

Collaborative Detection and Prevention of Errors and Conflicts in an Agricultural Robotic System

Praditya AJIDARMA^{1*}, Shimon Y. NOF²

¹ Bandung Institute of Technology, Jl. Ganesha 10, Bandung, 40132, Indonesia
ajidarmap@itb.ac.id (*Corresponding author)

² Purdue University PRISM Center and School of IE, 315 N. Grant St., West Lafayette, IN 47907, USA
nof@purdue.edu

Abstract: Agricultural robotic system (ARS) is a cyber-physical structure which consists of multiple collaborating agents with the objectives of monitoring, early detection, and responding within greenhouses. As a system with collaborating agents, ARS is prone to errors and conflicts. This research aims to develop a collaborative detection and prevention mechanism to process the sensor data, such that errors and conflicts in the system are prevented, or minimized. The scope and application of this research is limited to a controlled environment within the ARS. Two collaborative detection and prevention of errors and conflicts (CDPEC) algorithms are proposed, developed, illustrated, and validated in this study. The algorithms' effectiveness is measured in terms of Conflict and Error Prevention Ratio (CEPR). In terms of mean CEPR, the CDPEC 1 can reduce the potential errors and conflicts by 66.4% compared to the baseline scenario. Meanwhile, CDPEC 2 manages to reduce potential errors and conflicts by 86.9% on average. Between the two alternative algorithms, the performance of CDPEC 2 is 30.9% higher (better) compared to the one of CDPEC 1. Conclusions for the design of the cyber collaborative ARS architecture are observed.

Keywords: Collaborative Control Theory, Conflict and Error Prevention Algorithms, Cyber Collaborative Protocols, HUB-CI.

1. Introduction

Collaborative control theory (CCT) aims to support the effective design of collaborative e-Work, e-Business, and e-Service systems (Nof, 2007). While CCT improves the effective outcome of collaborative efforts, it must also overcome conflicts and errors, which disrupt performance and lead to ineffectiveness in the system (Nof, 2009). CCT has been developed, validated and implemented by researchers and engineers worldwide, for instance, in the field of decision support system (Filip et al., 2017; Nof, 2017; Zhong et al., 2016), agricultural cyber-physical system (Dusadeerungsikul, Nof, et al., 2019), components-service integration (Moghaddam & Nof, 2018), and the decentralization of manufacturing and service systems (Moghaddam et al., 2016). CCT operates with protocols in order to provide an efficient and effective platform for autonomous agents to interact and attain its objectives. With CCT, agents are able to automate and integrate a set of heterogeneous activities, components, and tasks in a network (Bruno & Antonelli, 2018; Nof, 2009).

In agriculture, a variety of collaborating agents enable the emergence of smart, automated system of precision agriculture. It becomes a farming management concept which observes, measures, and responds to inter and intra-field variability in crops. The system collects data from a heterogeneous network of devices, with formats and semantics of these data defined by a range of factors, including

sensor types and configurations of data loggers (Mcbratney et al., 2005). The automation within the agricultural system has increased agricultural productivity over the years (Zhang, 2013). The benefit of smart systems with automation and robotic applications has been tested in a controlled agricultural environment system (Xia et al., 2015), real-time monitored greenhouse (Baille et al., 2001), and a greenhouse system with a mobile multi-sensor platform (Bautista-Gallego et al., 2011). Furthermore, most agricultural-related production tasks, such as harvesting, sorting and packaging, are still performed manually by human labor as a fully automated agricultural system is more expensive (Bechar & Eben-Chaime, 2014). The complex and highly variable agricultural environments require advanced automation and robotics in order to be effective (Hiremath et al., 2014).

