AdaSharing: Adaptive Data Sharing in Collaborative Robots

Linghe Kong, *Member, IEEE*, Xi Chen, Xue Liu, *Member, IEEE*, Qiao Xiang, *Member, IEEE*, Yi Gao, *Member, IEEE*, Noam Ben Baruch, and Guihai Chen

***Abstract*—Collaborative robots are multirobot systems working together for the same industrial task such as robotic assembling. To achieve an efficient collaboration, robots require not only locally sensing the environmental data but also immediately sharing these data with neighbors. However, there exists a dilemma between the large amount of sensory data and the limited wireless bandwidth. In this paper, we study the problem of maximizing the throughput of sensory data sharing in collaborative robots. This data sharing is different from the transmissions in conventional mobile networks due to the real-time sharing requirement and the vicinity sharing pattern. Thus, existing adaptation methods cannot be applied directly. To maximize the throughput in dynamic environment, we propose a novel adaptation method AdaSharing based on control theory, which jointly adapts the combination of packet rate and transmission power according to the feedback of throughput. We implement AdaSharing in a nine-robot testbed, and conduct extensive experiments to verify its feasibility and effectiveness. Simulations based on ns-2 are further conducted to evaluate AdaSharing in large-scale scenarios. Both experiment and simulation results demonstrate that AdaSharing outperforms existing methods by improving the throughput up to 23%.**

***Index Terms*—Adaptive communication, data sharing, robotic networks.**

I. INTRODUCTION

COLLABORATIVE robots [2], [30] are revolutionizing the worldwide industry. Different from traditional robots, which are usually predefined for repetitive tasks, collaborative robots are envisioned to work together as coworkers, operate safely with humans, and adapt to versatile tasks and dynamic environments. In recent years, collaborative robots attract extensive attention. In academia, an MIT project [2] studies that multiple robots assemble a chair collaboratively [see Fig. 1(a)]. Robotic soccer teams compete against each other in RoboCup [12] every year [see Fig. 1(b)]. In industry, smart robots Baxter and Sawyer [5] are newly developed to perform tasks like humans do. Such robots are replacing manpower to perform more and more tasks.

(a) (b)

Fig. 1. Examples of existing collaborative robots. (a) Multirobot assemblers. (b) RoboCup soccer team.

Sensory data sharing is fundamental to support collaborative robots. Since every robot can only sense its surrounding data, sharing these data in its vicinity is helpful as it provides the benefits of sharing different views of the targets, avoiding humans from blind spots, reacting swiftly to dynamic environments, and responding other robots' requests.

With the development of robots, various sensors are equipped, such as camera, radar, acoustic, and photosensitive sensors. These sensors generate a large amount of sensory data [26], [29]. However, it is not easy to share these data in time due to the limited wireless bandwidth. In this paper, we study the problem of maximizing the throughput of sensory data sharing, which is to fully utilize the channel bandwidth to share as many data as possible in collaborative robots.

Sensory data sharing in collaborative robots distinguishes from data transmission in conventional mobile networks due to its unique temporal and spatial characteristics. Temporally, it has stringent *data freshness* requirement, where the sensory data need to be shared in real-time. Spatially, each robot only needs to *share data with its neighbors*, where neighbors are the robots within communication range. It is unnecessary to transmit data to long-distance robots because there is no instant collaboration between them. Hence, directly applying existing

Manuscript received August 16, 2016; revised January 2, 2017; accepted April 28, 2017. Date of publication May 25, 2017; date of current version October 24, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61672349, Grant 61672353, Grant 61472252, and Grant 61502417, in part by the National Key Research and Development Program under Grant 2016YFE0100600, in part by the 973 Project under Grant 2014CB340303, in part by the Zhejiang Provincial NSFC under Grant LY16F020006, and in part by the Huawei Project under Grant HO2016050002BT. *(Corresponding author: Linghe Kong.)*

L. Kong and G. Chen are with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: linghe.kong@sjtu.edu.cn; gchen@cs.sjtu.edu.cn).

X. Chen and X. Liu are with McGill University, Montreal, QC H3A 0G4, Canada (e-mail: xi.chen11@mail; xueliu@cs.mcgill.ca).

Q. Xiang is with Yale University, New Haven, CT 06520 USA (e-mail: qiao.xiang@cs.yale.edu).

Y. Gao is with Zhejiang University, Hangzhou 310027, China (e-mail: gaoyi@zju.edu.cn).

N. B. Baruch is with Ben-Gurion University, Be'er Sheva 8499000, Israel (e-mail: noambenb@post.bgu.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2017.2708000

adaptation methods in collaborative robots cannot achieve a satisfactory result.

In research community of robot [1], [34], adaptive communication methods are studied for various objectives such as guaranteeing the connectivity [3], reducing the energy consumption [35], and adapting the communication for self-reconfigurable robots [25]. Nevertheless, to the best of our knowledge, there is no research effort toward improving sensory data sharing by adaptive communication in collaborative robots.

Maximizing the throughput of data sharing is nontrivial due to two challenges. First, *multiple tunable variables*. There are several tunable variables in robotic communications such as packet rate and transmission power. Separately or serially tuning them cannot accomplish the maximal throughput because their combined effect is ignored. It is challenging to explore the correlation among multiple variables and jointly optimize them. Second, *dynamic environment*. Mobile robots work in a highly dynamic environment. Hence, it is difficult to capture the dynamics for precise and quick reactions.

In this paper, we investigate the sensory data sharing problem from the control theory perspective [20] and propose a novel AdaSharing method for every robot, in which all tunable variables are jointly adapted to maximize the throughput. In AdaSharing, we tackle the challenges by the following techniques. First, a multi-input single-output (MISO) control model is used to present the correlation between multiple tunable variables and the throughput performance. Second, for acquiring the dynamics, we design a lightweight online trainer, which can track the dynamic correlation quickly. Leveraging the closed-loop control concept, AdaSharing updates the dynamic correlation through the feedback of throughput. In this way, the optimal control strategy for the combination of multiple variables can be determined.

We implement AdaSharing in a nine-robot testbed, where every robot prototype is established by a programmable iRobot [10] for moving and operating, a laptop for computing and controlling, and a TelosB mote [29] for sensing and transmitting. These robots work collaboratively for an assigned task, which is to transport scattered cups into a given region. Experimental results demonstrate the feasibility and the effectiveness of AdaSharing in practice. Compared with the conventional wireless protocols, AdaSharing improves the throughput by 23.7% and further saves 36.1% time on completing the task.

To explore the scalability of AdaSharing, extensive simulations are conducted to evaluate its performance in large-scale robotic networks. Simulation results show that the highly dense scenarios can further embody the advantage of AdaSharing, which outperforms existing methods by up to 23.9% throughput while maintaining the packet delay within 29 ms for data freshness.

The main contributions of this paper are threefold.

1) We study the problem of the optimization of sensory data sharing in collaborative robots, where the data freshness and the vicinity sharing pattern are considered. This problem is fundamental to support the collaborative robots.
2) We apply control theory into adaptive communication design and propose a novel AdaSharing method, which adapts the communication variables based on the feedback of throughput. This method leverages the MISO model to present the complex correlation between multiple variables and throughput. Moreover, an online trainer is designed to quickly capture the dynamics.
3) We implement AdaSharing in a nine-robot testbed and evaluate its performance. We further conduct extensive simulations to evaluate AdaSharing in large-scale robotic networks. Performance results demonstrate that AdaSharing is feasible for practical collaborative robots and outperforms existing adaptive communication methods.

