Computers and Electronics in Agriculture 150 (2018) 439–454

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Agricultural cyber physical system collaboration for greenhouse stress management

Ping Guo[a,*], Puwadol. Dusadeeringsikul[b], Shimon Y. Nof[b]

[a] Network Monitoring Center of Jiangsu Province, School of Computer and Software, Nanjing University of Information Science and Technology, China
[b] PRISM Center and School of Industrial Engineering, Purdue University, IN, USA

ARTICLE INFO

Keywords:
Agricultural cyber physical system
Collaborative workflow
Application of CCT
Greenhouse monitoring, detection, response

ABSTRACT

This article presents a CPS (Cyber Physical System) oriented framework and workflow for agricultural greenhouse stresses management, called MDR (Monitoring, detecting and responding)-CPS. MDR-CPS has been designed to focus on monitoring, detecting and responding to various types of stress. CCT (Collaborative Control Theory) is applied in MDR-CPS to deploy CRP (Collaborative Requirements Planning), address CEs (Conflicts and Errors) and enable a collaborative architecture for better CPS interactions. Analytic studies and simulation experiments are conducted to compare our scheme with an alternative scheme for agricultural greenhouse stress management. The results show that MDR-CPS performance is better than the compared scheme, in terms of detection cooperated with human, CEs tolerance, and emergency response. Focus in this design and development is on the collaboration among sensors, robot, and human to improve the MDR-CPS' performance.

1. Introduction

Agricultural plants encounter many unexpected and abnormal stress situations, even in a greenhouse, such as abnormalities in temperature, humidity, water levels, disease emergence, and pests. If such abnormalities are not dealt with in a timely manner, through detection and localization, proper response and prevention, they may cause severe and irreparable damages (Ari et al., 2015). Sensors have been found effective in various types of application in agricultural monitoring (Gongal et al, 2015). Physical phenomena such as temperature, humidity, and rainfall over an agricultural region can be monitored by sensors. However, sensors produce massive amounts of data even in a very short period. If all those data must be analyzed with timely response by humans, it will impose unimaginable and tedious workloads. On the other hand, sensors cannot complete monitoring and responding tasks in typical agricultural environments without additional support, e.g., from humans and robots (Ko et al., 2015a, 2015b). Generally, robots are better suited than humans to take over repetitive tasks, report legitimate agricultural abnormal situations and conditions as soon as they emerge. Considering pests on plant leaves and stems rapidly sensed by infrared sensors that are faster than videos, agricultural experts can receive those abnormal reports, interpret them and dispatch a robot to arrive at that location, take pictures or record videos, and transmit them back to agricultural experts by WIFI or 4G.

The description above is a typical agricultural CPS (Cyber Physical System). With rapid development in information technology and cybernetics, intensive computing resources are used to connect computerized physical devices to provide control, communication, coordination, and collaboration. Networked manufacturing systems, intelligent transportation systems, smart infrastructures, and power grids are all appropriate examples of emerging CPSs (Zhong and Nof, 2015). Though CPSs have attracted significant research interest because of their promising applications across different domains; how to effectively model CPSs in real applications is still a challenge (Nayak et al., 2016). According to these studies, multiple factors have impact on agricultural CPS including humans, sensors, robots, agricultural plants and data, etc. How to make them work smoothly and operate in a way of cooperation, while avoiding conflicts, errors and disruptions, needs to be considered and designed carefully.

Recently, CCT (Collaborative Control Theory) has been widely applied to deal with the uncertainties in the market demand and capacity (Nof, 2007; Ko and Nof, 2012). Moghaddam and Nof (2014) combined two CCT-based protocols—Demand and Capacity Sharing Protocol (DCSP) and Best Matching Protocol (BMP) to maximize profit and resource utilization of supply enterprises. As to the management of conflicts and errors, a distributed conflict and error detection prediction network is designed and applied to a case study of collaborative control for a robotic nuclear decommissioning task (Zhong et al., 2013). An agent-base and distributed algorithm to identify and prevent errors (AEPA) was presented in production and service (Chen and Nof, 2012).

* Corresponding author.
E-mail address: guoping@nuist.edu.cn (P. Guo).

https://doi.org/10.1016/j.compag.2018.05.022
Received 2 June 2017; Received in revised form 14 May 2018; Accepted 19 May 2018
0168-1699/

Two constraint-based conflict and error prevention and detection (CEPD) algorithms, centralized and decentralized algorithms, were designed by Chen and Nof (2012). These previous concepts and methods could be suitably introduced to control and management in the agricultural procedure. The research reported in this article aims to facilitate the use of CCT in handling these uncertainties; and design a CCT-based CPS collaboration platform for prevention, detection and response system, targeting the stress situations and conditions in agricultural greenhouses. Response to detected stresses is considered too but is left for further research. The premise of this research is that to be effective. To summarize, this research aims at:

- Reducing uncertainty in agricultural CPS according to combined humans, sensors and robots who cooperate to monitor, detect, diagnose and respond to stresses in an agricultural greenhouse.
- Designing collaborative workflow based on the designed agricultural CPS.
- Applying collaborative control theory to the designed agricultural CPS.

This article is organized as follows. In Section 2, related work is described on Wireless Sensor Networks (WSNs), robots and CPS in agricultural areas. In Section 3, system framework and workflow are designed and elaborated, also, collaboration control theory is explained and applied for the designed agricultural CPS. The designed workflow and protocol are illustrated, evaluated, and validated through analysis and experiments in Section 4. Finally, concluding remarks and planned future research are found in Section 5.

2. Related work

Many agricultural crops and food is lost because of the damage by plant diseases, pests, and rats. Intensive research on monitoring and detection in agricultural environments has been deployed worldwide. New technologies such as CPS, Global Positioning Systems (GPS), WSN, and Agricultural Robots (AR) are widely used. In this research, we integrate WSN, a robot and humans to cooperate in an agricultural CPS for stresses monitoring and detection. First, we review applications of WSN, robot, CPS in agricultural field, respectively.

2.1. WSN in agriculture

WSN plays an important role in agriculture for monitoring different activities in the cultivation field. According to WSN application in agriculture, there are three typical WSNs.

First, WSN is deployed for monitoring and detection. For instance, an environmental monitoring system was implemented and was capable of measuring temperature, humidity, illumination, soil moisture, CO2 concentration of greenhouse using a sensor array and the digital signal processing (DSP) board (Kumar et al., 2010a,b). Humidity sensors were deployed to observe whether humidity influences transpiration, condensation, and disease incidence in a tomato greenhouse (Hoshi et al., 2016). Second, WSN in precision agriculture has drawn greater attention in recent years. Blackmore (1994b) defined it as a comprehensive system designed to optimize agricultural production by carefully tailoring soil and crop management to correspond to the unique conditions found in each field, while maintaining environmental quality. A scheme based on the collaboration of integrated system was proposed for automated irrigation management with an advanced novel routing protocol for wireless sensor networks in precision agriculture (Nikolidaks et al., 2015). Third, WSN was deployed for monitoring animal agriculture. For instance, a potential utilization of wireless sensors for increasing livestock production was investigated (Wang et al., 2006). A smart farm was developed to apply wireless sensor network for animal agriculture (Wark et al., 2007).

2.2. Robots in agriculture

The idea of robotic agriculture is new with the breakthrough of sensors' constraints, recently there is an ever-increasing awareness of the necessity to develop and apply robotic systems in the novel fields of agriculture, forestry, greenhouses, horticulture, etc. (Belforte et al. 2006, Gay et al. 2008, Mcintosh 2012). Smart vehicles should be capable of working 24 h a day all year round, in most weather conditions and have the intelligence embedded within them to behave sensibly in a semi-natural environment, unattended, while carrying out a useful task (Pedersen et al., 2005). However, Contrary to industrial applications which are usually well-specified and known a priori, an agricultural robot must deal with an unstructured, uncertain and varying environment which cannot be predetermined. Fundamental robot applications in agriculture include the followings.

First, agricultural robots replace farmers to take on tedious and repetitive operations, such as precise fertilization and spraying, optimal irrigation, and selective harvesting (Reid et al., 2000, Keicher and Seufert, 2000). A comprehensive review about the state-of-the-art in robotic fruit harvesting and challenges ahead was published by Bac et al. (2014) and showed that over the past three decades research has been carried out for about 50 systems for harvesting, e.g., apples, oranges, tomatoes, cucumbers, strawberries, and melons. Second, agricultural robots aid farmers to improve and enhance agricultural activities, such as inspection and detection of crops and plants diseases, or pest disasters. For instance, a weed control robot was developed by Astrand and Baerveldt (2002). With breakthroughs in sensors' reliability, more robots are equipped with different kinds of sensors to navigate in the fields, to carry on precise inspection and detection. For example, a watch-dog robot with cameras was deployed to automatically collect information in agricultural fields (Nagasaka et al., 2004). A remote monitoring system of agricultural robots using web application was described to make clear conditions about robots' combination and adequately manage agricultural task data (Ishibashi et al., 2013). A prototype agricultural mobile robotic platform was designed and developed for robots to automatically navigate in pesticide spraying application (Ko et al., 2015a, 2015b).

2.3. CPS in agriculture

Applications of CPS arguably have the potential to dwarf the 21th century IT revolution, which includes high confidence medical devices and systems, traffic control and safety, critical infrastructure control, and so on. The key role of cyber in such complex systems and networks is its ability to endow computational, operational intelligence to handle and overcome real time challenges and uncertainties. Agricultural field is one of the complex domains that can benefit from applicable CPS. A swarm of cooperative sensors was utilized within a cyber-physical system framework to optimize the use of pesticides with precision spraying (Stark et al., 2013). Another precision agriculture architecture based on CPS technology was developed (Nie et al., 2014), including three layers: the physical layer, the network layer, and the decision layer. Every layer was analyzed in detail, helping the exploration of CPS in precision agriculture. An integrated open geospatial web service-enabled cyber-physical infrastructure was proposed (Chen et al., 2015) to acquire, integrate, process, and distribute monitoring information from the physical sensor space over the World Wide Web. With the development of information technology, new monitoring and detection technologies have been utilized in agricultural greenhouses. The wide applications of WSN and robots, introducing a human operator into the system can help improve performance and simplify the robotic system (Bechar et al., 2009).

WSNs and robots are the currently popular monitoring and detecting techniques in agricultural applications, however, both are highly specific and tailored prototype and cannot be used on a broad scale. The full potential of sensor based disease detection has still not been exploited (Mahlein, 2016). They have their advantages and disadvantages, for instance, WSN can be deployed in a large scale of space, but are constraint by their computing, storage, power abilities; mobile robots cannot carry on a large-scale sensing task timely, but they can provide more detailed data of a small-scale detecting, processing and communicating immediately. If we rely only on sensors or robots to detect plant disease, it is difficult to accurate estimates of disease incidence, disease severity, and the negative effects of disease on the quality and quantity. Based on those previous work, we try to organize sensors, a robot, humans, to deploy a collaborative MDR-CPS for an agricultural greenhouse. To the best of our knowledge, collaboration of sensors, robots and humans is seldom studied and designed in agricultural management. Moreover, the research is motivated by the ongoing maturation of expert system's image processing and pattern recognition, and then promote the advanced applications in agriculture.

3. MDR-CPS framework and workflow

In this section, we focus on the workflow of the agriculture CPS designed for monitoring and detecting stresses, and deploy collaborative control theory to make sure the MDR-CPS works efficiently, optimizes performance, and decreases cost.

3.1. MDR-CPS framework

We design MDR-CPS framework as presented in Fig. 1.

3.1.1. WSN deployment

Sensors are the essential components of the overall system for environmental monitoring and control. As the crop conditions inside greenhouses are moderated, implementation of wireless sensor technologies is easier than in outdoor applications. The deployment of WSN refers to deployment of sensor nodes located in greenhouses to provide information of environmental parameters that influence the development of the agricultural crops. There are different kinds of heterogeneous agricultural sensors in our model: monitoring the environmental parameters or abiotic stresses, such as humidity, temperature, pressure, CO_2 density, sunshine density and water levels; monitoring diseases or static biotic stresses on crops, for instance, bacteria, fungi, viruses; monitoring mobile biotic stresses on crops, such as insects, rats and other unexpected intruders; monitoring chemical insults, such as crops nutrients, soil PH value and pesticides influence on crops. Sensor types are shown in Table 1.

