

Mutual Trust Based Scheduling for (Semi)autonomous Multi-Agent Systems

Xiaotian Wang*, Zhenwu Shi[†], Fumin Zhang[†], and Yue Wang*

Abstract—Mutual trust is a key factor in human-human collaboration. Inspired by this social interaction, we propose to analyze human-agent mutual trust in the collaboration of human and (semi)autonomous multi-agent systems. Human-agent mutual trust should be bidirectional and determines the human’s acceptance and use of autonomous agents as well as agents’ willingness to take human’s command. It is especially important when a human collaborates with multiple agents concurrently. In this paper, we propose time-series human-agent mutual trust models based on well known 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. We demonstrate the effectiveness of the proposed scheduling algorithm using Matlab simulations. It shows that the proposed algorithm guarantees the effective real-time scheduling of the human multi-agent collaboration system while ensuring a proper level of mutual trust.

I. INTRODUCTION

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 [1]. In this pursuit, building mutual trust between human operators and (semi)autonomous agents is of particular importance since the 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. 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). The recent meta-analysis [2] studies the factors involved in trust for human-robot interaction (HRI), which includes robot-related, human-related, and environmental-related factors. In [3], Freedy et al. study the critical performance attributes of trust in HRI and develop a collaborative performance model. In [4], Yanco and Desai investigate the HRI involved in remote robot teleoperation (RRT) and summarize five categories of trust models. Apart from the above qualitative works, the dynamic aspect of trust and different operating

conditions of an automated system are studied in [5]. Itoh and Tanaka propose a mathematical model of trust based on the expectation of humans from 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 human-agent collaboration system, trust between human and agents should be bilateral, including both human-to-agent trust and agent-to-human trust. Here, 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 respond to human commands [6]. Inspired by the time-series trust model proposed in [6] and the theoretical framework for trust in [2], in this paper, we propose time-series dynamic models for the mutual trust between the human operator and (semi)autonomous agents.

A human operator needs to share attention to each agent in a multi-agent system. This case is analogous to the classic research topic about task allocation in real-time scheduling. In this case, it is necessary to develop a dynamic scheduling algorithm so that human can allocate his/her attention to each agent in real time. In the literature [7], there are three categories of classical scheduling algorithms: fixed priority scheduling, dynamic priority scheduling, and mixed priority scheduling. This paper [7] also discusses three corresponding conditions for the schedulability test of these algorithms. However, these results are not applicable in our dynamic systems where the ultimate goal is not just to meet deadlines but to avoid both “over-trust” and “under-trust”. Therefore, in [1], Wang et al. introduce a novel scheduling algorithm called “highest-trust-first” scheduling, which can guarantee effective real-time scheduling of manual and autonomous control of agents. In [8]–[10], Shi et al. propose a dynamic timing model and necessary and sufficient conditions for schedulability test of dynamic systems. In this paper, based on the bilateral trust dynamic models and extending the “highest-trust-first” scheduling, we propose a rigorous schedulability test algorithm using the dynamic timing model to avoid both “over-trust” and “under-trust”.

The remainder of the paper is organized as follows. Section II introduces the mutual trust models and the definition of schedulability. The dynamic timing model is introduced in Section III. Section IV gives the schedulability test algorithm based on the mutual trust and dynamic timing models. We provide a set of Matlab simulation results in Section V and conclude the paper in Section VI.

*Department of Mechanical Engineering, Clemson University, Fluor Daniel Building, Clemson, SC 29634, E-mail: {xiaotiw, yue6}@clemson.edu; [†]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, E-mail: {zwshi, fumin}@gatech.edu.

II. PROBLEM SETUP

A. Trust Model

In this section, 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-A} , and agent-to-human trust model T_{A-H} , respectively.

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 [11]. It is further pointed out in the meta-analysis [2] that agent performance is strongly related to trust level, environmental factors are moderately associated with trust, and human performance has the least relationship with trust evolution. Therefore, combining the qualitative trust model [11] and time-series trust model proposed in [6], we have the following human-to-agent trust model:

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

where k denotes the discrete time step, $P_A(k)$ denotes the agent performance, $F_A(k)$ denotes 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 $T_{H-A}(k)$ is determined by the prior trust level $T_{H-A}(k-1)$, change of agent performance, and change of agent fault rate.

