Communications, Workshops Cyber Physical System*, pages 205–209, 2013.

[61] Zhenwu Shi and Fumin Zhang. Model predictive control under timing constraints induced by controller area networks. *arXiv preprint arXiv:1503.02300*, 2015.

[62] David JT Sumpter. The principles of collective animal behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1465):5–22, 2006.

[63] David JT Sumpter. *Collective animal behavior*. Princeton University Press, 2010.

[64] Qiuchen Wang. Modeling and design strategy of online advertising ecosystem. Master's thesis, Clemson University, August 2014.

[65] Sheng-De Wang, I-Tar Hsu, and Zheng-Yi Huang. Dynamic scheduling methods for computational grid environments. In *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, volume 1, pages 22–28. IEEE, 2005.

[66] Xiaotian Wang, Zhenwu Shi, Fumin Zhang, and Yue Wang. Dynamic real-time scheduling for human-agent collaboration systems based on mutual trust. *Cyber-Physical Systems*, (ahead-of-print):1–15, 2015.

[67] Xiaotian Wang, Zhenwu Shi, Fumin Zhang, and Wang Yue. Mutual trust based scheduling for (semi)autonomous multi-agent systems. In *Proceedings of American Control Conference*, 2015. `http://people.clemson.edu/~yue6/papers/conference/WaShZhWa-ACC-15.pdf`.

[68] Yue Wang, Islam I Hussein, and Adriana Hera. Evolutionary task assignment in distributed multi-agent networks with local interactions. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4749–4754. IEEE, 2010.

[69] Yue Wang, Zhenwu Shi, Chuanfeng Wang, and Fumin Zhang. Human-robot mutual trust in (semi) autonomous underwater robots. In *Cooperative Robots and Sensor Networks 2014*, pages 115–137. Springer, 2014.

[70] Kevin Warburton and John Lazarus. Tendency-distance models of social cohesion in animal groups. *Journal of Theoretical Biology*, 150(4):473–488, 1991.

[71] KL Yee and Nilay Shah. Improving the efficiency of discrete time scheduling formulation. *Computers & chemical engineering*, 22:S403–S410, 1998.

[72] Robert M Yerkes and John D Dodson. The relation of strength of stimulus to rapidity of habit-formation. *Journal of comparative neurology and psychology*, 18(5):459–482, 1908.

[73] Wenwu Yu, Guanrong Chen, Ming Cao, Jinhu Lü, and Hai-Tao Zhang. Swarming behaviors in multi-agent systems with nonlinear dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(4):043118, 2013.

[74] Fumin Zhang, Zhenwu Shi, and Shayok Mukhopadhyay. Robustness analysis for battery-supported cyber-physical systems. *ACM Transactions on Embedded Computing Systems*, 12(3):Artical 69(1–27), 2013.