Parameters of type 1 and type 2 in the Table1 are simpler to monitor by stationary sensors or wireless sensor networks. Parameters of type 3 are usually monitored by sensors that are difficult to be fixed on leaves or stems where they need to be monitored. In our scenario, sensors for monitoring type 3 are equipped on mobile robots, and robots navigate to the crops where it needs to be detected. Parameters of type 4 and 5 are usually to be sensed or captured by optical/thermal sensors or HD cameras, however, they are usually limited by computing ability and memory, photos transmitted may not be clear enough to be analyzed. Therefore, humans sometimes still need to dispatch a robot to obtain more information.

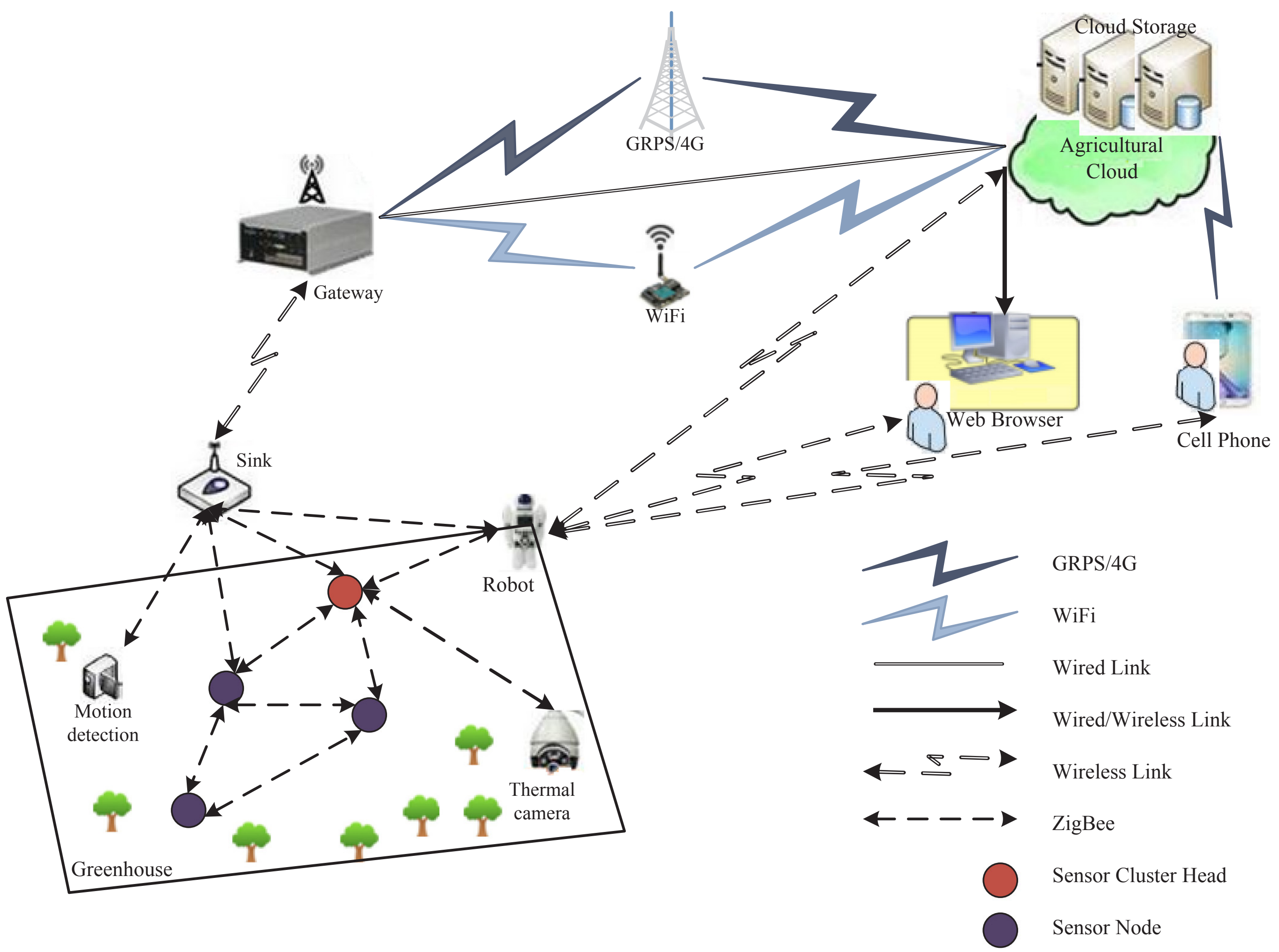

Fig. 1. Agricultural CPS framework.

Table 1
Different sensors monitoring different parameters.

Type	Monitoring category	Monitoring parameters	Monitoring devices
1	Environmental parameters	Temperature, humidity, CO_2 density, sunshine density, water levels	Stationary sensors
2	Soil parameters	Soil water density, soil pesticides, soil temperature, soil PH value, soil compaction	Stationary sensors
3	Crops growth parameters	Leaves' temperature, leaves' humidity, crony temperature, stem micro change, fruits swelling	Sensors equipped on mobile robots
4	Crops diseases	Bacteria, fungi, viruses	Stationary sensors + HD Cameras + mobile robots
5	Intruders	Insects, rats, bugs, worms	

3.1.2. Cloud platform

Agricultural cloud platform in our model is used in the agricultural field based on a few server clusters. It contains two components which are cloud storage and cloud computing/expert system, and not only stores a great deal of sensing data, but provides services, such as crop diseases analysis, intruders' alarm, and stresses identified. Agricultural cloud is constructed based on cloud computing theory and technology with the advantage of low cost, keeping and maintaining rich resources, and reducing the burden on farmers. According to the trained database, expert system can determine whether there are special stresses on crops or plants. For instance, if there is disease on plant leaves, cameras take and transmit images to the cloud platform, after expert system processing, and an appropriate alarm is delivered to farmers. It is the prior and current art of machine learning and artificial intelligence. However, we do not focus on how to establish and train the expert system in this article. Here we focus on how the agricultural CPS can be controlled by collaborative workflow to monitor, detect and respond to agricultural stresses.

3.1.3. Transmission mode

The network layer provides routing and data aggregation services. As shown in Fig. 1, in our framework, sensors transmit data to sink nodes, cameras, and sink nodes connect to the gateway through wireless link. The gateway connects the agricultural cloud by GPRS/4G, Internet, WIFI, or local area networks. Users or farmers can access agricultural data through web browser or smart phone. The detailed transmission technologies are shown in Table 2.

3.1.4. Agricultural robots and other actuators

Sensors monitor physical environment and actuators activate physical processes. Terminal computation module contains basic executive rules of actuator and has small storage capacity of real-time data. For instance, if temperature, humidity, and solar radiation do not match the preset parameters, the interfaces between software and hardware trigger the corresponding hardware/actuation equipment to adjust automatically. However, we pay much attention to agricultural robots in our model. We design an agricultural robot to aid detection in special situations for special stresses. Though sensors can do much of the monitoring work, and can obtain pictures or photos, they are limited by power, stationary location, and transmission ability. For instance, suppose the agricultural expert system finds out there may be an abnormal situation, such as fungi on plant leaves, according to photos obtained by cameras. It is difficult to decide what kind of disease, and at what scale it happens, because the data transmitted by WSN are insufficient, or unclear. That is why farmers need a mobile robot to reach the nodes which convey ambiguous images in that scenario. When the robot arrives at the given location, cameras installed on the robot could take more pictures or record a video, and transmit them to agricultural expert system to carry out further analysis. That is a reasonable requirement for the robot, because it can be designed as a mobile server with powerful computing ability, large memory and sufficient electrical power. According to the descriptions above, we can determine the computational structure of agricultural robot for MDR-CPS, as shown in Fig. 2. The purpose of the robot computer is to run the necessary software for interfacing to the robot platform and sensors, sensor information processing, mission planning and execution, navigation, implementation control, user interface, network communication, etc.

Table 2
Communication technologies applied in MDR-CPS (Fig. 1).

Connect methods	Application areas	Connection attribute
ZigBee	Sensors to sensors; sensors to sink nodes; robots to sensors or sink nodes	Wireless link
GPRS/4G, Internet, WIFI, LAN	Gateway to agricultural cloud; agricultural cloud to web browser (users) or smart phone; robots to gateway	Wired or wireless link

3.2. Workflow for MDR-CPS framework

In this section, we elaborate on the workflow in MDR-CPS framework, as shown in Fig. 3. The workflow chart is shown in Fig. 4.

The detailed workflow is described below:

Step1: MDR-CPS system initialization;
Step2: Sensors have been triggered to monitor the greenhouse;
Step3: Sensed data have been transmitted to agricultural cloud;
Step4: Data have been saved in the agricultural cloud storage;
Step5: Data have been analyzed by agricultural expert system;
Step6: If no stresses have been found, no action;
Step7: If confirmed stresses have been detected, corresponding mechanism will be triggered;
Step8: If agricultural expert system cannot confirm, because photos are unclear or insufficient, then produce an informed alarm signal and deliver it to humans;
- *Step8.1*: Humans order a robot to navigate to a given area to collect more data upon receiving the alarm signal;
- *Step8.2*: The robot is launched to navigate when it receives an order from humans;
- *Step8.3*: The robot takes photos or records videos when it reaches the given crops in term of the order;
- *Step8.4*: The robot transmits the data to the agricultural cloud when it finishes collection;

Step4-5: Agricultural expert system detects again according to the new data, if stresses can be confirmed, the procedure turns to step 7, otherwise step 8.

If humans receive a two-round alarm signal for one given plant or sensor packet, they can decide whether to order a robot to check the area, or check by a farmer.

The workflow is described by a chart with conversion conditions, as shown in Fig. 4.

3.3. Collaboration control theory in MDR-CPS

3.3.1. Collaborative requirement plan

According to our workflow and CCT (Nof et al., 2015), we deploy two stages for collaboration requirement planning, CRP-I and CRP-II. In