Next, we consider the agent-to-human trust model $T_{A-H}(k)$. Similar to the human-to-agent trust, $T_{A-H}(k)$ will depend on the change of performance P_H and fault rate F_H of the human partner. Analogous to Eq. (1), we propose the following agent-to-human trust model:

$$T_{A-H}(k) = A_2 T_{A-H}(k-1) + C_1 P_H(k) - C_2 P_H(k-1) + E_1 F_H(k) - E_2 F_H(k-1), \quad (2)$$

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

B. Trust and Automation Use

In this paper, we assume that both $T_{H-A}(k)$ and $T_{A-H}(k)$ must fall within two desired regions [12] denoted as, $T_{H-A} \in [T_{H-A,l}, T_{H-A,u}]$ and $T_{A-H} \in [T_{A-H,l}, T_{A-H,u}]$ for $k \geq 0$. For the human-to-agent trust level $T_{H-A}(k)$, if its value exceeds the corresponding upper limit, i.e., $T_{H-A}(k) \geq T_{H-A,u}$, it indicates the human operator has too much trust on the autonomous operation of the agents. To address this issue, the human operator should start to control the agents manually. On the other hand, if the human-to-agent trust level $T_{H-A}(k)$ goes below the lower limit, i.e., $T_{H-A}(k) \leq T_{H-A,l}$, it means the human operator has too little trust on the autonomous operation of the agents and put too much manual control. In this case, the human operator

should allow the autonomous operation rather than control them manually.

Similarly, for the agent-to-human trust level $T_{A-H}(k)$, if $T_{A-H}(k) \geq T_{A-H,u}$, this indicates that the agent relies too much on the human operator's manual control. In this case, the human operator should stop the manual control of agents but let them operate autonomously. On the other hand, if the agent-to-human trust level $T_{A-H}(k) \leq T_{A-H,l}$, the agent has too little trust on the human operator's manual control but operate autonomously by themselves. In this case, the human operator should start manual control of agents to increase their trust in the human operators.

C. Schedulability Definition

Consider the case where a human operator collaborates with N (semi)autonomous agents and denote an agent as $\{A_1, \dots, A_N\}$. Based on the above explanation of trust relation, 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 2.1: 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 regions, 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 $[k_a, k_b]$.

In order to save computational time, we develop necessary and sufficient conditions to check the schedulability of such a given human multi-agent system within a sub-interval, shown in Section III.

D. Agent Performance Model

We consider two modes when a human collaborates with agents: the autonomous and the manual mode, with different performance models given by the following two different equations [13]

$$P_{n,A}(k) = \begin{cases} (1 - k_{n,A})P_{n,A}(k-1) + k_{n,A}P_{n,A,\min}, & \text{(autonomous mode)} \\ (1 - k_{n,H})P_{n,A}(k-1) + k_{n,H}P_{n,A,\max}, & \text{(manual mode)} \end{cases} \quad (3)$$

where $P_{n,A,\max}, P_{n,A,\min} \in [0, 1]$ 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 (3) guarantees that $P_{n,A}$ of each agent A_n is bounded between $[P_{n,A,\min}, P_{n,A,\max}]$, given that their initial performance falls within $[P_{n,A,\min}, P_{n,A,\max}]$.

E. Human Performance Model

The Yerkes-Dodson law [14] 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 [15]

$$P_H(k) = (P_{H,\max} - P_{H,\min}) \left(\frac{r(k)}{\beta} \right)^\beta \left(\frac{1 - r(k)}{1 - \beta} \right)^{1-\beta} + P_{H,\min}, \quad (4)$$

where $\beta \in (0, 1)$ represents the difficulty of a task for a human (a smaller value of β represents a more difficult task [16]), $r(k)$ represents the utilization of which the definition and model can be found in [1], [17], and $P_{H,\max}$ and $P_{H,\min}$ represent the maximum and minimum human performance value, respectively. Note that the human performance model (4) guarantees that P_H is bounded between $[P_{H,\min}, P_{H,\max}]$.

The utilization ratio $r(k)$ is determined by the control modes of all agents. Note that the utilization $r(k)$ increases in the manual mode ($u_n(k) = 1$) and decreases in the autonomous mode ($u_n(k) = 0$). Since the human operator can only collaborate with one agent at a time, the utilization ratio $r(k)$ is bounded between 0 and 1.