II. Problem Statement

In this section, first, we present the motivation of this problem. We then describe the communication system of collaborative robots and state the problem.

A. Motivation

This work falls into the research of collaborative robots, in which multiple robots compose a mobile network and work collaboratively for assigned tasks. Sensory data sharing is an indispensable process in such a robotic network. Since any robot can only acquire its surrounding data via equipped sensors, sharing the sensory data (such as position, status, target, human, and environment information) effectively supports the collaboration among robots in dynamic scenarios.

Recent sensors generate massive sensory data. For example, the robot Baxter [5] equips the webcam with 1280×800 pixels resolution and 30 fps frame rate. The robotic fish in [29] equips various aquatic sensors and transmits the sensory data by the ZigBee wireless interface. However, sharing these data are constrained by the limited wireless bandwidth. For instance, the ZigBee module has a 2 MHz bandwidth and 250 kb/s data rate [13]. When the packet length is 133 Bytes and the packet rate is 60 Hz, only four robots would lead to the channel saturation, i.e., $133 \times 8 \times 60 \times 4 = 255\,360 \geq 250$ k. To support the collaboration, it is desired to share all sensory data. i.e., to maximize the throughput of data sharing.

Directly applying traditional wireless optimization methods into collaborative robots cannot achieve the maximal throughput due to two special characteristics in sensory data sharing.

From the temporal dimension, every robot requires the data freshness, which aims to share the sensory data immediately. It is desired that overdue data do not occupy any wireless resource. Current collaborative robots usually share their data by the broadcast manner. On one hand, the broadcast manner is more time-efficient than unicast in simultaneously sharing data with multiple neighbors. On the other hand, some collisions cannot be avoided in broadcast, which will lead to bit error and packet loss. Hence, we need to carefully control the packet rate to reduce the collision opportunity.

From the spatial dimension, sensory data sharing is also different from conventional transmission patterns in mobile networks. In conventional networks, data forwarding [33] is a source-destination transmission via relays, data dissemination [32] floods the packets from one to all, and data collection [13] is a convergecast process from all to one. On the

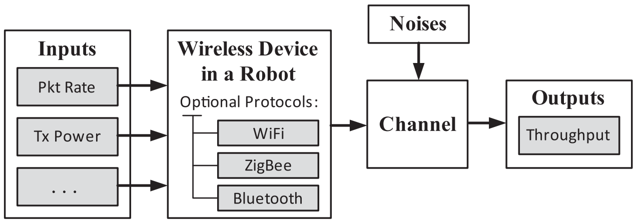


Fig. 2. Model of robotic communication system.

contrary, robots in sensory data sharing usually broadcast data to its single-hop neighbors. Multihop forwarding is unnecessary because it introduces huge traffics into networks, but there is no instant collaboration between long-distance robots. Especially, in large-scale robotic networks, local collaboration is more flexible than centralized control for dynamics. Thus, power control is required to adjust the number of neighbors for maximizing the throughput.

Due to the data freshness requirement and the vicinity sharing pattern, it is required to design an innovative and tailored method for collaborative robots.

B. System Description

In this paper, we consider a series of robots with the capabilities of operating, moving, sensing, computing, and communicating. To collaboratively accomplish the assigned tasks, these robots share the sensory data by their robotic communication systems.

The model of robotic communication system is depicted in Fig. 2, whose basic component is the wireless device. Currently, the commercial wireless devices in robots are WiFi [24], ZigBee [13], and Bluetooth [21]. All the other factors connected with the wireless device are classified into three categories: *output*, *inputs*, and *noises*. Especially, the *output* is decided by *inputs* and *noises*.

1) Output: The output is the performance of robotic communications. We consider the *throughput* as the main output, which is defined as the data received by a robot in one time slot (in this paper, one time slot is set to be 1 s by default).

2) Inputs: The inputs are the tunable variables in robotic communications. Among all the variables, we focus on *packet rate* and *transmission power*, which are two most common variables offered by general wireless protocols including WiFi, ZigBee, and Bluetooth.

1) The *packet rate* is the number of packets transmitted in one time slot, where all packets have the same length for a certain robotic application. The tunable range of packet rate is from 0 to the sensing rate, where the sensing rate is the number of sensory data generated in one time slot. The sensing rate is determined by the hardware of sensors. e.g., 30 fps webcam [5]. A high packet rate increases the throughput in an unsaturated channel. But a too high packet rate makes a channel congested, in which collisions lead to a reduction of throughput.
2) The *transmission power* is another tunable input. According to the standards, ZigBee can be adjusted from −33 to 0 dBm and WiFi [24] is from 0 to 20 dBm. Normally, a high transmission power results in a long communication range but strong interferences to more neighbors.

3) Noises: The noises are the uncontrolled variables in dynamic environment, which also affects the throughput, such as transmissions from other robots. What is worse, some noises cannot be directly measured, such as the multipath [27].

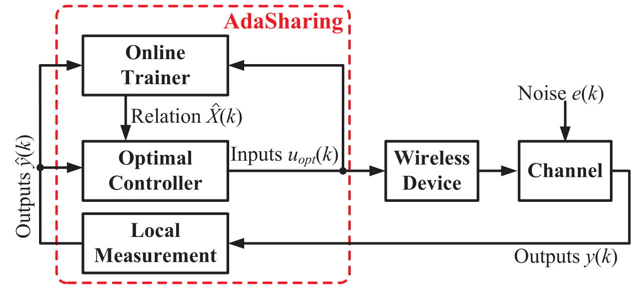


Fig. 3. Architecture of AdaSharing.

C. Problem and Challenges

Based on the above model, in order to efficiently support the collaboration in robots, we propose to study the *sensory data sharing* problem for maximizing the throughput in dynamic environment by jointly adapting the packet rate and the transmission power.

To solve this problem, we need to model the correlation between inputs and output, and then design the adaptation strategy based on the dynamic correlation. However, neither of these two steps are trivial as the following challenges exist.

Challenge 1: It is challenging to characterize the complex correlation between inputs and output. The packet rate and the transmission power are not independent. For example, when the power is increased, more robots are covered in the communication range. To maintain the maximal throughput, the packet rate should be reduced correspondingly. Thus, a joint adaptation for two inputs is necessary.

Challenge 2: It is difficult to capture the dynamic correlation precisely and quickly. Collaborative robots inherit the dynamic property from mobile networks as the robots are mobile and the environment is also dynamic. A predetermined formulation is unavailable to present the dynamic correlation because some noises cannot be measured. Hence, an online lightweight update of the correlation is needed.

III. Design of AdaSharing

In order to address the maximal data sharing problem, we propose a novel *Adaptive Sharing* (AdaSharing) method.

A. Design Overview

The basic idea of AdaSharing leverages the adaptive control [20] theory to adapt the combination of tunable variables (inputs) based on the closed-loop feedback of performance (output). The architecture of AdaSharing is illustrated in Fig. 3. AdaSharing is composed of three principal modules.

1) The *online trainer* is a lightweight learning module. It resorts to a typical MISO control model to present the correlation between inputs and output. Since such a cor-

relation is dynamic, this trainer updates the correlation estimation using the feedback of output and the inputs at every time slot.