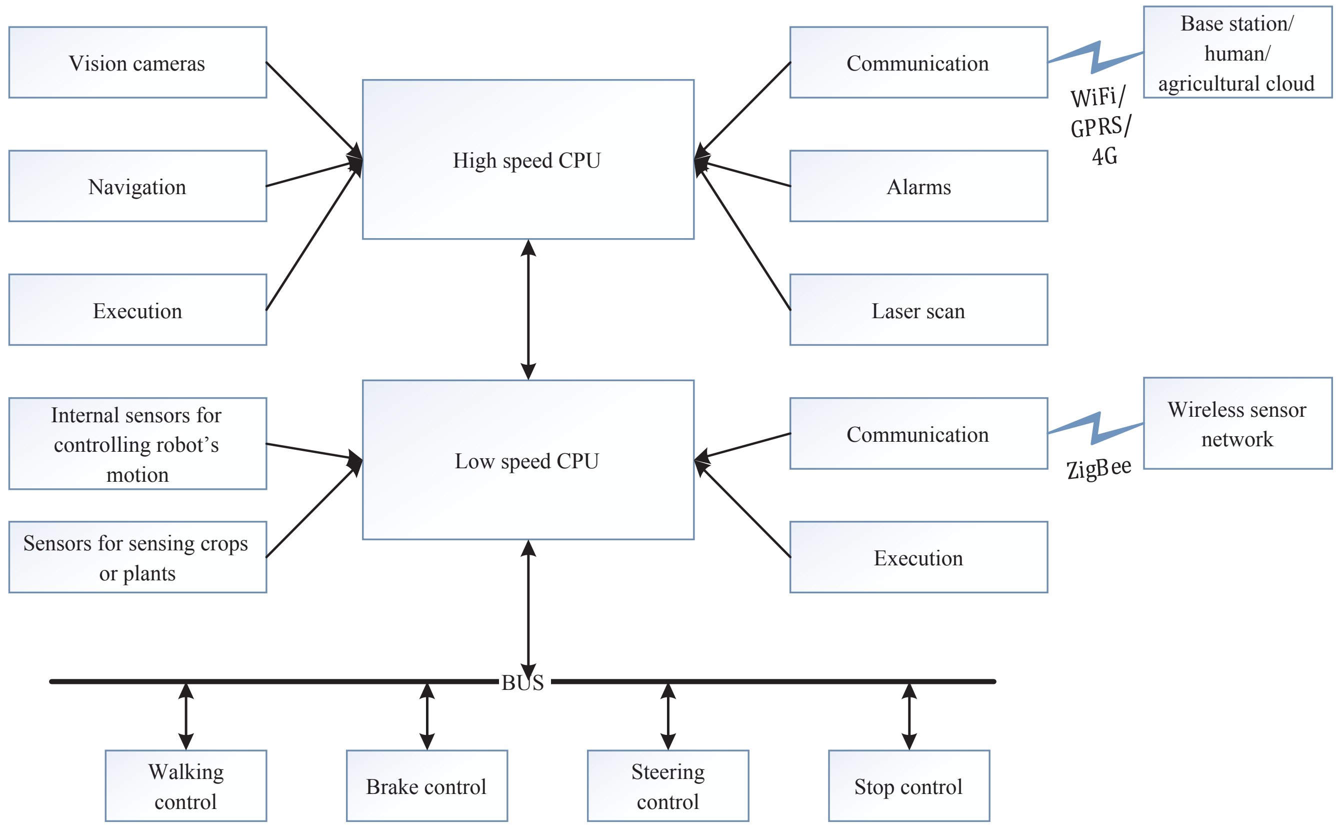

Fig. 2. Computational structure for a MDR-CPS agricultural robot.

CRP-I, a detailed requirement plan is generated, based on the work objectives and available resources. We represent tasks by set $T = \{T1, T2, \ldots, T11\}$, and resources by $R = \{R1, R2, R3, R4\}$. In the first stage of CRP-I plan generation, we determine the plan for the MDR-CPS, shown in Fig. 5.

In the second stage CRP-II (Nof et al., 2015) plan execution & revision, Fig. 6 shows an application of CRP plan in our agricultural MDR-CPS. CRP-II executes the plan generated by CRP-I real-time and revises the plan following spatial and temporal challenges, changes, and constraints. The purpose of CRP-II is to assign or reassign tasks to resources. CRP-II identifies all existing plans for each task, resolves the conflicts between the plans, and supports collaboration within a plan.

3.3.2. Collaborative architecture for MDR-CPS

The collaborations are needed to effectively utilize the available resources to achieve the required tasks in less time, with less errors and less efforts. The collaborations can be divided into five aspects in our model: collaborative sensing, collaborative data processing, collaborative communication, collaborative acting, and collaborative control, shown in Fig. 7.

Collaborative sensing means that sensors take charge of sensing environmental data and static plant physical attribute data, and a robot equipped with special sensors to approach the targeted plant and sense attributes, such as plant leaves' temperature, humidity, fruit swelling extent, and disease on leaves or stems. Collaborative processing refers that except for remote control center can analyze data, the robot is also equipped with high-end computer units that can be used by applications needing high-performance computing, such as high-resolution image processing, video processing, and pattern recognition. That is usually more important if they are operating in areas that are far from base stations and when the processing results are needed instantaneously to trigger some type of suitable actions. Collaborative communication is an important part in MDR-CPS. There are still specific situations in which the robot downloads data from sensors directly, at the same time, it collects data through sensors installed on it, and then transmits all the data to expert system or humans in real-time to diagnose plant disease. Collaborative acting requires sensors, robots, and humans to cooperate to monitor, detect and respond effectively. Collaborative acting is a crucial factor to effectively detect and quickly determine the stress, needing to be optimized for high quality accomplishment of results. Collaborative control requires that different control mechanisms are needed to coordinate among multiple parties to achieve a specific task, effectively use resources, provide safe operations, and control the fault tolerant mechanisms.

3.3.3. Error prevention and conflict resolution (EPCR)

This CCT Principle deals with cyber-supported detection of errors and conflicts among collaborating agents, and the cost associated with resolving the detected errors and conflicts. Naturally, any system that cannot overcome effectively its errors and conflicts will get out of control and eventually collapse. In MDR-CPS, we assume that each task matches an agent to be executed, and next we describe the definition of an error and a conflict.

Error: It is any input, output or intermediate result in an e-System, which does not meet the predefined specifications, expectations, or comparison objectives. An error is defined as follows (Nof et al., 2015):

$$\exists\ E[\omega_a(t)],\quad \text{if} s_{\omega_a} \overset{Dissatisfy}{\rightarrow} K_r(t) \tag{1}$$

where E: *Error*; $\omega_a(t)$:*Agent a at time t*; s_{ω_a}:*State of a at time t*; $K_r(t)$:*System constraint r at time t.*

Conflict: It refers to the inconsistencies between the goals, plans, task, and e-Activities of collaborating agents. A conflict is defined as follows (Nof et al., 2015):

$$\exists\ C[\Omega(t)],\quad \text{if} s_{\Omega} \overset{Dissatisfy}{\rightarrow} K_r(t) \tag{2}$$

where

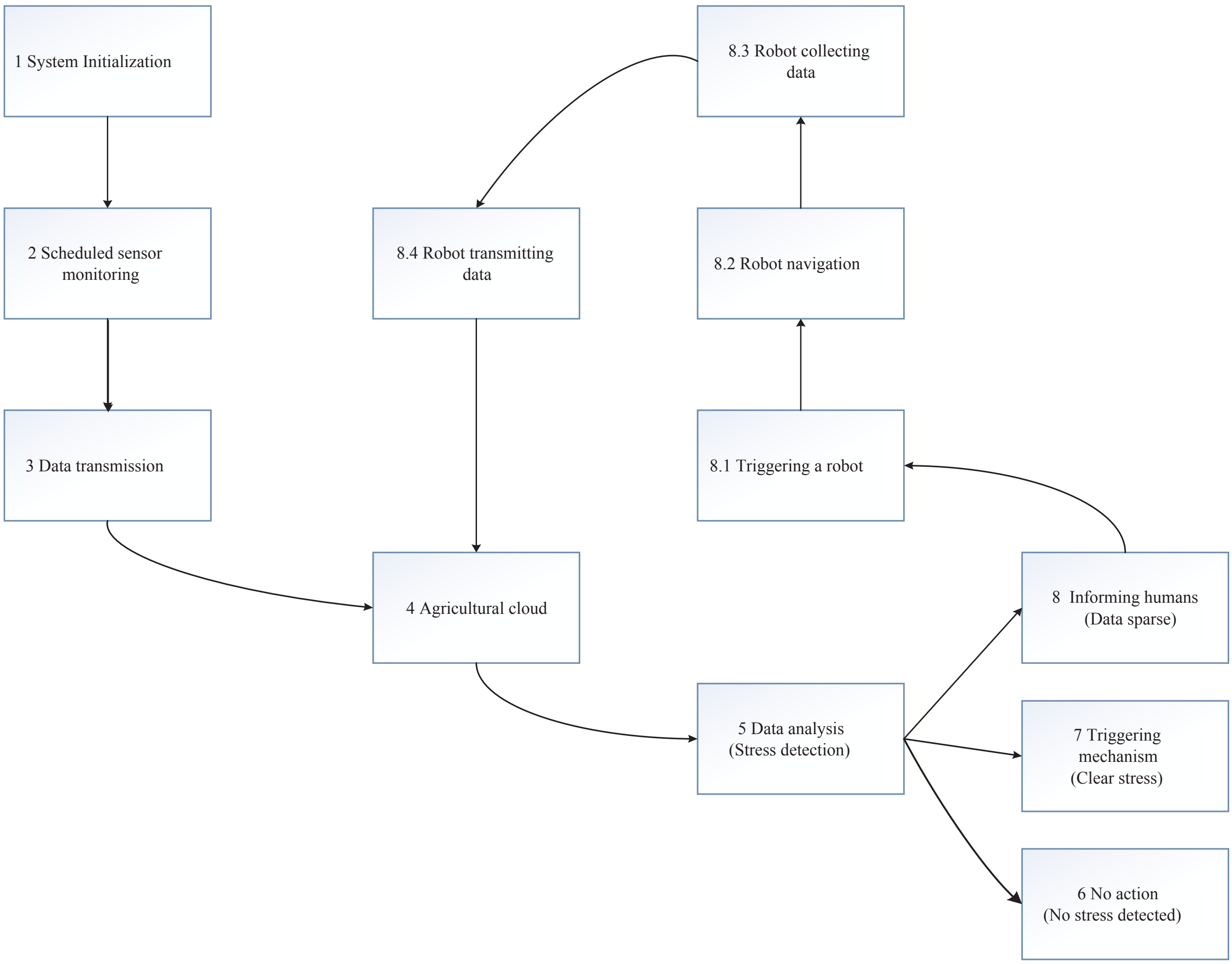

Fig. 3. MDR-CPS workflow diagram.

C: *Conflict*; $\Omega(t)$: *Integrated (network) agent a at time t*; s_Ω: *State of integrated agent* Ω*at time t*; $K_r(t)$: *System constraint r at time t*.

In MDR-CPS model, the errors and conflicts may emerge at any devices or cells. We mainly consider potential errors and conflicts by sensors, robots, and humans in Table 3.

The following example illustrates an error and conflict produced by a robot. Suppose the robot navigates at a greenhouse at 10o'clock in the morning according to a scheduled routine. During the process, it receives an order from a farmer to arrive at a given location where data delivered by a stationary sensor were found to be suspicious. Define a navigating task as *Event1(t)*, and a locating task as *Event2(t)*, and there are two agents *A1(t), A2(t)* matching them respectively, in terms of Eq. (2), define a conflict for the robot at time *t* as Eq. (3):

$$\exists\ (Event1(t) \cap Event2(t)) \xrightarrow{k_r(t)\ is\ \ a\ \ robot\ \ execute\ only\ \ one\ \ task\ \ at\ time\ t} C(\Omega(t))\ produce \tag{3}$$

When a conflict occurred, first, there is a mechanism to detect the conflict, for instance, the robot can call an interrupt to report the conflict to the farmer or control center. The following steps are specified for the algorithm to resolve a conflict in this scenario.

Algorithm 1: Resolve conflict in robot tasks case

Step 1: Detection. A control module on the robot calls an interrupt to report the conflict to the farmer or control center.

Step 2: Identification. According to the definition of conflict, a control center identifies it is a conflict.

Step 3: Diagnostics. Control center determines the type, magnitude, time and cause of the out-of-control status.

Step 4: Control center analyzes, predicts, and prevents propagation of the conflict.

Step 5: Conflict resolution. Control center resolves the conflict with a method that defines every event priority. For example, a mechanism is set to event's priorities. Define regulations as:

- Regular events' priorities are lower than emergent events.
- Earlier emergent events' priorities are higher than later emergent events' priorities. (It can be changed with emergency priority code if needed.)
- A robot can execute only one event task (order) at time *t*.

Step 6: Exception handing. Managing exceptions, i.e., constructive deviations from the process. In this example, the robot will respond the emergent event (new order) first even during the process of pre-scheduled navigation. Fig. 8 shows the process.

4. Analysis and experiments

Usually, there are three methods to monitor the greenhouse. (1) Wireless sensor network monitors the greenhouse (WSN). (2) A mobile robot mounted with sensors monitors the greenhouse (R + S). (3) Our MDR-CPS scheme, designated as (H + R + S) monitors the greenhouse. We compare them with different stress detection as shown in Table 4.