III. DYNAMIC TIMING MODEL

As we have introduced in Section I, there are three categories of scheduling algorithms in classical real-time scheduling and each of them has conditions for schedulability test. However, these algorithms have some constraints which make them not directly applicable in our problem. First, the fixed priority scheduling algorithms, such as rate-monotonic scheduling (R.M.S.) [7], can only be used in systems where the scheduling parameters are constant over each period, which is not the case for our dynamic systems. Second, the dynamic priority scheduling algorithms, such as earliest deadline first (E.D.F.) [7], are also not applicable here, as their goal is to guarantee the tasks to finish before deadline while our goal is to maintain the trust level within the desired region. Because of the new requirements in our systems, more conditions need to be considered, such as the human performance and the desired trust region, which increases the complexity of the schedulability test. Finally, the mixed scheduling algorithms that combine the R.M.S. and deadline driven dynamic scheduling algorithms still do not address the above mentioned limitations and hence do not apply to our problem either.

In this paper, we develop a new scheduling algorithm that can allocate human attention to each agent so that the mutual trust level of each human-agent pair falls within the desired trust region. 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 $\{A_1, \dots, A_n\}$ corresponds to executing a set of tasks $\Gamma = \{\tau_1, \dots, \tau_N\}$ on a single core processor. Based on the above analysis, we apply the highest-trust-first scheduling method in the multi-agent systems [7], [8]. As we discussed in the previous section, we cannot use Definition 2.1 to test if a given human multi-agent system

is schedulable. Instead, we develop necessary and sufficient conditions to achieve this goal by adopting the following dynamic timing model.

To describe the current status of agents at any time step k and build up the dynamic timing model, we define a state vector $Z(k) = [\mathbf{Q}(k), \mathbf{S}(k), \mathbf{O}(k)]$ as follows.

Definition 3.1: The dynamic arrival time is defined as $\mathbf{Q}(k) = [q_1(k), \dots, q_N(k)]$, where $q_n(k)$ denotes how long after time step, k , the next collaboration request from A_n will launch.

Definition 3.2: The residue time is defined as $\mathbf{S}(k) = [s_1(k), \dots, s_N(k)]$, where $s_n(k)$ denotes the remaining collaboration time required after time step, k , by agent A_n .

Definition 3.3: The dynamic response time is defined as $\mathbf{O}(k) = [o_1(k), \dots, o_N(k)]$. Here, we need to define $o_n(k)$ for 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 completion time.

Based on Definitions 3.1-3.3, we propose the evolution model of $Z(k)$ within any sub-interval $[k_w, k_{w+1}]$. The evolution of the dynamic arrival time $\mathbf{Q}(k)$ is given as follows:

$$\begin{cases} q_n(k_w) = q_n(k_w - 1) - 1, & \text{if } q_n(k_w - 1) > 1 \\ q_n(k_w) = L_n, & \text{if } q_n(k_w - 1) = 1 \\ q_n(k_w + \epsilon_w) = q_n(k_w) - \epsilon_w, & \epsilon_w \in [1, P_w - 1] \end{cases}, \quad (5)$$

where ϵ_w is an integer representing time step. The initial condition during the whole time interval is set as $q_n(k_a) = L_n$. P_w is the length of each sub-interval [8].

The evolution of the dynamic residue time $\mathbf{S}(k)$ is given as follows:

$$\begin{cases} s_n(k_w) = \max\{0, s_n(k_w - 1) - 1\}, & \text{if } q_n(k_w - 1) > 1 \text{ \& } P_w \neq 1 \\ s_n(k_w) = \max\{0, s_n(k_w - 1) - \max\{0, 1 - \sum_{HP} u_i(k_w - 1)\}\}, & \text{if } q_n(k_w - 1) > 1 \text{ \& } P_w = 1 \\ s_n(k_w) = I_n(k_w), & \text{if } q_n(k_w - 1) = 1 \\ s_n(k_w + \epsilon_w) = \max\{0, s_n(k_w) - \max\{0, \epsilon_w - \sum_{HP} \sum_{k=k_w}^{k_w+\epsilon_w} u_i(k)\}\}, & \text{otherwise} \end{cases}, \quad (6)$$

where the term HP represents a set containing all the agents which have higher priority than A_n . 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 evolution of the dynamic response time $\mathbf{O}(k)$ is given

as follows:

$$\left\{ \begin{array}{l} o_n(k_w) = o_n(k_w - 1), \\ \quad \text{if } q_n(k_w - 1) > 1 \ \& \ s_n(k_w - 1) = 0 \\ o_n(k_w) = o_n(k_w - 1) + 1, \\ \quad \text{if } q_n(k_w - 1) > 1 \ \& \ s_n(k_w - 1) > 0 \\ o_n(k_w) = 0, \quad \text{if } q_n(k_w - 1) = 1 \\ o_n(k_w + \epsilon_w) = o_n(k_w), \quad \text{if } s_n(k_w) = 0 \\ o_n(k_w + \epsilon_w) = o_n(k_w) + \min\{s_n(k_w) + \\ \quad \sum_{HP} \sum_{k=k_w}^{k_w+\epsilon_w} u_i(k), \epsilon_w\}, \quad \text{if } s_n(k_w) \neq 0 \end{array} \right. \quad (7)$$