2) The *optimal controller* module determines the optimal inputs for the next time slot. Based on the estimated correlation and the real-time feedback, the optimal inputs are desired to result in the optimal communication performance.
3) The *local measurement* module measures the output and feeds it back to *online trainer* and *optimal controller*. In order to measure the throughput of neighbors without additional overhead, the module is designed to estimate the throughput locally.

The design of AdaSharing tackles the aforementioned two challenges: First, MISO control model is used to present the complex correlation between inputs and output. Second, the local measurement and the online trainer keep pace with the dynamic correlation precisely and quickly.

With the dynamic correlation, the optimal controller determines the setting of inputs. As a result, the maximal throughput of sensory data sharing can be accomplished.

Next, we present the details of these three modules.

B. Online Trainer Design

Before introducing the design of online trainer, we first formulate the robotic communication system by a typical MISO control model. Note that the MISO here is a control model but not a MISO antenna system in communication area.

In our data sharing problem, multiple inputs are collectively denoted by an input vector $u(k)$, where k presents the kth time slot. Similarly, the output in the kth time slot is denoted by the vector $y(k)$. The number of inputs and output are denoted by i and o, respectively. Since there are $i = 2$ inputs, we have $u(k) = [u_1(k)\ u_2(k)]^T$, where u_1 is the packet rate and u_2 is the transmission power. There is only $o = 1$ output, so $y(k)$ is the throughput. We can extend the MISO model to be a general multi-input multioutput (MIMO) model if needed, in which more inputs and outputs are able to be added easily.

We adopt the MISO control model because it can bridge the inputs and the output by parameter matrices. Thus, the complex correlation between variables and throughput can be mathematically formulated.

Applying MISO has the condition of linear system. As we all know that communication system is a nonlinear system. However, in collaborative robots, the need of data sharing is frequently. Thus, a time slot can be defined as a very short duration, e.g., 1 s. In such a short duration, the robotic communication system can be considered as an approximate linear and time invariant system. Consequently, the MISO control model can be utilized in our problem.

According to the adaptive control theory [4], the correlation between multiple inputs to output can be linked by parameter matrices, which is presented by

$$\begin{aligned} y(k) = A_1(k-1)y(k-1) + \ldots + A_n(k-1)y(k-n) \\ + B_0(k-1)u(k-1) + \ldots + B_{n-1}(k-1)u(k-n) \\ + e(k) \end{aligned} \tag{1}$$

where $A_j(k)$ and $B_j(k)$ are parameter matrices with size $o \times o$ and $o \times i$, respectively, n $(0 < j < n)$ is the order of the model, and $e(k)$ is an identically distributed vector with zero mean. Moreover, e is assumed to be independent with y, u, A, and B. We consider $e(k)$ to be the noises in the robotic communication system. The order n is usually low in computer and communication systems [18], whose value can be obtained by the classic identification method in [4].

For simplicity of notation, we rewrite (1) as

$$y(k+1) = X(k)\phi(k) + e(k+1) \tag{2}$$

where

$$X(k) = [B_0(k)\ \ldots\ B_{n-1}(k)\ A_1(k)\ \ldots\ A_n(k)] \tag{3}$$

$$\begin{aligned} \phi(k) = [u^T(k)\ \ldots\ u^T(k-n+1) \\ y^T(k)\ \ldots\ y^T(k-n+1)]^T. \end{aligned} \tag{4}$$

In (3), $X(k)$ presents the correlation between inputs as well as their impact on the output. Since the robots are mobile and noises exist in the dynamic environment, the correlation matrix $X(k)$ is time-varying, which needs an online method to periodically update it.

The goal of our online trainer design is to quickly obtain the dynamic $X(k)$. We adopt the *recursive least squares* (RLS) method [18] in our online trainer because of its low computational overhead. RLS is able to update $\hat{X}(k+1)$ with only the latest sample $\phi(k)$ and the estimated correlation $\hat{X}(k)$ at kth slot.

Since RLS is a well-studied mathematic tool, we omit the derivation process. Leveraging RLS, the dynamic correlation $\hat{X}(k+1)$ can be estimated by the following:

$$\hat{X}(k+1) = \hat{X}(k) + \frac{\varepsilon(k+1)\phi^T(k)P(k-1)}{\lambda + \phi^T(k)P(k-1)\phi(k)} \tag{5}$$

$$\varepsilon(k+1) = y(k+1) - \hat{X}(k)\phi(k) \tag{6}$$

$$P(k) = \frac{P(k-1)}{\lambda} - \frac{P(k-1)\phi(k)\phi^T(k)P(k-1)}{\lambda(1+\phi^T(k)P(k-1)\phi(k))} \tag{7}$$

where $\hat{X}(k)$ is the estimate of $X(k)$, $\varepsilon(k)$ is the estimation error vector, $P(k)$ is the covariance matrix, and λ is the forgetting factor $(0 < \lambda \leqslant 1)$, which determines the weight between the older samples and the latest sample.

C. Optimal Controller Design

The optimal controller determines the values of inputs for maximizing the throughput, and we formulate it as

$$\begin{aligned} \text{Maximize}: \quad & E\{y(k+1)\} \\ \text{Subject to}: \quad & u_1(k) \in U_1 \\ & u_2(k) \in U_2 \end{aligned} \tag{8}$$

where $E\{.\}$ is the expectation operator, U_1 is the set of available packet rates, and U_2 is the set of power levels.

The design goal of (8) is that the expected y is steered to the optimal output while the inputs are subject to the alternative scopes. These constraints depend on the configuration of robots. For example, if the robots adopt 60 fps webcam as its sensor, the packet rate $u_1(k)$ ranges from 0 to 60. And if the robots utilize ZigBee as its wireless protocol, the transmission power $u_2(k)$ ranges from -33 to 0 dBm.

In accordance with (6), we have

$$\begin{aligned} E\{y(k+1)\} &= E\{\hat{X}(k)\phi(k) + \varepsilon(k+1)\} \\ &= E\{\hat{y}(k+1)\} + E\{\varepsilon(k+1)\} \\ &= \hat{y}(k+1) \end{aligned} \tag{9}$$

where $\hat{y}(k+1) = \hat{X}(k)\phi(k)$ is the estimate of $y(k+1)$, and $E\{\varepsilon(k+1)\}$ is the expectation of estimation error. According to RLS theory [4], $E\{\varepsilon(k+1)\} = 0$. Then, (8) is transformed to

$$\begin{aligned} \text{Maximize}: \;& \hat{y}(k+1) \\ \text{Subject to}: \;& u_1(k) \in U_1 \\ & u_2(k) \in U_2 \\ & \hat{y}(k+1) = \hat{X}(k)\phi(k). \end{aligned} \tag{10}$$

We find that (10) is a constrained multivariate optimization problem. In addition, each variable $u_1(k)$ or $u_2(k)$ has only finite alternative scopes. This typical optimization problem can be solved by existing methods such as direct search, gradient-based search, or quadratic programming. For simplicity, we solve (10) by the direct search method. The search space is $|U_1||U_2|$, where $|U_1|$ and $|U_2|$ are the numbers of different packet rates and power levels, respectively. Searching all $|U_1||U_2|$ combinations of inputs, the maximal output $\hat{y}(k+1)$ satisfying $\hat{y}(k+1) = \hat{X}(k)\phi(k)$ can be obtained. In this way, the searched result is the estimated optimal output $y_{\text{opt}}(k+1)$ and its corresponding input combination is the optimal control law $u_{\text{opt}}(k)$.