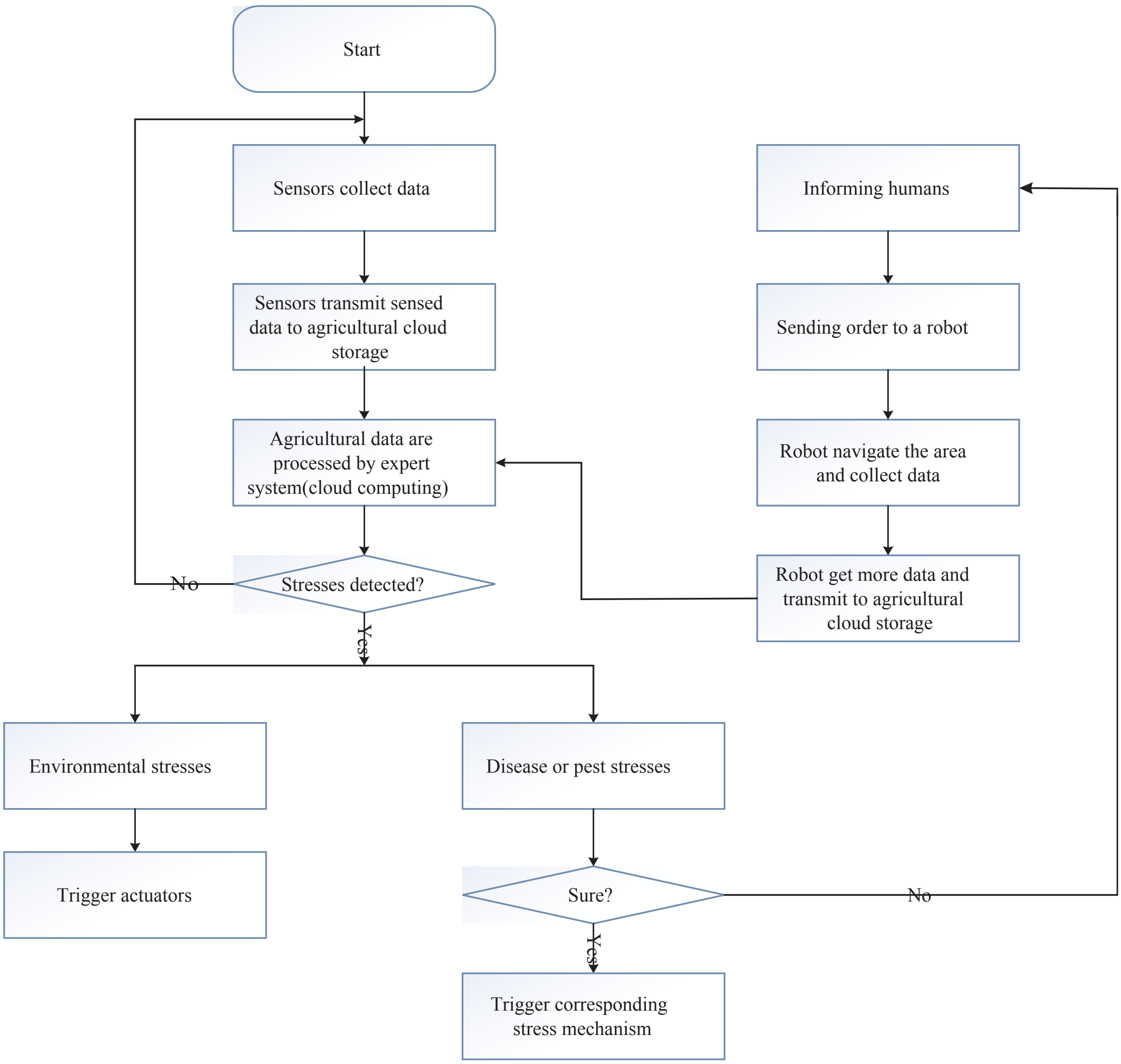

Fig. 4. MDR-CPS workflow chart.

WSN is good to collect agricultural data in 24 h a day without interrupt, however, the environmental parameters, such as temperature, humidity, water level and so on, the greenhouse can adjust them automatically. It is difficult to determine the plant health if only the data transmitted by WSN are relied on. Fig. 9 (Zhang, 2016) presents infrared thermal images vs visible light images. Thermal images are transmitted by most of sensors in agriculture monitor, while visible light images are taken by cameras. As the images shown, it is difficulty for expert system or humans to estimate plant healthy states or whether a kind of disease occurred, if they only count on thermal images. To attain detailed and rich correlative data to make an accurate analysis, it is necessary to dispatch a robot to detect marked samples. To evaluate and validate the design of MDR-CPS (H + R + S, marked as Scheme 1 in the next section), we analyze our model, and experimentally compare our scheme in two scenarios with alternative scheme (R + S, marked as Scheme 2 in the next section). As to WSN, we consider sensor functions are mounted on a robot, therefore, we no longer compare Scheme 1 and WSN.

4.1. Experimental design of the two schemes

To evaluate the new framework, computer simulation experiment is utilized to test the performances. The experiments are designed, implemented by simulation to examine and compare performance indicators, and outcomes are measured. Two schemes (Scheme 1. H + R + S and Scheme 2. R + S) are described in this part. The explanation of two schemes are as below.

Scheme 1. H + R + S: In this model, sensors are equipped on a robot, and the robot reach a plant to detect stresses. If sensors detect the infected plants, the robot will send signals to humans who may command the robot to measure more metrics. It will be delaying time since the human needs to decide for some unclear pictures with low quality, and the robot needs to detect again. The operation time of this scheme are transportation time (t1), localization time (t2), time for obtaining 3D image of a target (t3), time for stretching out one arm along trajectory (t4), time for touching the target and sensing (t5), and time for Human making decisions (t6). At each operation, it might have some conflicts or errors (C&E) that make the robot redo each specific task as described in the previous section. Scheme 1 is demonstrated in Fig. 10. A robot starts from Human-Robot Based Point, using time t1 to visit location 1. The robot mounted with sensors performs the task to measure parameters at the location by using time t2, t3, t4, and t5. After that, the robot sends signals to humans at the human-robot based point. The human can decide whether parameters measured are satisfied or

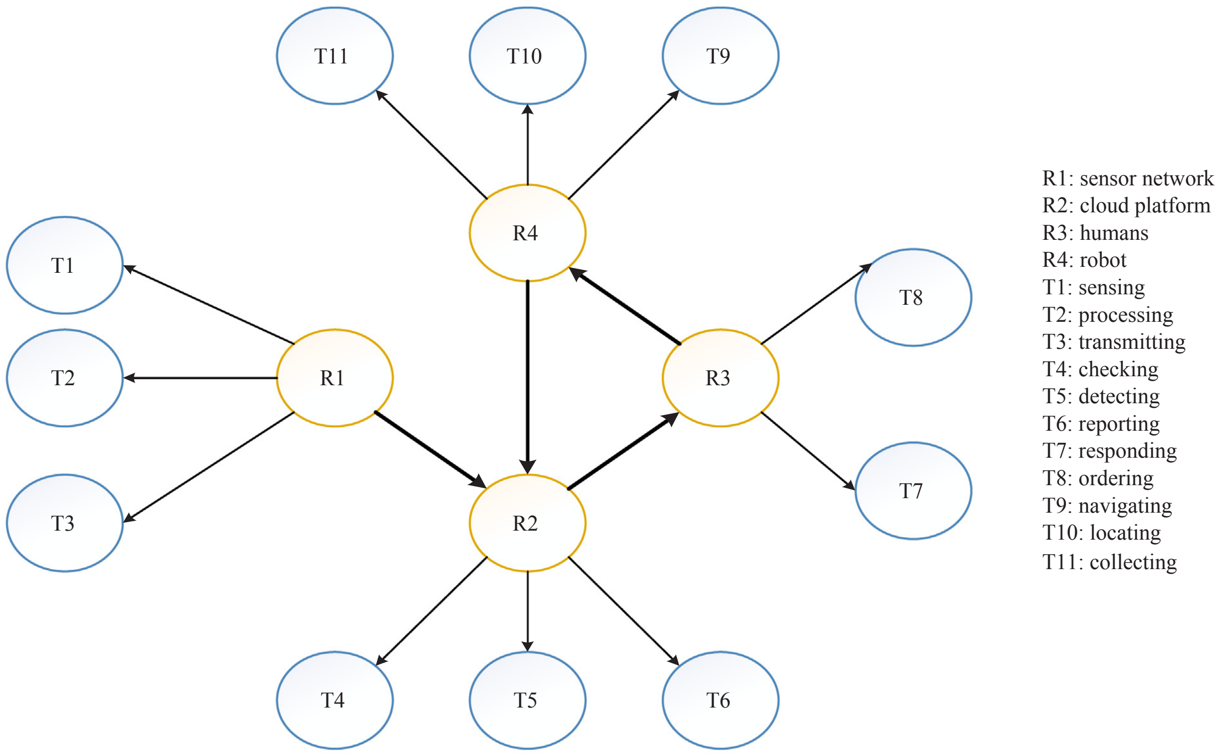

Fig. 5. The tasks matching the resources in MDR-CPS.

not, which costs t6. If data are not clear enough, the robot will measure again. Otherwise, the robot will move to the next location till visits all assigned locations.

Scheme 1. **R + S**: In this model, the robot reaches a crop to detect stresses and there is no immediate response from the human to make sure whether the quality of the data is clear or not. Therefore, after the robot arrived at robot hub, if base point needs more information about a sample, the robot must travel again to obtain the data. The operation time of this scheme is as same as the Scheme 1 except t6 that it is not concluded. However, transportation time (t1), localization time (t2), time for obtaining 3D image of a target (t3), time for stretching out one arm along trajectory (t4), and time for touching the target and sensing (t5) might need to perform more than once (t1′, t2′, t3′, t4′, and t5′) since at each operation, it may also have some conflicts or errors that make a robot need to redo each specific task. Scheme 2 is demonstrated in Fig. 11. A robot starts from human-robot based point, using time t1 to

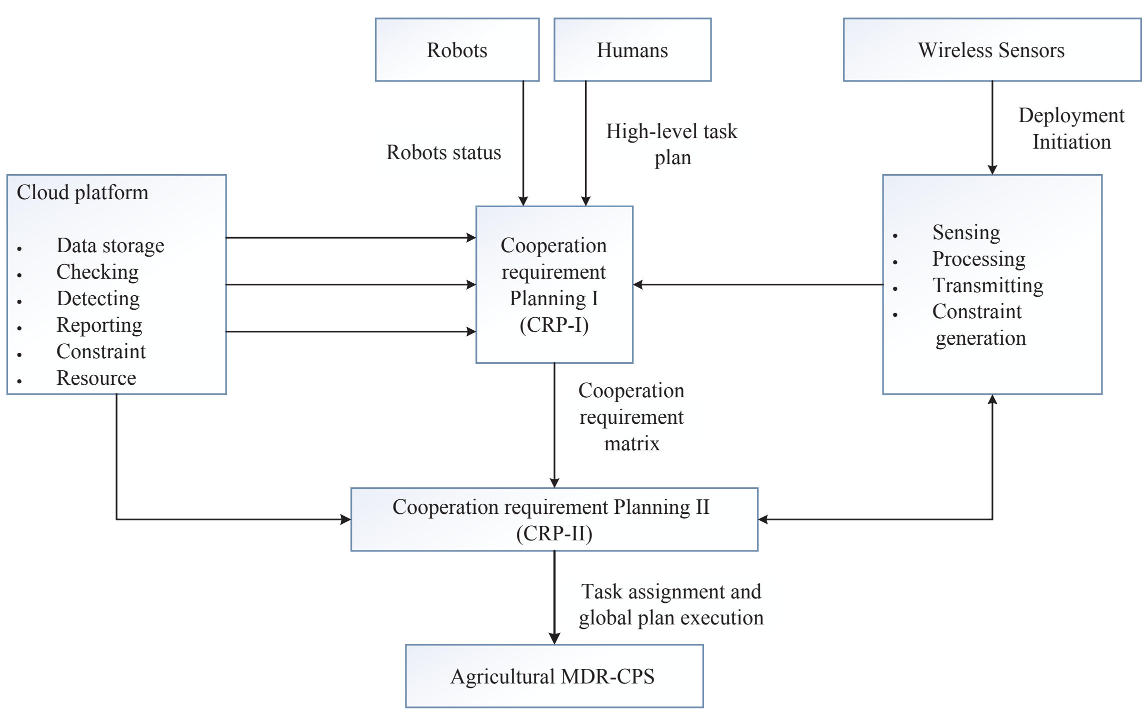

Fig. 6. CRP architecture for agricultural MDR-CPS.