The initial condition is $o_n(k_a) = 0$.

Based on the evolutions of state variables and Definition 2.1, the schedulability of the human-agent collaboration system can now be formally defined as follows.

Definition 3.4: 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 A_n within a sub-interval $[k_w, k_{w+1}]$.

Theorem 3.1: (Necessary and Sufficient Condition) An agent A_n is schedulable within $[k_w, k_{w+1}]$ if and only if it satisfies one of the following conditions:

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

Note that the above necessary and sufficient conditions are conservative. The following theorem gives a less conservative sufficient condition.

Theorem 3.2: (Sufficient Condition) If an agent A_n is schedulable, it will satisfy one of the following conditions within $[k_w, k_{w+1}]$:

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

The difference between Theorem 3.1 and 3.2 is the condition, $s_n(k_{w+1} - 1) = 1$. If we use the necessary and sufficient conditions in Theorem 3.1, some time steps will lead to unschedulability. Hence, Theorem 3.2 as a necessary condition can be viewed as a supplement to Theorem 3.1. Readers can find the proof and more detailed discussions about the conservative property of these given conditions in [8].

IV. 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 3.1. This algorithm is composed of 3 parts, shown below. It iteratively checks the schedulability of each agent, A_n , as follows.

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 result during the time interval $[k_a, k_b]$. 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. We define $e_{n,A-H}(k+1) = T_{n,A-H,u} - T_{n,A-H}(k)$ in a similar fashion. The agent with the minimum error, i.e. closest to the ‘‘over-trust’’ situation, will be chosen to collaborate with. Furthermore, we update the collaboration time of each agent $I_n(k)$ by calculating the maximum and minimum mutual trust level. When the trust becomes too high, we increase the amount of collaboration time, as shown in Line 31-36. Note that the value of $I_n(k)$ cannot go beyond L_n . On the other hand, when the mutual trust becomes too low, we decrease the amount of collaboration time, as shown in Line 37-42. Note that the value of $I_n(k)$ cannot be smaller than zero. Note that $\delta_1, \delta_2, \delta_3, \delta_4 > 0$ are arbitrarily small values guaranteeing that $I_n(k)$ is adjusted before the mutual trust level goes beyond upper and lower limit.

V. SIMULATION RESULTS

A. Simulation

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 Eq. (3) and the choice of parameters for each agent is listed in Table I. 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 Eq. (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 level between the human operator and each agent A_n follows the dynamic models discussed in Eqs. (1) and (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 follow the standard normal distribution $N(0, 1)$. The initial mutual trust value 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 of the human operator is to make sure that the trust level $T_n(k) = [T_{n,H-A}(k); T_{n,A-H}(k)]$ with each agent A_n 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 III, we choose the initial param-