D. Local Measurement Design

The local measurement is used to acquire the real-time output and feed it back to the other two modules.

Recall the objective of AdaSharing is to maximize the throughput $y(k)$. It is nontrivial to acquire other robot's throughput because collaborative robots adopt the broadcast manner for vicinity sharing and there is no acknowledgement about the packet delivery information. Based on the channel reciprocity [27], neighbors can have similar transmission performance because every robot operates the same AdaSharing method. Thus, the core idea of local measurement is to utilize a robot's own throughput $\hat{y}(k)$ to approximate the average throughput of its neighbors $y(k)$. A robot's own throughput can be measured locally, which is the total amount of received data in the kth time slot.

E. Discussion: Beyond Two Inputs and One Ouput

Although AdaSharing in this paper adopts a two-input one-output MISO design, it is in fact more general. AdaSharing can be easily scaled to a MIMO design with more inputs and outputs. For instance, if robots adopt WiFi as their wireless protocol, the data rate $u_3(k)$ can be added into the input vector $u(k)$, whose range is from 6 to 54 Mb/s. In addition, outputs such as packet delivery ratio (PDR) and packet latency are also significant in some emergent robotic tasks. To taking multiple outputs into account, the PDR $y_2(k)$ and the packet latency $y_3(k)$ can be set as extra outputs and added into output vector $y(k)$. With this three-input three-output configuration, AdaSharing could adjust the equations of optimal controller and still work in a similar process.

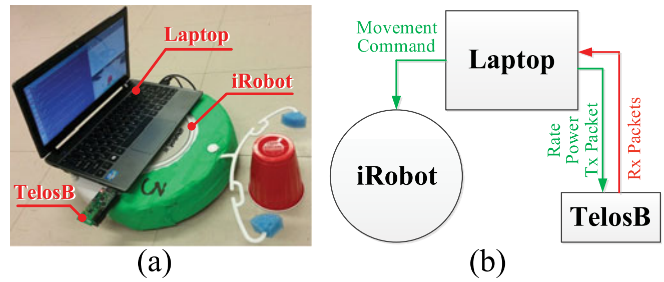


Fig. 4. Prototype of robot in our testbed.

IV. Implementation and Experiment

In order to demonstrate the efficiency of AdaSharing in practice, we implement it in a robotic testbed and evaluate its performance by extensive experiments.

A. Implementation

We design and establish a nine-robot testbed, in which every robot consists of a laptop, an iRobot Create, and a TelosB mote. A prototype of our robot is shown in Fig. 4(a).

1) The *laptop* is the "brain and eyes" of our robot. As the brain, it serves as a central controller of iRobot and TelosB, with AdaSharing implemented on it. With the feedback of throughput $\hat{y}(k)$ from TelosB, the laptop computes the optimal control law $u^*(k) = [u_1^*(k)\, u_2^*(k)]^T$. Then, it pushes the sensory data into the queue in TelosB according to the packet rate $u_1^*(k)$, and sets the transmission power as $u_2^*(k)$. In addition, the laptop is able to control the movement of iRobot by sending commands via serial port, as shown in Fig. 4(b). And the laptop is equipped with a webcam whose maximal frame rate is 60 fps.
2) The *iRobot* is the "hands and feet" of our robot. The iRobot Create [10] is a mobile robot, which moves following the commands from the laptop. We set that its straight-line speed ranges from 0 to 0.5 m/s and its angular speed is 90 °/s. In addition, we retrofit iRobot by installing a front prong, as shown in Fig. 4(a). This prong is used to transport items.
3) The *TelosB* is the "ear and mouth" of our robot, i.e., the robotic communication system for sharing data. The TelosB mote transmits and receives sensory data using ZigBee protocol, which is the most popular wireless interface in industrial robots [14]. The packet length is set to be 133 Bytes, including 127 Bytes payload and 6 Bytes header.

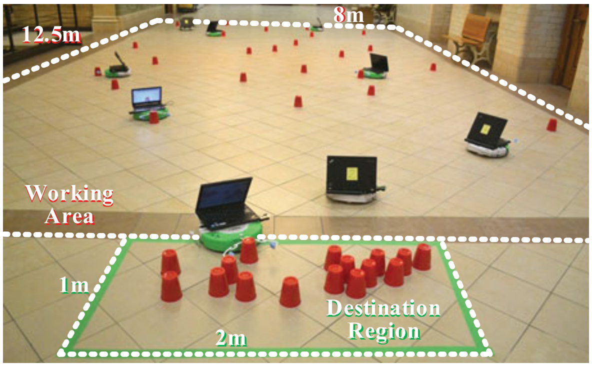


Fig. 5. Snapshot of the experiment.

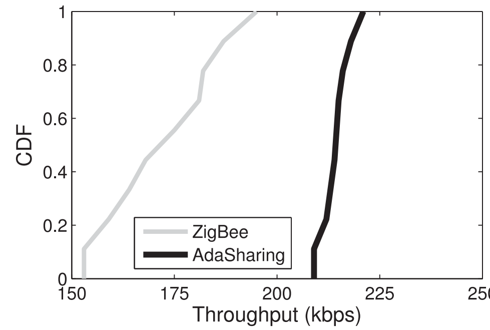


Fig. 6. CDF of throughput.

B. Collaboration Scenario: Transportation Task

A transportation task is assigned to these collaborative robots, as shown in Fig. 5. In detail, 40 sanitary cups are randomly scattered in a 12.5 m × 8 m working area. Nine robots are required to find these cups and transport them to a 2 m × 1 m destination region. Robots detect the red cups using its webcam and image processing. Every time, one robot can push only one cup using its prong. It is desired that robots can complete this task as quickly as possible.

In this transportation scenario, two kinds of collaborations are needed. First, when multiple robots detect and plan to transport the same cup, only the robot, which is closest to the cup, continues to work on this cup and all the others move on to look for new targets. Second, to avoid crash during movement, robots obey the following two priorities. First, the robot farther to the destination region yields the road rights to the closer one. Second, the slower robot yields to the faster one.

Both of these collaborations rely on the sensory data sharing, including vision, speed, direction, and position. The vision information is acquired by webcam, including whether "see" any robots, cups, humans, or boundary along with the estimation of their relative positions. The speed is measured by iRobot. In addition, different from mature industrial robots, our robots are the simple prototypes, which have no localization system such as GPS. To acquire directions and positions, we leverage a classic vision-based indoor localization system [11] with two high-resolution cameras. The views of the cameras cover the whole area, so all robots' positions can be analyzed. In addition, the directions of robots can be determined by the green head, as shown in Fig. 4(a). This localization system sends directions and positions to robots via WiFi. It does not interfere the data sharing via ZigBee as the transmissions are allocated in different channels.

C. Experiment Setting

We conduct experiments using the nine-robot testbed and assign them the transportation task. In our robotic communication system, the inputs are the packet rate and the transmission power. The rate can be tuned from 0 to 60 Hz. The power ranges from −33 to 0 dBm (corresponding to the power level from 1 to 31 in TinyOS). The output is the throughput.