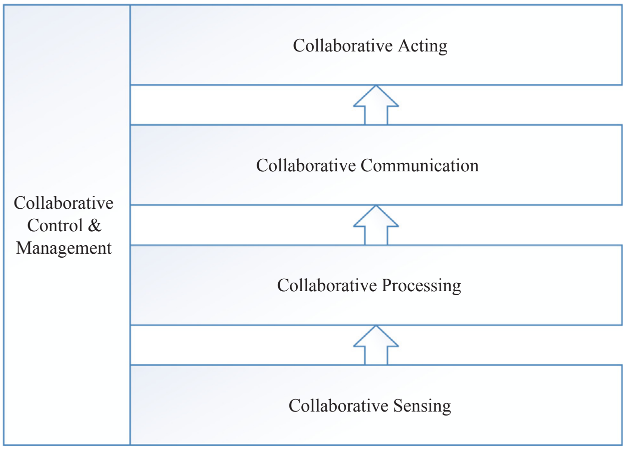

Fig. 7. Collaborative Architecture for MDR-CPS.

visit location 1. A robot and sensors perform the task by using time t2, t3, t4, and t5. Then, a robot moves to the next location without sending the signal back to human-robot based point. After visiting all locations, a robot goes back to the based point and if data are not satisfied, a robot needs to travel to particular locations again (orange line in Fig. 11) with t1′ and performs the task with t2′, t3′, t4′, and t5′.

In both schemes, a robot's motions include t1, t2, t3, t4, and t5. The process for a robot sampling a target is shown as Fig. 12. The list of motions that a robot finishes detecting once sample and their explanations are presented in Table 5.

4.2. Experimental simulation

We conduct three experiments to measure and compare performance indicator. The three experiments are described in Table 6 and simulation assumptions are shown in Table 7 in which the value or the reasonable interval range of a robot operations' time follows (Ji, 2014).

4.2.1. Experiment 1 – Measuring detection time under different schemes

4.2.1.1. Experimental parameters. To test these two schemes, computer simulation experiment is conducted to compare the detection time. With the same targets, map, parameters and resources, two schemes also have the same objective which is to visit every assigned location and measure the target, however, the only different between them is whether humans are involved or not. Time to perform the task is an indicator in this experiment. The scheme that can perform the same task with less time would be preferable. The parameters in the simulation experiment are shown in Table 7.

4.2.1.2. Result and analysis. The result of the 100 replications of simulation is presented in Fig. 13. Table 8 is summarized the measurements from the simulation experiment. At the first observation, the average operating time for both schemes is not clear different, however, the standard deviation (SD) of the Scheme 2 is significantly higher. Fig. 13 is also showed the same characteristic. To compare both schemes' operation time whether two means are statistically different, *t*-test is conducted. The *t*-test in Table 9 indicating the null hypothesis of the two schemes with the same operation time is rejected at the significant level of 0.005. Therefore, Scheme 1 which has less operation time is preferable to Scheme 2.

4.2.2. Experiment 2 – Comparing tolerant performance of conflicts and errors under different schemes

4.2.2.1. Experimental parameters. This experiment aims to compare two schemes with the different probability of Conflicts and Errors (CEs). The better scheme should be superior to tolerant the change by not increasing significantly operation time. To test these two schemes,

Table 3
Examples of potential errors and conflicts, and countermeasures in MDR-CPS.

Cells	Errors	Conflicts	Policies	Detailed examples
Sensors	Nodes failure; nodes compromised; nodes power off because of battery exhaustion; nodes damage because of physical reasons (lighting or hurricane).	Transmission conflicts between nodes; synchronized conflicts between nodes; conflict data sensing between sensors and robot.	Detecting agent deployed; isolation policy for the failure nodes; propagation of a node failure to WSN; security mechanisms to protect WSN.	Physical protection; key management; node authentication; secure routing between sensor nodes; encryption; intrusion detection and so on.
Robots	Hardware failure; power off because of battery exhaustion; unexpected accidents, such as an eagle crashing with a robot.	Communication conflicts between navigating and location; acting conflicts between navigation and sensing by sensors on a robot.	Precise agents for robots deployed; detecting, propagating, error recovery, conflict resolution mechanisms to be operated to robots.	Secure design; secure assessments; secure standard; secure detection.
Humans	Wrong judge; wrong order; out of duty etc.	Conflict orders made for robots; conflicts with human activities.	Detection, identification, exception handling.	Expert training; agricultural experience gaining.
Cloud platform	Hardware failure; Wrong analysis because of software failure; wrong results transmitted; wrong decision made.	Software (agents) conflicts; control conflicts.	Hardware and software security protection; detection, diagnostics, error recovery and conflict resolution policies.	Access control; data encryption; identity authentication; key management; architecture and framework security; portability, interoperability and application security; consensus assessments; legal protection.

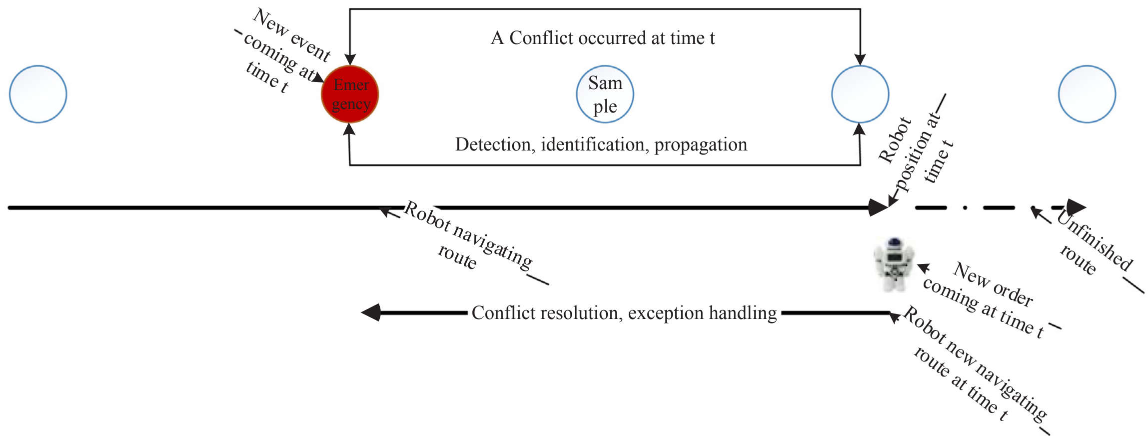

Fig. 8. Robot conflict propagation and the resolution process.

Table 4
Comparison of different monitoring schemes.

Monitoring type	Monitoring mechanisms		
	H + R + S	R + S	WSN
Environmental parameters	Controlled by greenhouse	Controlled by greenhouse	Controlled by greenhouse
Soil parameters	Sensors equipped on a mobile robot	Sensors equipped on a mobile robot	Stationary sensors
Crops growth parameters	Sensors equipped on a mobile robot	Sensors equipped on mobile robots	Stationary sensors
Crops diseases or Intruders	A mobile robot with sensors and HD Cameras	A mobile robot with sensors and HD Cameras	Stationary sensors
Human involvement	Y	N	N
Emergency responded mode	Emergency is responded with high priority by humans	Emergency cannot be responded until the robot carries out next pass navigation	Emergency cannot be responded

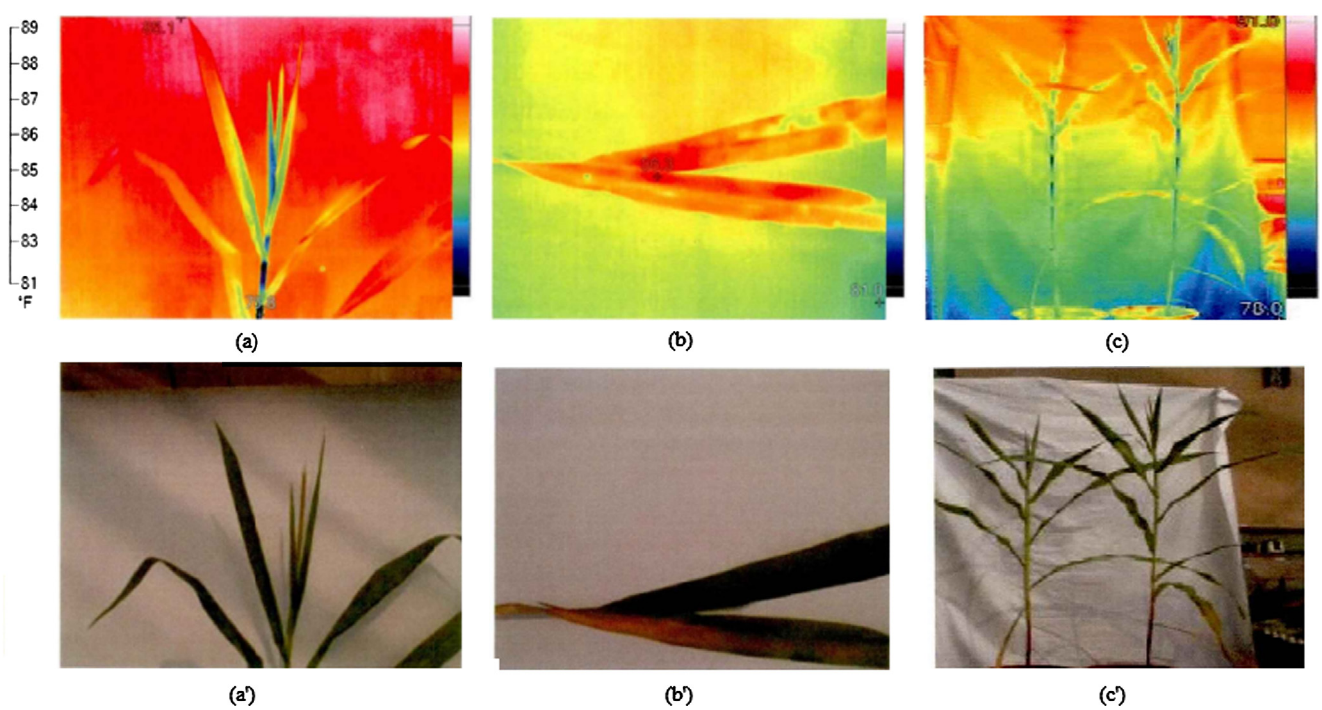

Fig. 9. Corn leaves images ((a) infrared thermal image of normal corn leaf vs. (a′) visible light image of normal corn leaf; (b) infrared thermal image of disease corn leaf vs. (b′) visible light image of disease corn leaf; and (c) infrared thermal image of two unhealthy and entire corn plants vs. visible light image of two unhealthy and entire corn plants).

two schemes described in the Table 6 are implemented by computer simulation. Similarly, the operation time is a crucial reference. The parameters in the simulation experiment as well as the two schemes are shown in Table 7. In the simulation, we start from 0% (no CEs at all) probability of CEs.