Agricultural robotic system (ARS) is proposed as a middle ground between a fully automated agricultural system and conventional farming. The system equips human operators with the necessary semi-automated analytical tools and machinery to perform agricultural tasks (Bechar & Vigneault, 2016; Bloch et al., 2017). Equipped with IoT wearables, human operators are able to track positions and viewpoints (Okayama & Miyawaki, 2013), promote working motion templates (Morio et al., 2016), and knowledge transfer between agricultural experts and trainees (Sharma et al., 2015). Furthermore, human operators are better

equipped with personal environmental sensors, capable of conducting real-time and in-situ electrochemical analysis (Zhao et al., 2020), tracking crops' transpiration and field humidity (Lan et al., 2020). This collaboration of IoT wearables on human agent and robotic sensors enables ARS as a Cyber-Physical Structure to perform better in terms of Monitoring, Detection, and Responding (MDR-CPS), specifically in terms of emergency response (Guo et al., 2018).

Research in precision agriculture has proven that mobile robots are capable of various agricultural tasks, e.g., vision-based harvesting (van Henten et al., 2002), pursuing a trajectory path as either a cooperating group or an individual entity (Li et al., 2015), and performing various complex tasks (Haibo et al., 2015). Only limited research, however, has addressed potential errors and conflicts originated from such agricultural tasks. ARS mobile robots may fail to follow and fulfill its designated inspection route. Furthermore, it may also present measurements which are in contradiction to those of human operators, which leads to inconclusive decision-making. Based on the definitions provided by Nof (2009), the former is classified as error and the latter is classified as conflict. Prevention of error and conflict is crucial for any collaborative framework of agents where multiple participants interact.

Protocols regarding error and conflict prevention have been established in complex systems with distributed agents (Chen & Nof, 2012b, Chen & Nof, 2012c), and centralized system (Chen & Nof, 2012a). A learning algorithm to minimize fault in the ARS has also been developed (Ajidarma, 2017). Specific error and conflict prevention protocols for collaborating agents within the ARS, however, have not been yet developed. This research aims to design a collaborative detection and prevention of errors and conflicts (CDPEC) in an ARS which consists of sensors-mounted robots, human, and computer as collaborating agents.

The remainder of this article is organized as follows: The research methodology, including the framework, models, and algorithms, is presented in Section 2. The design of experiments and their simulation results are explained in Section 3. The algorithms' performance as observed in the experiments is analyzed in Section 4, and it is further discussed in Section 5. Lastly, a summary of the study and observations, and open challenges for future research are outlined in Section 6.

2. Methodology

The methodology to design collaborative detection and prevention of errors and conflicts (CDPEC) algorithms is developed in four steps. First, the architecture of ARS is presented, including assumptions, technical constraints, and limitations of the system, in accordance with previous research (Bechar & Vigneault, 2016; Dusadeerungsikul, Nof, et al., 2019; Guo et al., 2018). Second, a mathematical model of optimal location sampling is developed, which complements the previously developed ARS routing algorithm (Dusadeerungsikul & Nof, 2019). The model is further developed into two CDPEC algorithms. Third, the design of experiments is outlined and simulated to illustrate and validate the proposed algorithms. Lastly, the results are analyzed and tested to infer and compare the algorithms' performance.

2.1 Architecture of the ARS

This research focuses on the Agricultural Robotic System (ARS) that combines sensors, robots, humans, and agricultural greenhouses as integrated elements of CPS. They operate under a HUB-CI, hub of collaborative intelligence "brain model" with cyber collaborative protocols and algorithms (Nair et al., 2019; Sreeram & Nof, 2021) such as outer space, underwater, telesurgery, manufacturing, and production. In precision agricultural robotics, target monitoring, recognition, and detection is a complex task, requiring expertise. Hence, they can be more efficiently performed by collaborative human-robot systems. A HUB is an online portal, a platform to create and share scientific and advanced computing tools. Multiple HUBs have been developed recently for scientific research objectives. HUB-CI is a tool developed by PRISM Center at Purdue University to enable cyber-augmented collaborative interactions over cyber-supported complex systems. The research reported here, sponsored in part by BARD, implements the HUB-CI model to improve the Collaborative Intelligence (CI). There are heterogeneous crop plants and a variety of sensors involved in the ARS. This research considers an ARS of i heterogeneous plants, indexed by $i = 1, 2, \dots, I$, also considered as objects. The crop plants are monitored by j different sensor types, denoted by $j = 1, 2, \dots, J$. A concept of timeframe is introduced, with each timeframe of data collection

phase in the ARS defined as k , identified by $k = 1, 2, \dots, K$.