Algorithm 1: Main Algorithm

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

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

```
1 for each  $A_n \in \Gamma$  do
2    $DS_n = []$ ;
3    $ds_n = 1$ ;
4    $k_w = k_a$ ;
5   Algorithm2;
6 while  $k_w < k_b$  do
7   for each  $A_n$  do
8     /* Update Mutual Trust,  $P_{n,A}, P_{n,H}$  (1)-(4) */
9     /* Update State Variables (5)-(7)*/
10     $e_{n,H-A}(k+1) = T_{n,H-A,u} - T_{n,H-A}(k)$ ;
11     $e_{n,A-H}(k+1) = T_{n,A-H} - T_{n,A-H,l}(k)$ ;
12     $G = []$ ;
13    if  $s_n(k_w + \epsilon_w) > 0$  then
14       $G = [G, A_n]$ ;
15    if  $G$  is not empty then
16       $i = \min_{A_n \in G} \{[e_{n,H-A}(k+1), e_{n,A-H}(k+1)]\}$ ;
17      if  $n == i$  then
18         $u_n(k+1) = 1$ ;
19      else if  $n \neq i$  then
20         $u_n(k+1) = 0$ ;
21    for each  $A_n$  do
22       $T_{n,H-A,max} = \max_{k-L_n \leq \tau \leq k} T_{n,H-A}(\tau)$ ;
23       $T_{n,A-H,max} = \max_{k-L_n \leq \tau \leq k} T_{n,A-H}(\tau)$ ;
24       $T_{n,H-A,min} = \min_{k-L_n \leq \tau \leq k} T_{n,H-A}(\tau)$ ;
25       $T_{n,A-H,min} = \min_{k-L_n \leq \tau \leq k} T_{n,A-H}(\tau)$ ;
26      if  $T_{n,H-A,max} > T_{n,H-A,u} - \delta_1$  and
27       $T_{n,H-A,min} > T_{n,H-A,d}$  then
28         $\phi_1 = 1$ ;
29      if  $T_{n,A-H,min} < T_{n,A-H,l} + \delta_2$  and
30       $T_{n,A-H,max} < T_{n,A-H,d}$  then
31         $\phi_2 = 1$ ;
32      if  $\phi_1 \cup \phi_2 = 1$  then
33         $I_n(k) = \min\{I_n(k) + 1, L_n\}$ ;
34      if  $T_{n,H-A,min} < T_{n,H-A,l} + \delta_3$  and
35       $T_{n,H-A,max} < T_{n,H-A,d}$  then
36         $\phi_3 = 1$ ;
37      if  $T_{n,A-H,max} > T_{n,A-H,u} - \delta_4$  and
38       $T_{n,A-H,min} > T_{n,A-H,d}$  then
39         $\phi_4 = 1$ ;
40      if  $\phi_3 \cup \phi_4 = 1$  then
41         $I_n(k) = \max\{I_n(k) - 1, 0\}$ ;
42      Algorithm 3;
43    $w=w+1$ ;
44 return  $\{DS_n\}_{n=1}^N$ ;
```

Algorithm 2: Determination of the fixed priority window

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

Result: P_w

```
1 while  $k_w < k_b$  do
2   for each  $A_n$  do
3     if  $q_n(k_w - 1) == 1$  then
4        $q_n(k_w) = L_n$ ;
5     else
6        $q_n(k_w) = q_n(k_w - 1) - 1$ ;
7    $P_w = \min\{q_1(k_w), \dots, q_N(k_w), k_b - k_w\}$ ;
8    $k_{w+1} = k_w + P_w$ ;
9 return  $P_w$ ;
```

Algorithm 3: Schedulability test

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

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

```
1 if  $q_n(k_{w+1} - 1) == 1$  then
2   if  $o_n(k_{w+1} - 1) < L_n - 1$  then
3      $ds_n(k) = 1$ ;
4   else if  $o_n(k_{w+1} - 1) = L_n - 1$  and
5    $s_n(k_{w+1} - 1) = 0$  then
6      $ds_n(k) = 1$ ;
7   else
8      $ds_n(k) = 0$ ;
9 else
10   $ds_n(k) = ds_n(k - 1)$ ;
11  $DS_n = \{DS_n, ds_n(k)\}$ ;
12 return  $\{DS_n\}_{n=1}^N$ ;
```

eters in the periodic strategy as

$$[I_1, L_1] = [2, 10]s \quad [I_2, L_2] = [3, 10]s \quad [I_3, L_3] = [4, 10]s \quad (8)$$

where each pair $[I_n, L_n]$ denotes that the human operator must collaborate with 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 1. If the trust level $T_{H-A,n}(k)$ and $T_{A-H,n}(k)$ are too close to the lower bound, i.e. the “under-trust” situation, the human operator will collaborate less with this agent and thus the value of $I_n(k)$ will decrease. On the other hand, if the mutual trust level is too close to the upper bound, the value of $I_n(k)$ will increase.

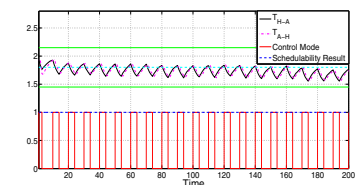
B. Results and Discussions

Fig. 1 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.