4.2.2.2. Result and analysis. From the 100 replications of simulation, the results are shown in Fig. 14. The statistics from the experiment is

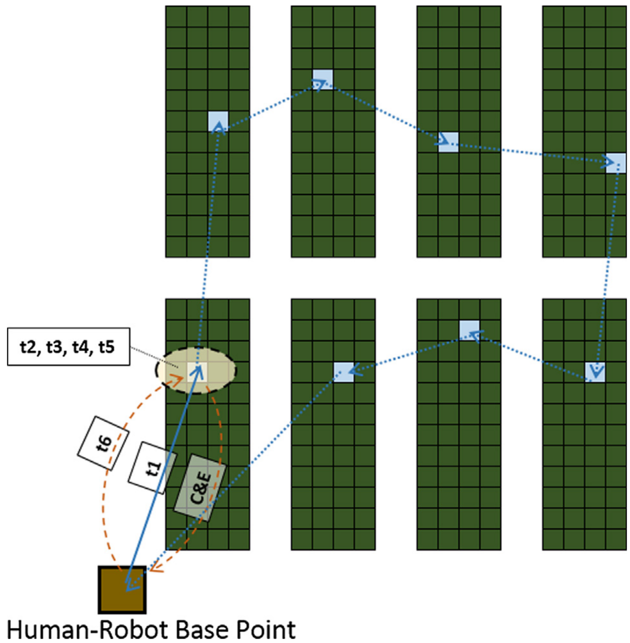

Fig. 10. Scheme 1 (H + R + S) assumption.

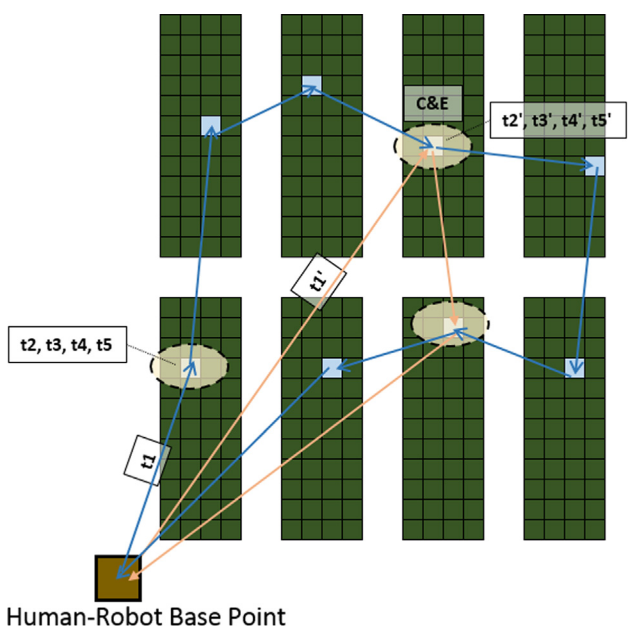

Fig. 11. Scheme 2 (R + S) assumption.

summarized in Table 10. The average operation time for both schemes is increased when the probability of CEs is increased, however, operation time of Scheme 2 rises significantly faster than Scheme 1. The standard deviation shows the same trend. With the higher CE rates, standard deviation tends to be higher and Scheme 2 has the higher increasing rate of standard error than Scheme 1. Total operation time of Scheme 1 and 2 is compared in Fig. 14. Obviously, Scheme 1 has the slower rate of increasing operation time when percentage of CEs is increased.

4.2.3. Experiment 3 – Comparing emergency response under two schemes

4.2.3.1. Experiment design and parameters. We assume an emergency occurred, for instance, a burst of plant disease. Remote humans are urgent to know several important factors: (i) the detection of a disease at an early point in time, (ii) the separation of diseases caused by abiotic stresses, and (iii) the quantification of disease severity (Mahlein, 2016), because site-specific and targeted applications of pesticides according to precision crop protection strategies results in potential reduction in pesticide use and can thus reduce the economic expense and ecological impact in agricultural crop production systems (Gebbers and Adamchuk, 2010). Humans dispatch a robot to detect that doubtful area. The robot needs to confirm the infected plant coverage as soon as possible. It is rapid that the robot is indicated by humans instead of detecting plants one by one. Scheme 1 is developed an algorithm to help the robot to lock the infected area, while the robot in Scheme 2 must detect the possible infected plant one by one or one per several samples, which means the robot in Scheme 2 lack of self-adaptability. Fig. 15 presents the procedure. The robot finds out specious data when it carries on regular navigation, and delivers data to remote humans. Humans analyze specious data, determine step x according to data attributes (for instance, if yellow rust appears on a kind of plant leaves, humans estimate possible infected range according to their experience), and deliver the next sample's coordinate to the robot. If sample $i + x$ has been infected, humans indicate another step y, delivering to the robot. The robot repeats algorithm of *searching infected range*.

The parameters in the simulation experiment are shown in Table 7.

4.2.3.2. Result and analysis. Fig. 16 presents time cost of the two schemes. To compare fairly, we only calculate time cost that the two schemes confirm the infected area. Scheme 1 (H + R + S) shows obvious advantages comparing with Scheme 2 (R + S), because humans guide the robot to search potential infected samples in an efficient manner instead of searching one by one (R + S in Fig. 16) and searching one per 5 samples ((R + S)-5 in Fig. 16) as the manner conducted by Scheme 2.

Agricultural robots are not smart enough because unlike industrial applications which deal with relatively simple, repetitive, well-defined and predetermined tasks in stable and replicable environments, agricultural applications for automation and robotics require advanced technologies to deal with complex and highly variable environments and produce (Bechar et al., 2003). Furthermore, agricultural production deals with plants or crops more sensitive to environmental and physical conditions. For instance, the same plant disease bursting in this year might express different symptoms with last year's due to changes of environment, transmission mode or photos taken by robots under different angles or degree of bright and dark light. This characteristic makes the replacement of human ability by machines or automation extremely difficult. Therefore, stresses diagnosing, harvesting, sorting and packaging, are still performed manually.

Though there are those difficulties to develop totally automatic R + S system to diagnose plant disease and take appropriate action without humans aids, we go further to summarize the agricultural robots trained by deep learning. Humans input pictures of the leaves of the sick crops and the healthy crops on the computer and identify which crops are healthy by the deep learning algorithm. 26 diseases of 14 crops through deep learning algorithms can be detected by Hughes and Salathé (2016) . They import more than 50 thousands photos of plant leaves into computers and run corresponding deep learning algorithms. The accuracy of the final procedure for correctly identifying crop diseases is as high as 99.35% (Mohanty et al., 2016). But the foundation for that is to take photos of plants under bright light conditions and standard backgrounds. If the photos of plant leaves are randomly selected on the Internet, the accuracy rate will be reduced to 30% to 40% (Mohanty et al., 2016). Needless to say, the low rate of photos taking by a robot with different angles or blurred light can be recognized as a kind of disease. Misdiagnosis of crops causes farmers to abuse pesticides and herbicides, which is a waste of time and money. Ideally, robots with enough artificial intelligence can help farmers quickly and accurately identify the stresses, but the final decision should be combined with the field investigation by human experts. One of our future work is to establish sufficient data set for one or two crop diseases to train

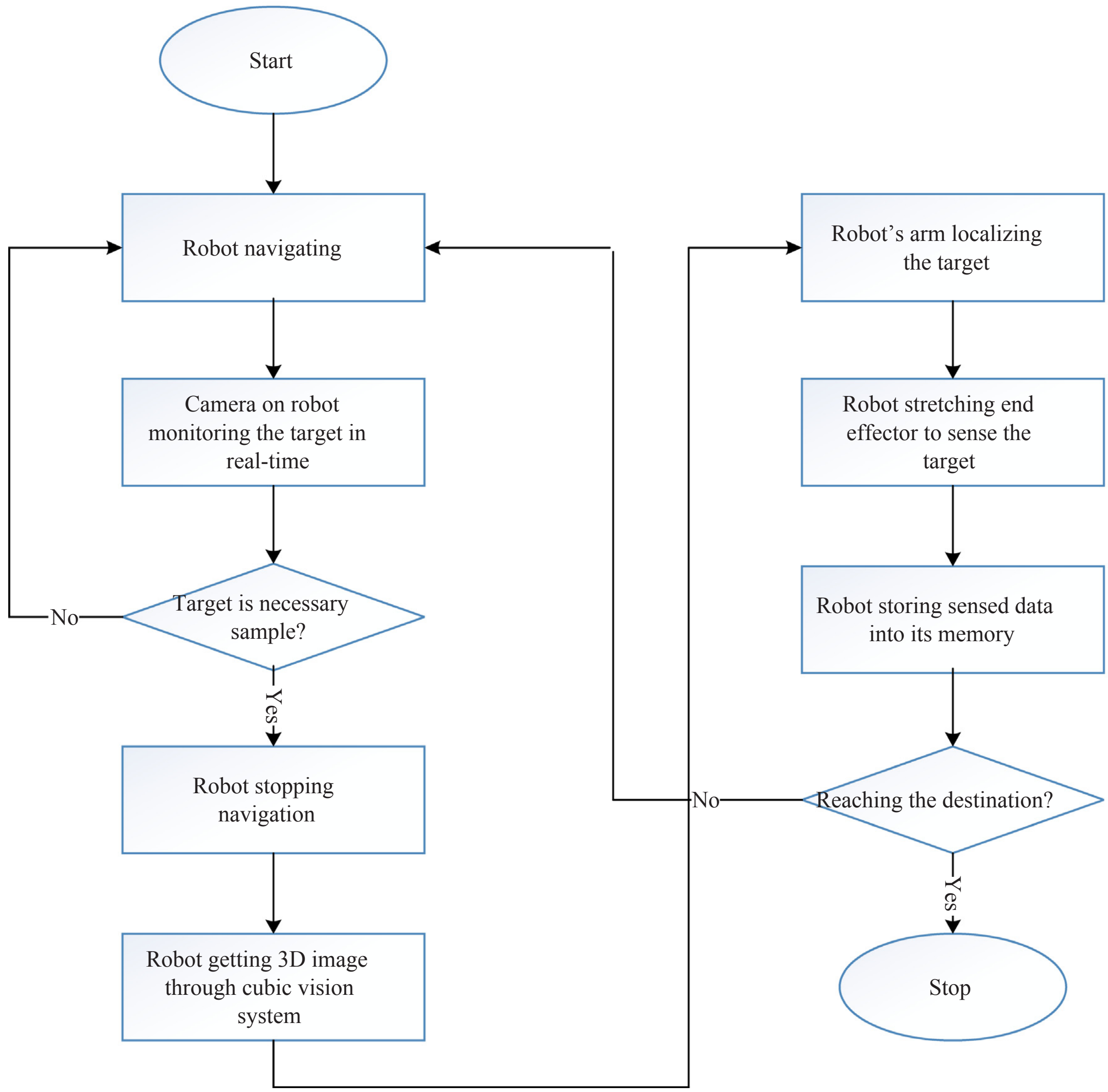

Fig. 12. Flowchart of a robot detecting motion.

Table 5
Motions and Time cost representations for a robot finishing once sample.

No	Robot motion	Time cost (Second)	Note
1	Navigating	l/v = t1	*l* is the distance between two adjacent samples, and *v* is a robot walking speed
2	Localizing	t2	Time for a robot localizing an area
3	Obtaining 3D image of a target	t3	Time for a robot figuring out a target
4	Stretching out one arm along trajectory	t4	Time for a robot stretching out its arm to touch the target
5	Touching the target and sensing	t5	Time for a robot sensing the target
6	Decision time from Human	t6	Time for human to make decision

robots to recognize a kind of disease of that crop and then automatically take appropriate action.

5. Conclusion

In this paper, a framework, workflow and application of collaborative control theory for an agricultural greenhouse are established, in which we associate sensors, a robot, and humans as an agricultural CPS focused on collaborative monitoring, detecting and responding stresses. We firstly design a monitoring-detecting-responding CPS framework for the greenhouse. A wireless sensor network collects environmental parameters, a robot equipped with special sensors senses data, such as leave's humidity, leave's temperature, and fruits sallow, and humans involve in the activities, for instance, ordering a robot to carry on an emergent check of plants, and responding to the robot quickly. Secondly, we develop a workflow based on MDR-CPS framework, which provides a detailed description of how MDR-CPS work. Finally, we apply collaborative control theory to MDR-CPS, deploying CRP-I (Collaborative Requirement Planning) and CRP-II, analyzing CEs (conflicts and errors) might occurred in MDR-CPS, and putting forward collaborative architecture for MDR-CPS.