There are two different agents operating during each period k : ARS robot, and human operator. The ARS robot is defined as an autonomous mobile agent, augmented with GPS, gyroscope, and guidance camera, based on an agricultural robotic platform developed by Bak & Jakobsen (2004). The human operator is defined as a manual mobile agent that collects the data using static sensors, as proposed in the research conducted by Bloch et al. (2017). The data gathered by autonomous observation is denoted as Y_{ijk} while the one collected by manual observations referred to as Z_{ijk} . The indices of both variables indicate the observation conducted of plant region $i \in I$, using a sensor type $j \in J$, at time $k \in K$.

Each time period Y_{ijk} begins when the ARS robot is initialized and mobilized into the ARS field. The time period ends when the robot finishes gathering Y_{ijk} data. Once the robot stops collecting data, human operator begins operation by sampling a portion of the locations previously visited by the robot, and collects the data manually, which results in Z_{ijk} data. This second data collection aims to verify certain unusual/vague data provided by the robot's sensors. Manual observation data Z_{ijk} are assumed to be more accurate and precise compared to Y_{ijk} , due to the static measurement process that is conducted by human operator. Furthermore, the robot in ARS can only be equipped by a limited quantity of sensors to maintain its mobility. Due to this restriction, Y_{ijk} data observed by the robot will have a higher risk of deviating from the actual state of object i , compared to the Z_{ijk} data.

In the ARS framework, human agent and robot agent have the same set of sensor types with different degrees of accuracy. Human operators are capable of recording highly accurate crops data due to spatial image digitizer (Bloch et al., 2017) and other IoT wearables (Lan et al., 2020; Zhao et al., 2020) we developed a smart plant-wearable biosensor, which can be applied for in-situ analysis of organophosphorus pesticide on crop surfaces. Herein, the serpentine three-electrode system was prepared via the laser-induced graphene (LIG). Meanwhile, robot agents are mounted with single camera, which provides multi-spectral images. Human observation is more time-consuming than robot observation, and thus, the number of human observation instances is significantly lower than

the number of robot observation instances for each period.

In addressing errors and conflicts within ARS, two variables are introduced: sensor observation data and actual state of the agricultural object a_m . Both Y_{ijk} and Z_{ijk} are considered as elements of sensor observation data s_m . Error is defined as the difference between the observation data s_m and actual state of the object a_m . The ratio of error (δ) is expressed as follows:

$$\delta = \frac{1}{M} \sum_{m=1}^M \frac{|s_m - a_m|}{s_m} \quad (1)$$

Furthermore, conflict is defined as the way in which each observation data generated by each sensor s_m varies with one another. It can also be expressed as the deviation of each sensor's observation value from the mean of sensor observation data for a certain $j \in J$. The conflict (ε) is expressed as follows:

$$\varepsilon = \frac{1}{M-1} \sum_{m=1}^M (s_m - \bar{s}_m)^2 \quad (2)$$

There are two instances of sensor observation data based on the agents. Instances of observation conducted by robot agent, or robot observation (RO), is denoted by I^Y , while instances of observation done by human agent, or human observation (HO), is denoted by I^Z . The observation data for both Y_{ijk} and Z_{ijk} are measured in terms of absolute deviation with regard to the ideal state of the object $i \in I$ as measured by sensor type $j \in J$. Raw numerical data s_{ijk} from the sensors are processed with Equation 3 and 4.

$$Y_{ijk} = \min \left\{ \frac{|s_{ijk} - \bar{s}_{ij}|}{s_{ij}^{UB} - s_{ij}^{LB}}, 1 \right\}; s_{ijk} \in I^Y \quad (3)$$

$$Z_{ijk} = \min \left\{ \frac{|s_{ijk} - \bar{s}_{ij}|}{s_{ij}^{UB} - s_{ij}^{LB}}, 1 \right\}; s_{ijk} \in I^Z \quad (4)$$

where s_{ij}^{UB} and s_{ij}^{LB} are the known upper bound and lower bound observation data for every object $i \in I$, measured by sensor type $j \in J$.