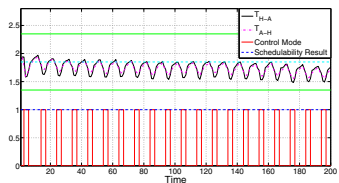
TABLE I: Coefficients in Agent Performance Model

	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

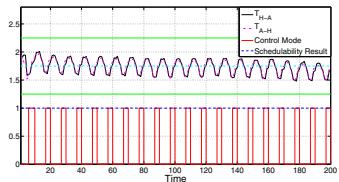
The black dashed lines represent the ideal expert level. The blue lines represent the human-to-agent trust level and the pink dotted lines represent the agent-to-human trust level. The red lines represent the (de)activation of manual control. From the figures, we can observe that the human operator can interact 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 brown dotted lines represent the schedulability result, where “1” means that the system is schedulable and “0” means NOT schedulable. The mutual trust level in each human-agent pair is consistently bounded within desired regions and the human-agent systems are 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.



(a) $T_1(k)$ between Human and Agent 1



(b) $T_2(k)$ between Human and Agent 2



(c) $T_3(k)$ between Human and Agent 3

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

VI. CONCLUSION

In this paper, we propose a human-to-agent trust model based on well-known qualitative results in human factors and a novel agent-to-human trust model based on human-human collaboration. We also 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. Furthermore, we develop a schedulability test algorithm using the dynamic timing model to avoid both over- and under-trust. The simulation results show that our scheduling algorithm can guarantee the mutual trust level in the desired trust regions.

REFERENCES

- [1] Y. Wang, Z. Shi, C. Wang, and F. Zhang, “Human-robot mutual trust in (semi) autonomous underwater robots,” in *Cooperative Robots and Sensor Networks 2014*, pp. 115–137, Springer, 2014.
- [2] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. Chen, E. J. De Visser, and R. Parasuraman, “A meta-analysis of factors affecting trust in human-robot interaction,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 53, no. 5, pp. 517–527, 2011.
- [3] E. Freedy, E. DeVisser, G. Weltman, and N. Coeyman, “Measurement of trust in human-robot collaboration,” in *Collaborative Technologies and Systems, 2007. CTS 2007. International Symposium on*, pp. 106–114, IEEE, 2007.
- [4] M. Desai and H. Adviser-Yanco, “Modeling trust to improve human-robot interaction,” 2012.
- [5] M. Itoh and K. Tanaka, “Mathematical modeling of trust in automation: Trust, distrust, and mistrust,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 44, pp. 9–12, SAGE Publications, 2000.
- [6] J. Lee and N. Moray, “Trust, control strategies and allocation of function in human-machine systems,” *Ergonomics*, vol. 35, no. 10, pp. 1243–1270, 1992.
- [7] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.
- [8] Z. Shi, *Non-worst-case Response Time Analysis for Real-time Systems Design*. PhD thesis, Georgia Institute of Technology, April 2014.
- [9] F. Zhang, Z. Shi, and S. Mukhopadhyay, “Robustness analysis for battery-supported cyber-physical systems,” *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 3, pp. Article 69(1–27), 2013.
- [10] Z. Shi and F. 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*, pp. 205–209, 2013.
- [11] T. Sanders, K. E. Oleson, D. Billings, J. Y. Chen, and P. Hancock, “A model of human-robot trust theoretical model development,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 55, pp. 1432–1436, SAGE Publications, 2011.
- [12] N. M. Alexandrov, J. E. Dennis Jr, R. M. Lewis, and V. Torczon, “A trust-region framework for managing the use of approximation models in optimization,” *Structural Optimization*, vol. 15, no. 1, pp. 16–23, 1998.
- [13] J. W. Crandall, M. A. Goodrich, D. R. Olsen Jr, and C. W. Nielsen, “Validating human-robot interaction schemes in multitasking environments,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 4, pp. 438–449, 2005.
- [14] R. M. Yerkes and J. D. Dodson, “The relation of strength of stimulus to rapidity of habit-formation,” *Journal of comparative neurology and psychology*, vol. 18, no. 5, pp. 459–482, 1908.
- [15] J. M. Bertrand and H. Van Ooijen, “Workload based order release and productivity: a missing link,” *Production Planning & Control*, vol. 13, no. 7, pp. 665–678, 2002.
- [16] M. Mendl, “Performing under pressure: stress and cognitive function,” *Applied Animal Behaviour Science*, vol. 65, no. 3, pp. 221–244, 1999.
- [17] K. Savla and E. Frazzoli, “A dynamical queue approach to intelligent task management for human operators,” *Proceedings of the IEEE*, vol. 100, no. 3, pp. 672–686, 2012.