Validation of the model for MDR-CPS is performed through computer simulation experiments. In term of observation and analysis in the experimental studies, we conclude the following. (i) Our model for the regular detection (environmental parameters, such as temperature, humidity, and solar light) cannot show obvious advantage, however, involving humans in the system will allow the immediate response which can save time to perform the given task dramatically. In addition, response timely from human can stabilize the operation time instead of

Table 6
Description of experiments.

Experiment objective	Description	Monitoring mechanisms	
		H + R + S	R + S
Experiment 1: Comparison operation time	Monitoring crops diseases, such as bacteria, fungi, viruses on the leave or monitoring some intruders, such as bugs, worms, rots etc. with the same conflict and error rate. (CEs' rate = 10%)	In our model, we assume a robot reach the crop to detect its diseases. A robot takes photos and transmits them to humans for making decisions of moving to the next location or not.	In this model, sensors are equipped on a robot. The robot reaches a crop to detect and take photos, and moves to the next location. After visiting all locations, if some photos are blurred, the robot must revisit the location.
Experiment 2: Comparison tolerance of the conflicts and errors rate	The difference from Experiment 1 is CEs' rate changes from 0% to 90%.	Based on Experiment 1, the different of conflicts and errors rate can influence numbers of information transmitted.	Based on Experiment 1, CEs' rate in each operation is changed and indicates the number of routing that a robot needs to revisit the locations.
Experiment 3: Comparison emergency response time	The difference from Experiment 1 and 2 is that emergency might occur with probability *p*, for instance, a burst of plant disease.	If a robot transmits specious data to indicate a potential disease spread, humans could respond immediately and guide the robot to confirm the possibly infected area as soon as early.	In this model, a robot works as the preset routine without communications with humans during the detection, therefore it cannot obtain any aid from humans when it encounters emergencies.

Table 7
Simulation assumptions.

	Simulation assumptions
Common assumptions for three experiments	(1) Operation time 1 (transportation time: t1): distance between each location is assumed 10 m and robot travel speed is 0.7 m/s. (2) Operation time 2 (localization time: t2): N (6, 3) seconds (3) Operation time 3 (time for obtaining 3D image of a target: t3): N (4, 1) seconds (4) Operation time 4 (time for stretching out one arm along trajectory: t4): N (17, 5) seconds (5) Operation time 5 (time for touching the target and sensing: t5): N (4, 1) seconds (6) Operation time 6 (decision time for Human: t6): N (45,15) seconds
Different assumption only for Experiment 1	(7) Conflict and Error rate = 10% at each operation shown in table 5
Different assumption only for Experiment 2	(7) Conflict and Error rate = the probability of CEs from 0% to 90%
Different assumptions only for Experiment 3	(7) Emergency occurred probability: B (100, p), p is decided by random function (8) x and y (detection step) are decided by binary searching strategy when looking for the front or back of the detected sample for determining infected target or not.

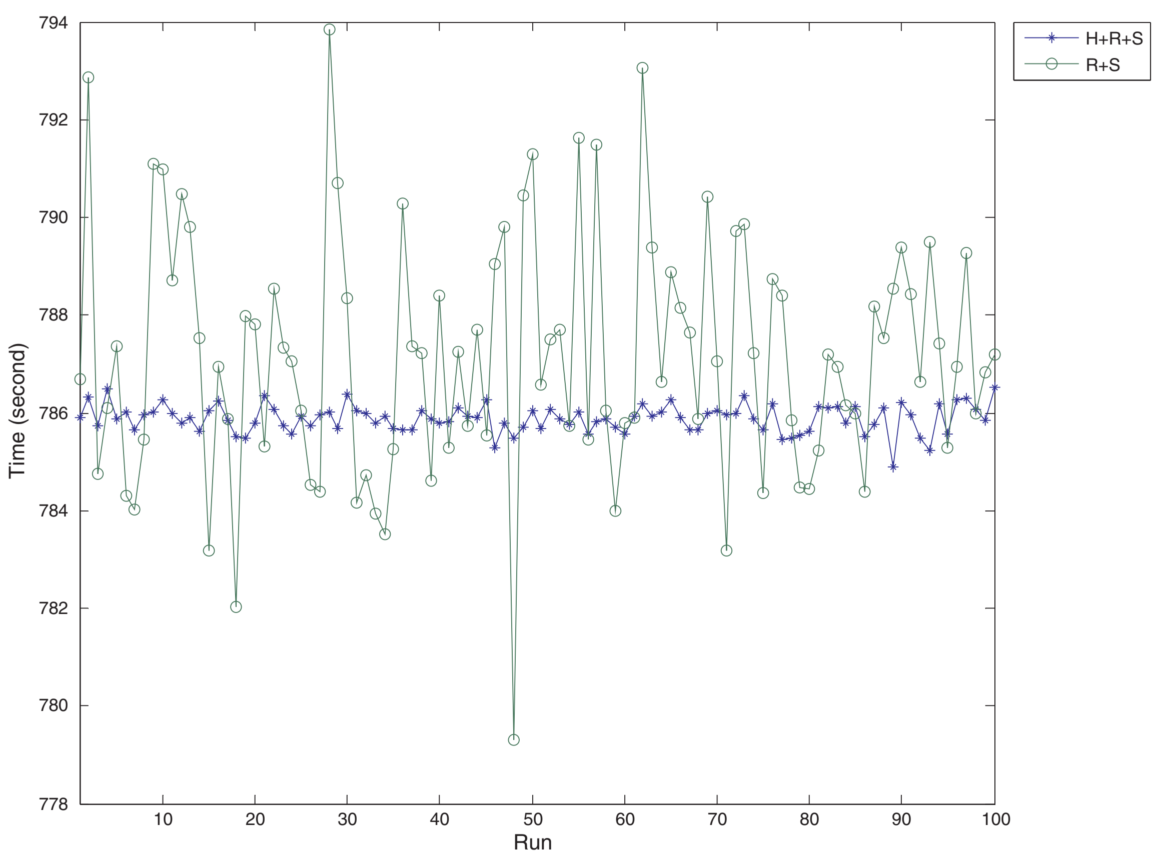

Fig. 13. Experiment 1-detection time difference of Scheme 1 vs Scheme 2.

Table 8
Results of Experiment 1.

	Scheme 1 (H + R + S)	Scheme 2 (R + S)
Average operating time (second)	785.88	787.15
SD of operation time (second)	0.28	2.48

Table 9
Hypothesis testing for Experiment 1.

H0 Hypothesis	P value
Average operating time for Schemes 1 and 2 is the same.	< 0.005*

*At the significant level 0.005, based on P values of less than 0.005, null hypothesis shown is rejected.

Table 10
Results of Experiment 2.

Probability of CEs	Scheme 1 average operating time (Second)	Scheme 2 average operating time (Second)	Scheme 1 SD (Second)	Scheme 2 SD (Second)
0	777.45	744.40	1.89	0.98
0.1	785.60	833.91	3.34	35.91
0.2	795.37	927.35	4.24	58.22
0.3	809.03	1080.14	5.36	70.23
0.4	827.12	1261.97	7.36	108.13
0.5	852.22	1495.88	10.65	141.09
0.6	891.16	1942.06	10.97	211.05
0.7	953.54	2567.85	19.99	275.25
0.8	1069.58	3844.81	24.70	456.52
0.9	1448.91	7764.73	55.64	1026.42

fluctuating as the scheme without human involved. (ii) With the increasing rates of conflicts and errors, our model shows higher fault tolerance capability. Collaboration between three parties increases ability to endure with the different rate of conflicts and errors by not rising operation time and its standard deviation drastically. (iii) As the emergency response, our scheme shows distinct difference of efficiency and time cost from the compared Scheme 2, since human involvement accounts for a large proportion. Though communicative traffic might increase between human and robot, our experiments and analysis show that collaboration between robot and human is superior to no collaboration between them based on our three designed experiments which are common applications or situations in agricultural greenhouse.

Future research can focus on three important directions:

(1) Research on deep learning and big data to develop smart algorithms for MDP-CPS to support robots to deal with more complicated situations in greenhouse.
(2) Expand CEs detection and prevention work to design various conditions for different agents and identify the optimal timing of error prevention.
(3) Consider the cyber security for MDP-CPS, especially the secure communication between sensors, robots and humans.

Acknowledgment

Research reported in this article was supported by the PRISM Center for Production, Robotics, and Integration Software for Manufacturing and Management at Purdue University. The first author was funded by Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and Nanjing University of Information Science and Technology, China, partially by National Natural Science Foundation of China Grant 61375030. The second author was funded partially by BARD Grant IS-4886-16R for "Development of a robotic inspection system for early identification and locating of biotic and abiotic stresses in greenhouse crops." The authors also acknowledge Dr. Avital Bechar, Volcani Agricultural Research Organization, Israel; and Dr. Yang Tao, University of Maryland, Co-PIs on this BARD project, for their contributions to the development of MDR-CPS.

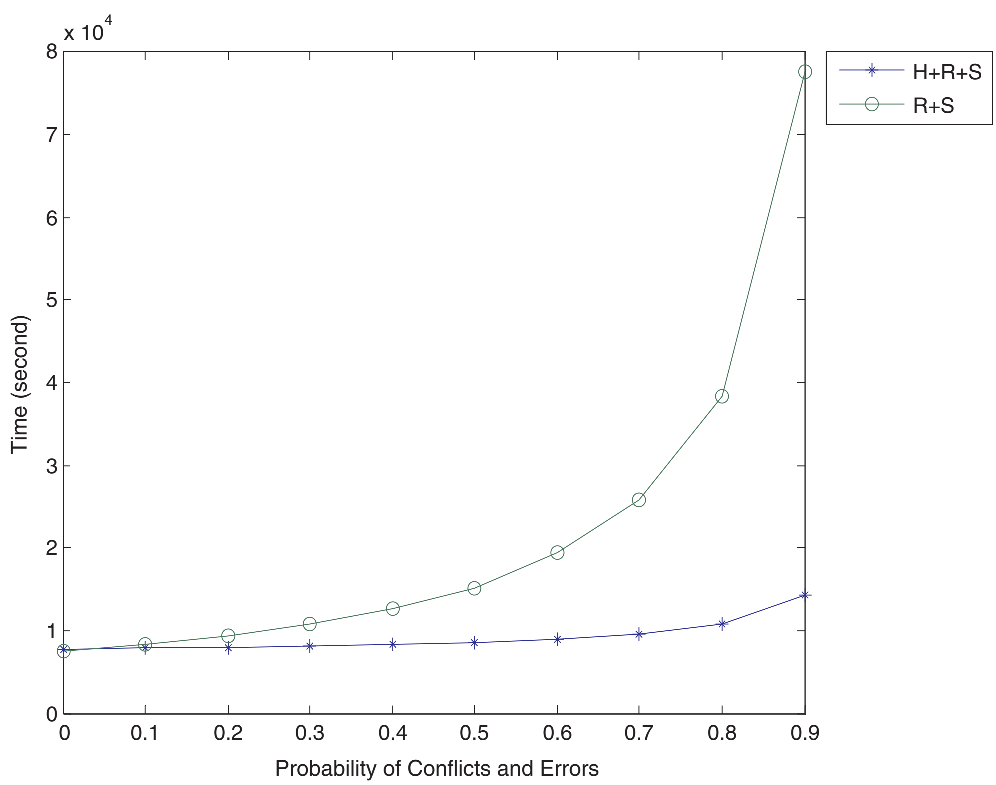

Fig. 14. Experiment 2-detection time difference of Scheme 1 vs Scheme 2.