In our experiment, we compare the proposed AdaSharing with the standard ZigBee protocol. Our setting is to make a highly saturated channel, because an unsaturated channel does not need any adaptation and directly maximizing all variables can achieve the maximal throughput. Hence, the rate and the power in ZigBee are fixed at 60 Hz and 0 dBm, respectively. The initialization of AdaSharing is also 60 Hz and 0 dBm. For the forgetting factor, we set the empirical value $\lambda = 0.95$. A small λ will lead to too long convergence time and a large λ will lead to inaccurate control. The optimal λ is not easy to derive because it relies on the devices, applications, and environments. Hence, we set its value by large amount of experiments in order to achieve a tradeoff between accuracy and convergence time.

The objective of AdaSharing is to support the collaborative robots in dynamics. Thus, robots can increase the transportation efficiency for time saving and reduce the total travel distance for energy saving. And the cost is only negligible computational overhead.

D. Experiment Result

In order to demonstrate the feasibility and the efficiency of AdaSharing, nine robots collaboratively operate the transportation task using AdaSharing and standard ZigBee, respectively. The cumulative distribution function (CDF) of nine robots' throughput is depicted in Fig. 6. With AdaSharing, the throughput of every robot is larger than 209 kb/s, and their average throughput achieves 214 kb/s. Compared with the average throughput of ZigBee (173 kb/s), AdaSharing significantly outperforms the standard ZigBee by improving about $(214 - 173)/173 = 23.7\%$ amount of data sharing.

The CDFs of selected packet rate and transmission power are illustrated in Figs. 7 and 8, respectively. These figures elaborate the reason why AdaSharing outperforms ZigBee. Since ZigBee adopts a greedy strategy, in which both rate and power are always set at the maximum, every robot's transmission interferes the others' and the channel is overload due to too many transmitted data. Thus, many packets are collided, leading to the low throughput. On the contrary, AdaSharing adaptively tunes the combination of rate and power to guarantee a high throughput. For example, when robots are dense, AdaSharing reduces the rate to mitigate the probability of collision. When robots are sparse, AdaSharing reduces the power, and multiple robots can transmit data without interference. Consequently, the selected

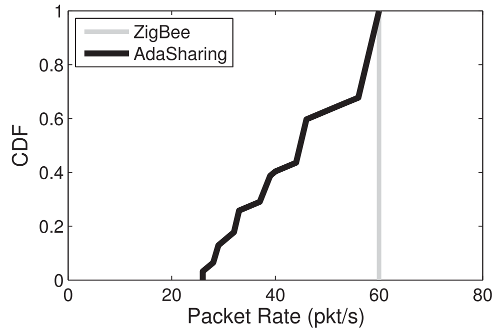


Fig. 7. CDF of packet rate.

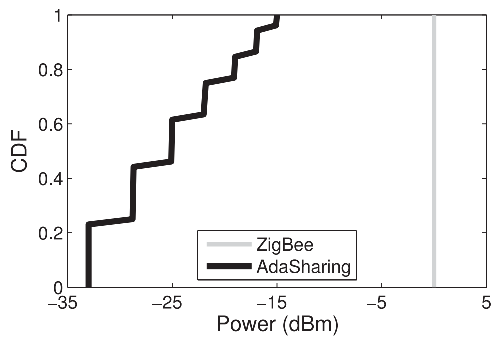


Fig. 8. CDF of transmission power.

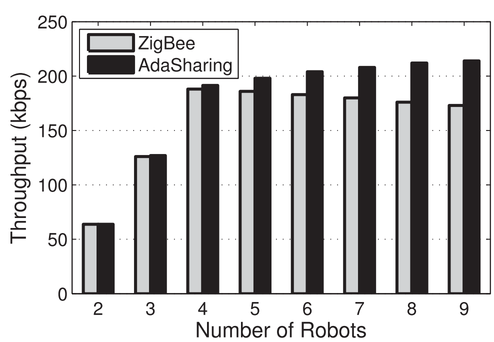


Fig. 9. Throughput in different number of robots.

rates of AdaSharing are distributed from 26 to 60 pkt/s, as shown in Fig. 7 and the selected power levels are distributed from -33 to -15 dBm, as shown in Fig. 8.

To evaluate AdaSharing in different densities, we conduct experiments by varying the number of robots from 2 to 9. In Fig. 9, when the number is smaller than 4, AdaSharing and ZigBee achieve the similar throughput. The reason is that a small number of robots cannot lead to a saturated channel, even when they transmit at their maximal rate and power. However, when the number is larger than 4, the channel becomes congested, and the strength of joint adaptation is embodied. AdaSharing is 6.5%, 11.5%, 15.6%, 20.5%, and 23.7% better than ZigBee at 5, 6, 7, 8, and 9-robot cases, respectively.

Benefiting from the high throughput, AdaSharing helps robots to accomplish task quickly. We show the time cost on complet-

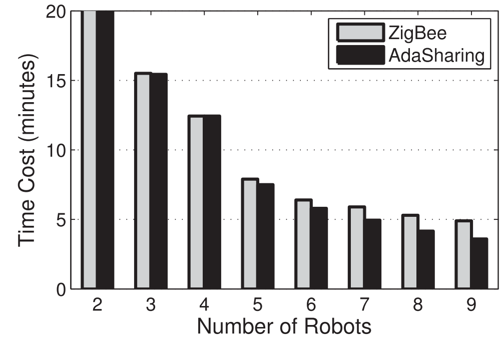


Fig. 10. Time cost in different number of robots.

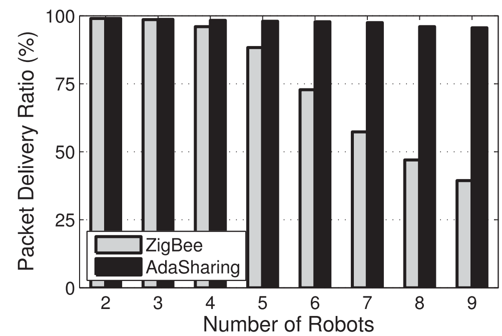


Fig. 11. PDR in different number of robots.

ing a transportation task by different number of robots in Fig. 10. Compared with ZigBee, AdaSharing saves 5.3%, 13.2%, 19.2%, 27.4%, and 36.1% time at 5, 6, 7, 8, and 9-robot cases, respectively. The reason is that the robots collaborate more efficiently when more sensory data are shared. For example, an early data sharing about the target detection can allow other robots to search for new targets early. In addition, sharing more data during the transportation would allow robots to move with a higher speed while avoiding crashes.

We also study the PDR, which is a common metric in wireless communication. PDR is defined by $N_R(j,k)/N_T(j,k)$, where $N_R(j,k)$ is the total number of received packets by robot j at kth time slot and $N_T(j,k)$ is the total number of packets transmitted to j. All transmitted packets are logged in laptops, so we can calculate $N_T(j,k)$. The average PDRs are plotted in Fig. 11. When the number of robots is increased, the PDR of ZigBee decreases because of large amount of transmissions in the saturated channel. At the 9-robot case, ZigBee's PDR is only 39.4%. In contrast, AdaSharing maintains the PDR larger than 95%. This result demonstrates the accuracy of AdaSharing's joint control that nearly all transmitted packets are successfully received.

V. Simulation

Other than the testbed experiment, we also conduct extensive ns-2 simulations to understand the performance of AdaSharing in large-scale scenarios. In addition, more parameters such as packet delay could be measured and compared in simulations.