Furthermore, $\overline{s_{ij}}$ is the median between s_{ij}^{UB} and s_{ij}^{LB} , and s_{ijk} is the raw numerical observation data of object $i \in I$, measured by sensor type $j \in J$ at time period $k \in K$.

2.2 Mathematical Model of Crop Sampling

For every Y_{ijk} data, a corresponding weight w_{jk} is assigned as a real value between zero and one. The weight indicates the level of accuracy and reliability of each sensor. In a multi-sensor ARS, the inference regarding the system's state is made based on the integrated information of all sensors. Sensors which are not capable of detecting a disease, are to be penalized by lowering the corresponding weight w_{jk} . The objective criterion of the system can be expressed as maximizing the detection of diseased crops, which is formulated as a Mixed-Integer Linear Programming (MILP) of crop sampling:

$$\max_{x_{ijk}} \sum_{i \in I} \sum_{j \in J} w_{j(k-1)} Y_{ijk} x_{ijk} \quad (5)$$

$$\sum_{i \in I} \sum_{j \in J} t_{ijk} x_{ijk} \leq T_k, \forall k \in K \quad (6)$$

$$\sum_{i \in I} x_{ijk} \geq 1, \forall j \in J, k \in K \quad (7)$$

$$x_{ijk} \in \{0, 1\}, \forall i \in I, j \in J, k \in K \quad (8)$$

where Y_{ijk} is the robot observation data, w_{jk} is the weight of each sensor type $j \in J$ at time period $k \in K$, and x_{ijk} is a binary decision variable denoting whether further observation is to be conducted at object $i \in I$, using sensor type $j \in J$ at period $k \in K$. If an observation is collected, value of x_{ijk} equals one, otherwise zero. For each time period $k \in K$, the objective criterion incorporates weight $w_{j(k-1)}$ from the previous time period, or period $k-1$ (Equation 5).

Human observations (HO) and robot observations (RO) are modeled sequentially. The time required by the robot to collect Y_{ijk} data is unbounded by any constraint as the robots are capable of operating continuously. Meanwhile, data gathered by human operator, denoted by Z_{ijk} , is limited by the data collection duration. The time required to manually collect data is defined as t_{ijk} . Manual agents are presumed to be less efficient and slower in gathering the Z_{ijk} data and thus, bounded by

T_k (Equation 6), which is the available duration to gather Z_{ijk} for each period $k \in K$. Lastly, every sensor type is to be used at least once to measure any crops (Equation 7).

After the sampling is conducted, the decision variables x_{ijk} route the human agent to visit a partial subset of locations that robot agent has visited previously, as follows:

$$HO = \begin{cases} Z_{ijk}, & \text{if } x_{ijk} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

The learning nature of the model is represented by updating the w_{jk} value for each time period. At the first period, all weights for each RO are assumed to be 1.

2.3 Mathematical Model of Sensor Weight Update

In the ARS, multiple agents and sensors are utilized. The first agent is the less accurate, sensor-mounted ARS robot, and the second agent is human operator, which conducts the observation manually with a highly-accurate static sensor. According to the principle of CCT, a system of collaborating agents is prone to error and conflict. In this subsection, a sensor weight update model, which aims to minimize both error and conflict, is presented. The weight w_{jk} is updated such that the weighted RO has minimum difference when compared with the weighted HO. A vector \vec{f} is defined for each weight w_{jk} as follows:

$$\vec{f}(w_{jk}) = \{f_1(w_{jk}), f_2(w_{jk}), \dots, f_i(w_{jk})\}; \quad (10)$$