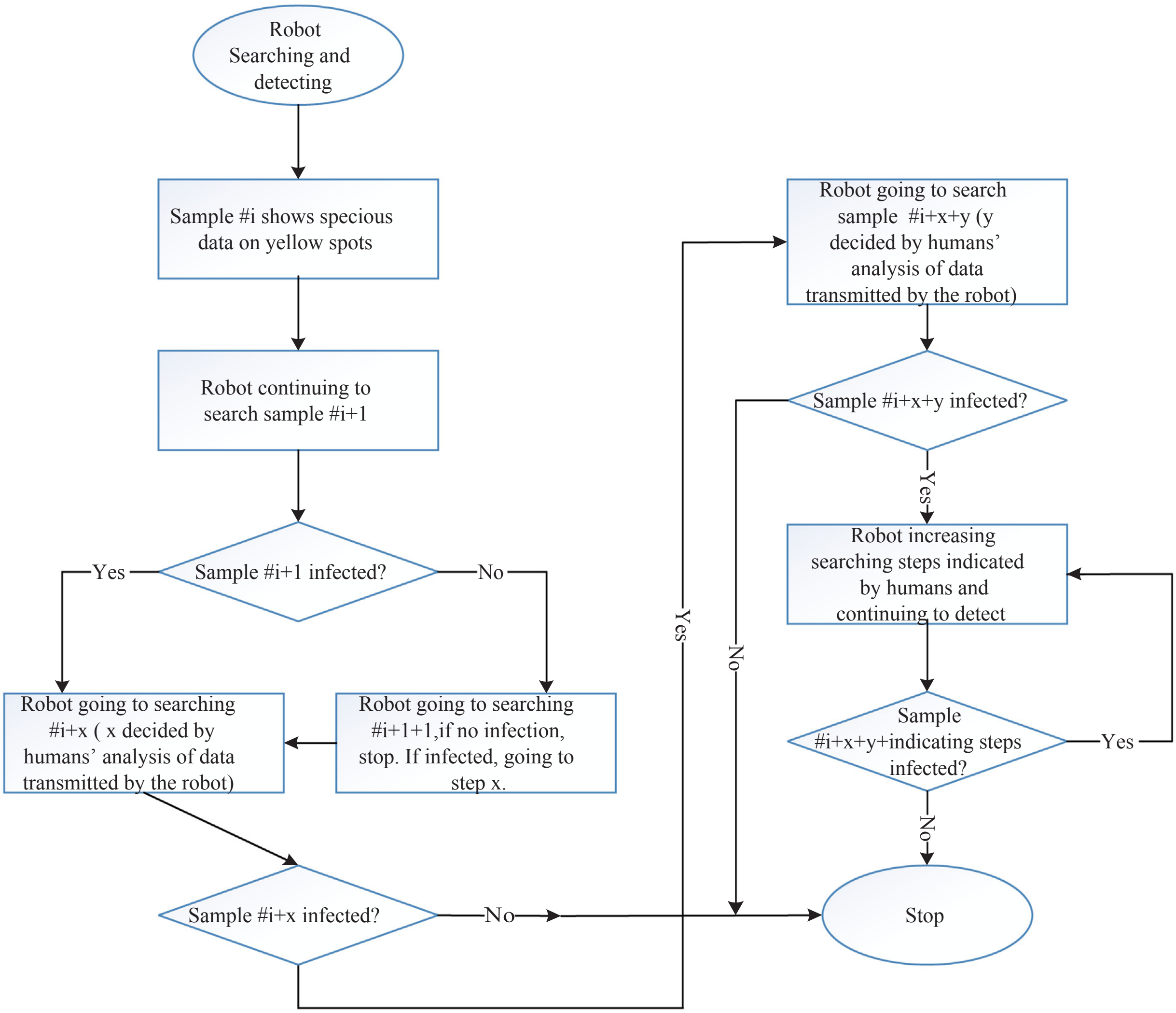

Fig. 15. An example of the robot searching infected plant coverage guided by humans.

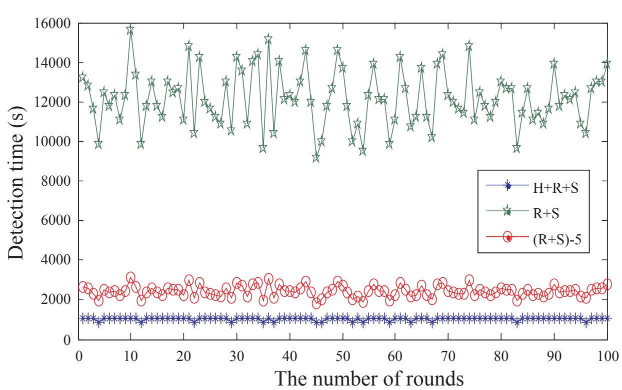

Fig. 16. Experiment 3-detection time difference of Scheme 1 vs Scheme 2.

References

Ari, A.A., Gueroui, A., Labraoui, N., Yenke, B.O., 2015. Concepts and evolution of research in the field of wireless sensor networks. Int. J. Comp. Netw. Commun. 7 (1), 81–98.

Astrand, B., Baerveldt, A.J., 2002. An agricultural mobile robot with vision-based perception for mechanical weed control. Auton. Robots 13, 21–35.

Bac, C. Wouter, Henten, E.J.V., Hemming, J., Edan, Y., 2014. Harvesting robots for high-value crops: state-of-the-art review and challenges ahead. J. Field Robot. 31 (6), 888–911.

Bechar, A., Edan, Y., 2003. Human-robot collaboration for improved target recognition of agricultural robots. Indust. Robot: Int. J. 30 (5), 432–436.

Bechar, A., Meyer, J., Edan, J., 2009. An objective function to evaluate performance of human–robot collaboration in target recognition tasks. IEEE Trans. Syst., Man and Cybernet.-Part C: Appl. Rev. 39 (6), 611–620.

Belforte, G., Gay, P., Aimonino, D.R., 2006. Robotics for improving quality, safety and productivity in intensive agriculture: challenges and opportunities. In: Kin Huat Low (Ed.), Industrial Robotics: Programming, Simulation and Applications, Advanced Robotic Systems. Vienna, Austria.

Blackmore, S., 1994b. Precision farming: an introduction. J. Outlook Agric. 23, 275–280.

Chen, X.W., Nof, S.Y., 2012b. Constraint-based conflict and error management. Eng. Optim. 44 (7), 821–841.

Chen, X.W., Nof, S.Y., 2012a. Agent-based error prevention algorithms. Experts Syst. Appl. 39, 280–287.

Chen, N.C., Zhang, X., Wang, C., 2015. Integrated open geospatial web service enabled cyber-physical information infrastructure for precision agriculture monitoring. Comput. Electron. Agric. 111, 78–91.

Gay, P., Piccarolo, P., Aimonino, D., 2008. Robotics for work and environment safety in greenhouse. In: International Conference on Innovation Technology to Empower Safety, Health and Welfare in Agriculture and Agro-food Systems.

Gebbers, R., Adamchuk, V.I., 2010. Precision agriculture and food security. Science 327, 828–831.

Gongal, A., Amatya, S., Karnee, M., 2015. Sensors and systems for fruits detection and location: a review. Comput. Electron. Agric. 116, 8–19.

Hoshi, T., Ochi, S., Masuyama, T., Yasuba, K.I., Kurosaki, H., Okayasu, T., 2016. Weatherability evaluation of low-cost relative humidity sensors for use in greenhouse environments. Agric. Inf. Res. 25 (3), 79–85.

Ishibashi, M., Iida, M., Suguri, M., Masuda, R., 2013. Remote monitoring of agricultural robot using web application. In: 4th IFAC Conference on Modeling and Control in Agriculture, Horticulture and Post Harvest Industry, pp. 27–30.

Ji, C., 2014. Vision Information Acquisition for Fruit Harvesting Robot and Development of Robot Prototype System, Ph.D Dissertation. China agricultural University, pp. 72–78 (in Chinese with English abstract).

Keicher, R., Seufert, H., 2000. Automatic guidance for agricultural vehicles in Europe.

Comput. Electron. Agric. 25, 169–194.
Ko, H.S., Nof, S.Y., 2012. Design and application of task administration protocols for collaborative production and service systems. Int. J. Product. Econ. 135 (1), 177–189.
Ko, M.H., Ryuh, B.S., Kim, K.C., Suprem, A., Mahalik, N.P., 2015b. Autonomous greenhouse mobile robot driving strategies from system integration perspective: review and application. IEEE/ASME Trans. Mechatron. 20 (4), 1705–1716.
Ko, M.H., Ryuh, B.H., Kim, K.C., 2015a. Autonomous greenhouse mobile robot driving strategies from system integration perspective: review and application. IEEE/ASME Trans. Mechatron. 20 (4), 1705–1716.
Kumar, A., Singh, A., Singh, I.P., Sud, S.K., 2010a. Prototype greenhouse environment monitoring system. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, pp. 218–229.
Kumar, A., Singh, A., Singh, I.P., Sud, S.K., 2010b. Greenhouse environment monitoring system. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, pp. 17–19.
Mahlein, A.K., 2016. Plant disease detection by imaging sensors-parallels and specific demands for precision agriculture and plant phenotyping. Plant Dis. 2, 241–251.
Mcintosh, P., 2012. Agricultural Robotics: Here Come the Agribots < http://www.maximumyield.com > .
Moghaddam, M., Nof, S.Y., 2014. Combined demand and capacity sharing with best matching decisions in enterprise collaboration. Int. J. Product. Econ. 148, 93–109.
Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7. http://dx.doi.org/10.3389/fpls.2016.01419.
Nagasaka, Y., Zhang, Q., Grift, T.E., Kanetani, Y., Umeda, N., Kokuryu, T., 2004. An autonomous field watching-dog robot for information collection in agricultural fields. In: ASAE Annual International Meeting, pp. 4215–4222.
Nayak, A., Reyes Levalle, R., Lee, S.K., Nof, S.Y., 2016. Resource sharing in cyber-physical systems: modeling framework and case studies. Int. J. Product. Res. 54 (23), 1–15.
Nie, J., Sun, R.J., Li, X.H., 2014. A precision agriculture architecture with cyber-physical systems design technology. Appl. Mech. Mater. 543 (547), 1567–1570.
Nikolidakis, S.A., Kandris, D., Vergados, D.D., Douligeris, C., 2015. Energy efficient automated control of irrigation in agriculture by using wireless sensor networks. Comput. Electron. Agric. 113, 154–163.
Nof, S.Y., 2007. Collaborative control theory for e-Work, e-Production, and e-Service. Annu. Rev. Control 31 (2), 281–292.
Nof, Y.S., Ceroni, J., Jeong, W., Moghaddam, M., 2015. Revolutionizing collaboration through e-Work, e-Business, and e-Service. Springer Publishers, pp. 33–75.
Pedersen, S.M., Fountas., S., Have, H., Blackmore, B.S., 2005. Agricultural robots: an economic feasibility study. Prec. Agric., 589–596.
Reid, J.F., Zhang, Q., Noguchi, N., Dickson, M., 2000. Agricultural automatic guidance research in North America. Comput. Electron. Agric. 25, 155–167.
Stark, B., Rider, S., Chen, Y.Q., 2013. Optimal pest management by networked unmanned cropdusters in precision agriculture: a cyber-physical system approach. In: 2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems, pp. 20–22.
Wang, N., Zhang, N., Wang, M., 2006. Wireless sensors in agriculture and food industry: recent development and future perspective. Comput. Electron. Agric. 50, 1–14.
Wark, T., Corke, P., Sikka, P., Klingbeil, L., Guo, Y., Crossman, C., Valencia, P., Swain, D., Hurley, G.B., 2007. Transforming agriculture through pervasive wireless sensor networks. Perv. Comput. 6 (2), 50–56.
Zhang, M., 2016. Information Collection of the Plant Biological Characteristic Parameters and Growth Environment of Agriculture and Forestry, Ph.D Dissertation. China Agricultural University, pp. 46-49 (in Chinese with English abstract).
Zhong, H., Nof, S.Y., 2015. The dynamic lines of collaboration model: collaborative disruption response in cyber-physical systems. Comp. IE 87 (9), 370–382.
Zhong, H., Wachs, J.P., Nof, S.Y., 2013. A collaborative telerobotics network framework with hand gesture interface and conflict prevention. Int. J. Product. Res. 51 (15), 4443–4463.