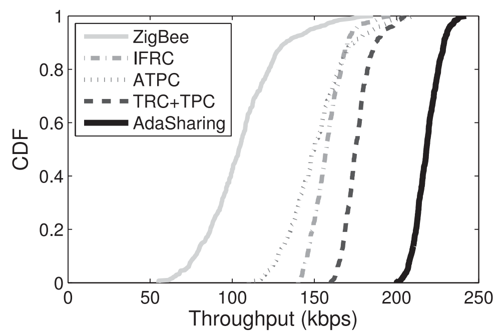


Fig. 12. Comparison of throughput.

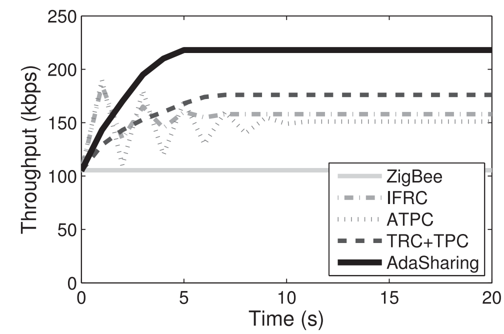


Fig. 13. Comparison of convergence.

A. Simulation Setting

Our simulations are conducted in a 1000 m $\times$ 1000 m square area. Robots move in this area following the random walk mobility model with the speed varying from 0 to 10 m/s. The total number of robots are set as 50, 100, 200, and 400. The widely adopted two-ray ground model [27] is selected to characterize the wireless channel. According to the physical layer of ZigBee [13], a packet can be successfully decoded by the receiver when its reception power is larger than -85 dBm. The additive white Gaussian noise [27] is also added.

Two inputs $u(k)$ are tunable in our simulations. The packet rate can be changed from 0 to 60 Hz and the power level ranges from -33 to 0 dBm. The inputs are allowed to be adjusted every time slot, where one time slot is set as 1 s. The main performance metric is the throughput, which is defined as the average throughput of all robots in our simulation.

We compare the performance of the following methods.

1) *ZigBee* is the standard wireless protocol with fixed packet rate and transmission power. We set a greedy strategy for ZigBee, where rate and power are always at their maximal values.
2) *IFRC* [22], *Interference-aware Fair Rate Control*, is the classic rate adaptation method in ZigBee communications, which controls the packet rate based on the awareness of interference.
3) *ATPC* [17], *Adaptive Transmission Power Control*, is the class power control method in ZigBee communications, which controls the transmission power based on the estimation of dynamic link quality.
4) *TRC+TPC* [9], *Transmission Rate Control + Transmission Power Control*, is the closest method to our work in literature, which jointly controls the transmission rate and the power in vehicular networks. Its optimization objective is to reduce the transmission delay, which is similar to improve the throughput.
5) AdaSharing is our proposed method based on the feedback control, which jointly adapts the rate and the power to achieve the maximal throughput.

B. Simulation Result

In Fig. 12, we plot the CDFs of throughput for different methods when the number of robots is 400. This figure demonstrates the effectiveness of AdaSharing in large-scale robotic networks.

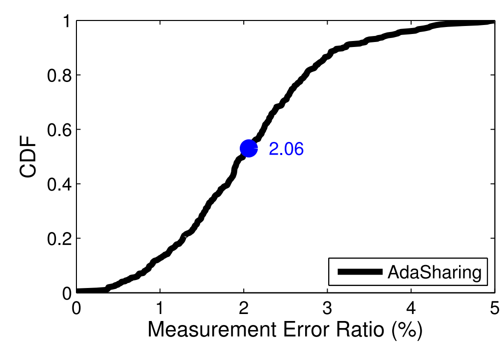


Fig. 14. Accuracy of local measurement.

First, AdaSharing outperforms the other four methods on the throughput. In detail, the average throughput of AdaSharing is 218 kb/s, while the average throughput of ZigBee, RFRC, ARPC, and TRC+TPC is 105, 158, 151, and 176 kb/s, respectively. Second, AdaSharing achieves a relative fair throughput, i.e., the throughput of every robot is similar. For example, about 90% (from 5% to 95%) robots' throughput in AdaSharing is within a small range between 209 and 227 kb/s. Nevertheless, the difference of throughput in ZigBee with the same 90% range is $141 - 69 = 72$ kb/s.

Convergence is an important metric in control systems. To verify the convergence, we conduct a simulation that 400 robots are randomly scattered and fixed in the area. The environment is stationary but every robot's adjustment will affect the others. In mobile scenario, since variables keep changing according to the dynamics, the convergence cannot be recognized clearly. Thus, we simulate convergence by stationary scenario, which is a general snapshot of mobile scenario. Then, we plot the change of throughput of one certain robot in Fig. 13. We can summarize the following. First, AdaSharing is able to converge. In Fig. 13, AdaSharing reaches a constant output after 5 s. Second, AdaSharing converges smoothly. Although IFRC, ATPC, and TRC+TPC also converge, the convergence process of AdaSharing is quicker and smoother than others.

AdaSharing adopts $\hat{y}(k)$ to replace $y(k)$ in local measurement mechanism, where $\hat{y}(k)$ is the throughput of a robot and $y(k)$ is the average throughput of its neighbors. To demonstrate the accuracy of local measurement, we show the CDF of the measurement error ratio in Fig. 14. The measurement error ratio

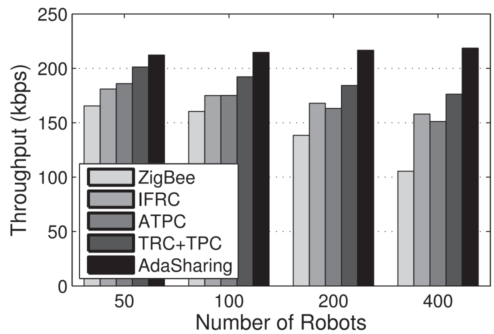


Fig. 15. Impact of density.

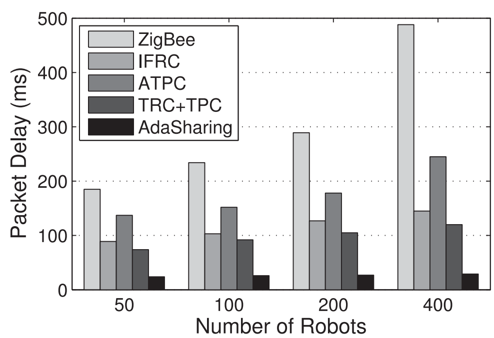


Fig. 16. Comparison of data freshness.

is defined by $\frac{|\hat{y}(k)-y(k)|}{y(k)}$. As shown in Fig. 14, all error ratios are less than 5% and the average error ratio is only 2.06%. Such results indicate that $\hat{y}(k)$ and $y(k)$ are approximately of the same value. Thus, we can leverage $\hat{y}(k)$ to estimate $y(k)$ for sensory data sharing.

To understand the impact of density, we conduct simulation with different number of robots 50, 100, 200, and 400. The results of average throughput are shown in Fig. 15. AdaSharing provides the highest throughput in any density because it keeps fully exploiting the bandwidth by adaptation. Especially, the higher density, the more percentage of throughput AdaSharing exceeds others. For example, at 50-robot case, AdaSharing shares $(212 - 201)/201 = 5.5\%$ data more than TRC+TPC. And at 400-robot case, over 23.9% data are shared by AdaSharing compared with TRC+TPC.