$$\forall j \in J, k \in K$$

Using the previous definitions of RO and HO, the function f_i equals:

$$f_i(w_{jk}) = w_{jk} Y_{ijk} - Z_{ijk} x_{ijk}; \forall i \in I \quad (11)$$

Therefore, decision regarding the optimal value of w_{jk} can be formulated as a least square optimization problem. The optimum weight w_{jk} from least square fitting obtained for each sensor $j \in J$ at time period k is to be used for the optimal sampling problem in $k+1$. The first weight update formulation is expressed as follows:

$$\min_w \|\bar{f}(w_{jk})\|_2^2 = \min_w \sum_{i \in I} f_i^2(w_{jk}) \quad (12)$$

Subject to:

$$w_{jk} \in \mathbb{R}^{J \times K} \quad (13)$$

$$\bar{f}(w_{jk}) = \begin{pmatrix} f_1(w_{jk}) \\ \vdots \\ f_I(w_{jk}) \end{pmatrix} \in \mathbb{R}^I; \forall j \in J, k \in K \quad (14)$$

$$f_i(w_{jk}) = w_{jk} Y_{ijk} - Z_{ijk} x_{ijk}; \forall i \in I \quad (15)$$

Another sensor weight update model is formulated. In the second formulation of optimal weighting, \bar{f} is modelled as a temporary memory to store the RO and HO data from only one-step iteration behind; unlike the first optimal weighting where RO and HO data are cumulatively added into \bar{f} over time K .

Therefore, in this second formulation, the size of vector \bar{f} remains constant with regard to the number of sensors, or in other words, the weight matrix in Equation 13 is now one dimensional:

$$w_{jk} \in \mathbb{R}^I \quad (16)$$

2.4 Proposed Algorithms

Two algorithms are developed based on the problem formulation. Figure 1 depicts the conceptual model of the collaborative detection and prevention of errors and conflicts (CDPEC)

in the ARS. The algorithm works in a sequence of stepwise functions. The model's input is the scaled observational data, which was transformed from raw sensor data using Equations 3 and 4. The conceptual model is developed into two collaborative detection and prevention of errors and conflicts algorithms.

The two algorithms differ in terms of number of calls and memory requirements. The pseudocode for the first algorithm is presented in Table 1.

Table 1. Collaborative Detection and Prevention of Errors and Conflict Algorithm 1

CDPEC Algorithm 1

```

initialize the values of  $I, J, K, t, T, \delta, \varepsilon$ 

generate  $Y_{ijk}, t_{ijk}, w_{j0}$ 
for  $k:=1$  to  $K$  do
  call crop sampling

  generate  $Z_{ijk} = I^Z(x_{ijk})$ 
  for  $j:=1$  to  $J$  do

    call weight update  $\bar{f}_k$ 
  end for
  calculate MSE, CEPR of each time  $k$ 

  reset  $\bar{f}_k$ 
end for
calculate CEPR of the ARS
end