Regarding the data freshness, we measure the delay from a packet pushed into queue to this packet received by the farthest single-hop neighbor. The average delay of five methods are depicted in Fig. 16. We observe that AdaSharing achieves the best packet delay among all, which maintains within 29 ms. The other methods consume much more time. For example, at 400-robot case, ZigBee, IFRC, ATPC, and TRC+TPC needs 491, 135, 243, and 112 ms, respectively. Hence, AdaSharing shares more fresh data than others. AdaSharing can achieve such short delay due to its adaptation of packet rate.

With respect to the vicinity sharing pattern, we measure the *number of neighbors* N_n and the *average throughput from each neighbor* (ATfEN). The ATfEN is the amount of received data from one neighbor at the kth time slot, which is defined by

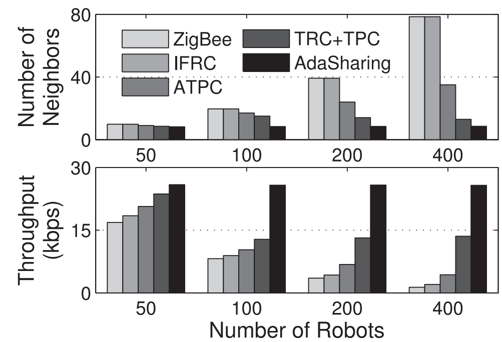


Fig. 17. Average throughput from each neighbor.

$\frac{\hat{y}(k)}{N_n(k)}$. We show the number of neighbors and ATfEN in Fig. 17. ZigBee has the most neighbors, but the lowest ATfEN. In contrast, AdaSharing has not many neighbors, but perform the highest ATfEN. Such results indicate that AdaSharing collaborates only a few short-distance robots, but the collaboration among these neighbors is extremely efficient because sufficient data are shared among them.

VI. Related Work

In the literature, plenty of adaptation methods have been studied to improve the performance of wireless communications, such as power control and rate selection. In this section, our discussion focuses on the adaptation methods in two mostly related categories: robotic networks and quasi-robotic networks.

A. Adaptation Methods in Robotic Networks

Most robotic networks [8] directly adopt the standard wireless protocols and classic adaptation methods. Commercial 802.11-based WiFi device is adopted to establish the robotic wireless networking testbed SCAN [24]. The standard ZigBee is leveraged in the robotic fishes in [29]. In [21], robots with Bluetooth are utilized for indoor localization. Moreover, the classic PCMAP [19] controls the power based on the successive samples of SNRs. And the classic RRAA [31] selects the rate based on the packet loss ratio. All these methods demand a long learning process. Hence, they are not sufficient to response the dynamics in collaborative robots.

A few robotic researches pay attention to adaptive communication for various objectives. For example, to guarantee the connectivity among autonomous robots, Fink *et al.* [3] propose to jointly adapt the mobility and the routing variables. For reducing energy consumption, Yan and Mostofi [35] develop a co-optimization framework to plan robot's speed, transmission rate, and stop time. Leveraging the hormone concept, Shen *et al.* [25] design an adaptive communication and control method for self-reconfigurable robots. To the best of our knowledge, there is still no adaptation work studying the data sharing enhancement in collaborative robots.

Adaptive neural networks are widely used to control the robotic systems. ASFSRBNC [16] applies an adaptive law to modify the fuzzy consequent parameter to manipulate a robotic system. He *et al.* [7] adopts adaptive neural networks to handle system uncertainties and disturbances in robots with full-state constraints. In [15], a new control formulation is proposed for robotic manipulator, which unifies existing neural network control tasks with prescribed performance bound. Existing neural networks are only used for robotics movement and manipulation but not communication control. Moreover, the computational complex of neural networks is higher than adaptive control.

B. Adaptation Methods in Quasi-Robotic Networks

Wireless sensor networks (WSNs), especially mobile WSNs [6], are closely related to collaborative robots, because both of them need to transmit sensory data in the networks. Lots of efforts have been contributed on adaptation methods in WSNs. For example, to eliminate the transmission congestion in dense WSNs, IFRC [22] adapts the packet rate based on the awareness of interference. For the purpose of energy-saving, ATPC [17] controls the transmission power according to the change of link quality. Directly applying these methods, nonetheless, cannot improve the sensory data sharing due to the difference on transmission patterns. While WSNs usually adopt data flooding and collection, collaborative robots only share data with neighbors.

Vehicular ad hoc networks operate in a highly dynamic environment, which are also similar to collaborative robots. In this research community, SoftRate [28] leverages physical layer information to estimate the bit error rate and determine the best rate. Rawat *et al.* [23] develop a joint controller of power and contention window size for efficient data dissemination. Huang *et al.* [9] propose to adapt transmission rate and power for transmitting safety message timely. The objective of these methods is reliable transmission of safety messages, which are not appropriate to collaborative robots for maximizing the throughput of data sharing.

VII. Conclusion

Collaborative robots are promising to replace manpower for operating industrial tasks. It is fundamental to share sensory data for efficient collaborative robots in dynamic environments. To enhance the throughput for data sharing, we presented AdaSharing, a control theory based solution jointly adapts the packet rate and the transmission power according to the feedback of throughput. Both experiments and simulations demonstrated that AdaSharing is a tailored method to collaborative robots, which significantly outperforms existing methods on throughput improvement.

We believe AdaSharing has wider implications for wireless design than explored in this paper. For example, AdaSharing motivates a more accurate control strategy in collaborative robots by exploiting the optimized communications. Moreover, AdaSharing provides a general adaptation framework, which can be extended to other systems such as driverless cars.

References

[1] E. DiGiampaolo and F. Martinelli, "Mobile robot localization using the phase of passive UHF RFID signals," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 365–376, Jan. 2014.

[2] M. Dogar, A. Spielberg, S. Baker, and D. Rus, "Multi-robot grasp planning for sequential assembly operations," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 193–200.

[3] J. Fink, A. Ribeiro, and V. Kumar, "Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams," *Proc. IEEE*, vol. 100, no. 1, pp. 164–178, Jan. 2012.

[4] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. New York, NY, USA: Dover, 2013.

[5] E. Guizzo and E. Ackerman, "How Rethink Robotics built its new Baxter robot worker," *IEEE Spectrum*, (2012, Sep. 18). [Online]. Available: http://spectrum.ieee.org/robotics/industrial-robots/rethink-robotics-baxter-robot-factory-worker

[6] L. He, L. Kong, Y. Gu, J. Pan, and T. Zhu, "Evaluating the on-demand mobile charging in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1861–1875, Sep. 2015.

[7] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 620–629, Mar. 2016.

[8] Y. Hu and A. Ribeiro, "Adaptive distributed algorithms for optimal random access channels," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2703–2715, Aug. 2011.

[9] C.-L. Huang, Y. P. Fallah, R. Sengupta, and H. Krishnan, "Adaptive intervehicle communication control for cooperative safety systems," *IEEE Netw.*, vol. 24, no. 1, pp. 6–13, Jan./Feb. 2010.

[10] J. L. Jones, "Robots at the tipping point: The road to iRobot Roomba," *IEEE Robot. Autom. Mag.*, vol. 13, no. 1, pp. 76–78, Mar. 2006.

[11] G.-H. Kim, J.-S. Kim, and K.-S. Hong, "Vision-based simultaneous localization and mapping with two cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 1671–1676.

[12] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara, "Robocup: A challenge problem for AI," *AI Mag.*, vol. 18, no. 1, p. 73, 1997.

[13] L. Kong and X. Liu, "mZig: Enabling multi-packet reception in ZigBee," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 552–565.

[14] L. Kong, D. Zhang, Z. He, Q. Xiang, J. Wan, and M. Tao, "Embracing big data with compressive sensing: A green approach in industrial wireless networks," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 53–59, Oct. 2016.

[15] X. Li and C. C. Cheah, "Adaptive neural network control of robot based on a unified objective bound," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1032–1043, May 2014.

[16] R.-J. Lian, "Adaptive self-organizing fuzzy sliding-mode radial basis-function neural-network controller for robotic systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1493–1503, 2014.

[17] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "ATPC: Adaptive transmission power control for wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sens. Syst.*, 2006, pp. 223–236.

[18] X. Liu, X. Zhu, P. Padala, Z. Wang, and S. Singhal, "Optimal multivariate control for differentiated services on a shared hosting platform," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 3792–3799.

[19] J. P. Monks, V. Bharghavan, and W.-M. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proc. IEEE INFOCOM*, 2001, pp. 219–228.

[20] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "The wireless control network: A new approach for control over networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2305–2318, Oct. 2011.

[21] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran, "Accurate mobile robot localization in indoor environments using bluetooth," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 4391–4396.

[22] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2006, pp. 63–74.

[23] D. B. Rawat, D. C. Popescu, G. Yan, and S. Olariu, "Enhancing VANET performance by joint adaptation of transmission power and contention window size," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1528–1535, Sep. 2011.

[24] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, "Connectivity maintenance in mobile wireless networks via constrained mobility," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 935–950, Jun. 2012.

[25] W.-M. Shen, B. Salemi, and P. Will, "Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 700–712, Oct. 2002.

[26] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 34–40, 2004.

[27] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[28] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 3–14.

[29] Y. Wang, R. Tan, G. Xing, X. Tan, J. Wang, and R. Zhou, "Spatiotemporal aquatic field reconstruction using robotic sensor swarm," in *Proc. IEEE 33rd Real-Time Syst. Symp.*, 2012, pp. 205–214.

[30] Z. Wang and D. Gu, "Cooperative target tracking control of multiple robots," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3232–3240, Aug. 2012.

[31] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. ACM MOBICOM*, 2006, pp. 146–157.

[32] Q. Xiang, X. Chen, L. Kong, L. Rao, and X. Liu, "Data preference matters: A new perspective of safety data dissemination in vehicular ad hoc networks," in *Proc. IEEE INFOCOM*, 2015, pp. 1149–1157.

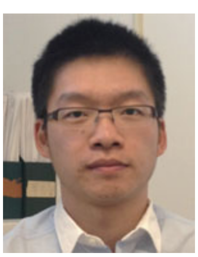
[33] F. Xu *et al.*, "Utilizing shared vehicle trajectories for data forwarding in vehicular networks," in *Proc. IEEE INFOCOM*, 2011, pp. 441–445.

[34] J.-X. Xu, Z.-Q. Guo, and T. H. Lee, "Design and implementation of a Takagi–Sugeno-type fuzzy logic controller on a two-wheeled mobile robot," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5717–5728, Dec. 2013.

[35] Y. Yan and Y. Mostofi, "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1562–1572, Apr. 2013.

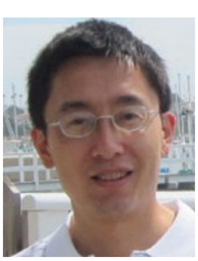
Linghe Kong (S'09–M'13) received the B.E. degree in automation from Xidian University, Xi'an, China, in 2005, the M.E. degree in telecommunications from TELECOM SudParis, Évry, France, in 2007, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2012.

He is currently a Research Professor at Shanghai Jiao Tong University. Previously, he was a Postdoctoral Fellow at Columbia University, McGill University, and Singapore University of Technology and Design. His research interests include wireless communication, sensor networks, mobile computing, Internet of things, and smart energy systems.



Xi Chen (M'07) received the B.E. and M.E. degrees in electronic engineering from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. He is currently working toward the Ph.D. degree in computer science in Prof. Xue Liu's Cyber Physical Laboratory, School of Computer Science, McGill University, Montreal, QC, Canada.

He is also a Research Assistant in Prof. Xue Liu's Cyber Physical Laboratory. His research interests include vehicle-to-vehicle (V2V) communications and vehicular networks, optimization of electric vehicles, energy and cost management of cloud computing, and green and energy-aware computing.



Xue Liu (M'07) received the B.S. degree in mathematics and the M.S. degree in automatic control from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, IL, USA in 2006.

He is an Associate Professor in the School of Computer Science, McGill University, Montreal, QC, Canada. He has also worked as the Samuel R. Thompson Associate Professor at the University of Nebraska-Lincoln and HP Labs in Palo Alto, CA, USA. His research interests include computer networks and communications, smart grid, real-time and embedded systems, cyber-physical systems, data centers, and software reliability.

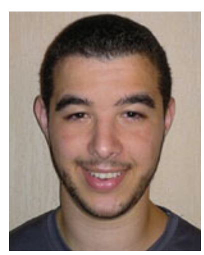
Qiao Xiang (M'14) received the B.E. degree in information security and the B.A. degree in economics from Nankai University, Tianjin, China, in 2007, and the M.E. and Ph.D. degrees in computer science from Wayne State University, Detroit, MI, USA, in 2012 and 2014, respectively.

He is a Postdoctoral Associate with the Department of Computer Science, Yale University, New Haven, CT, USA, and Tongji University, Shanghai, China. From 2014 to 2015, he was a Postdoctoral Fellow in the School of Computer Science, McGill University. His research interests include wireless cyber physical systems, vehicular networks, wireless sensor networks, smart grid, and network economics.

Yi Gao (S'11–M'15) received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2009 and 2014, respectively.

He is currently a Research Assistant Professor at Zhejiang University. His research interests include wireless sensor networks, network measurement, and mobile computing.

Dr. Gao is a member of the China Computer Federation.



Noam Ben Baruch was born in Israel in 1990. He is currently working toward the Master's degree in computer science at Ben-Gurion University, Be'er Sheva, Israel.

He joined Intel Corporation, Qiryat-Gat, Israel, in 2015 as a Manufacturing Engineer.

Mr. Baruch is a member of the Electrical Engineers Association of Israel.

Guihai Chen received the B.E. degree in computer software from Nanjing University, Nanjing, China, in 1984, the M.E. degree in computer science from Southeast University, Nanjing, China, in 1987, and the Ph.D. degree in computer science from the University of Hong Kong, Hong Kong, in 1997.

He is a Distinguished Professor at Shanghai Jiaotong University, Shanghai, China. He has been invited as a Visiting Professor by many universities, including Kyushu Institute of Technology, Japan, in 1998, University of Queensland, Australia, in 2000, and Wayne State University, USA, during September 2001 to August 2003. He has a wide range of research interests with focus on sensor networks, peer-to-peer computing, high-performance computer architecture, and combinatorics.