```

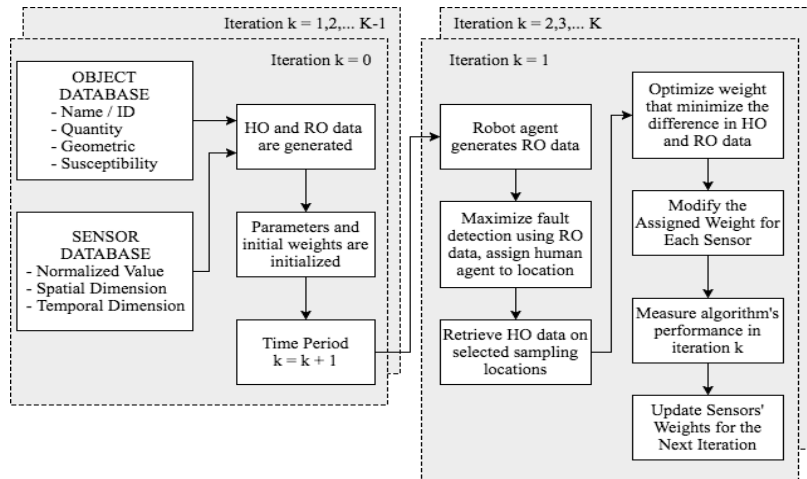


Figure 1. Logical Model of the Algorithms

the steps based on the models explained in subsections 2.2 and 2.3. In CDPEC 1, the sensor weight update uses only the human observation and robot observation data from $k-1$ period. When iteration k begins, the observation data from the ARS sensors are gathered in a database vector \vec{f} , containing the data from time $k-1$. Before progressing into the next iteration, the algorithm automatically resets \vec{f} . Therefore, the size of \vec{f} remains constant as the number of collected data from each iteration is also constant.

Table 2. Collaborative Detection and Prevention of Errors and Conflict Algorithm 2

CDPEC Algorithm 2

initialize the values of $I, J, K, t, T, \delta, \varepsilon$

generate set $\vec{f}_k = [0]_{J \times K}$

for $k:=1$ to K **do**

call crop sampling

generate $Z_{ijk} = I^Z(x_{ijk})$

for $j:=1$ to J **do**

call weight update \vec{f}_k

end for

calculate MSE, CEPR of each time k

end for

set $\vec{f}_k = [\vec{f}_{k-1}, \vec{f}_k]$

calculate CEPR of the ARS

end

The pseudocode of the second algorithm is presented in Table 2. In the case of CDPEC 2, the algorithm accumulates HO and RO data since $k=1$, and unlike CDPEC 1, \vec{f} entry is not emptied after each iteration.

As iterations progress, the dimension of \vec{f} expands with the accumulated values of w_{jk} . Consequently, the increasing size of \vec{f} in CDPEC 2 demands a relatively larger memory. It may be prohibitive with many sensors and agents, as collaborative interactions and data exchange increase exponentially.

3. Design of Experiments

3.1 Performance Metrics and Baseline Scenario

Performance metrics help determine the algorithms' effectiveness, and are measured by

comparing the baseline case with the cases when CDPEC 1 or CDPEC 2 are executed in ARS. In the baseline scenario, every sensor has a uniform weight, $w_{jk} = 1$. With CDPEC 1 or CDPEC 2, the weights vary between zero and one.

The algorithms' effectiveness is measured in terms of Mean Squared Error for each object and each sensor MSE_{ij} (Equation 17), which measures the mean sum of squares of the deviation between the RO data (Y_{ijk}) and HO data (Z_{ijk}). The Mean Squared Error can be aggregated and measured for each sensor MSE_j , as stated in Equation 18.

$$MSE_{ij} = \frac{1}{K} \sum_{k \in K} (w_{jk} Y_{ijk} - Z_{ijk})^2 \quad (17)$$

$$MSE_j = \frac{1}{I} \sum_{i \in I} MSE_{ij} \quad (18)$$

Another performance metric is Conflict and Error Prevention Rate (CEPR). It measures the effectiveness of the collaborative protocols in detecting and preventing both error and conflict. As stated in Equation 19, the metric calculates the ratio of MSE between the cases when an algorithm is implemented and the cases when the sensors are not weighted (baseline scenario). CEPR is measured for each sensor type (Equation 19). The efficacy of the algorithms on the entire ARS is also calculated by averaging $CEPR_j$ for all sensor type $j \in J$ (Equation 20).

$$CEPR_j = 1 - \frac{MSE_j \text{ for weighted sensor scenario}}{MSE_j \text{ for baseline scenario}} \quad (19)$$

$$CEPR_{ARS} = \frac{1}{J} \sum_{j \in J} CEPR_j \quad (20)$$

Statistical significance test, pairwise t-test, is used to compare a sample from the MSE of the algorithm with a sample of the MSE of the baseline, uniform-weighted sensor observations. Rejecting the null hypothesis means that the algorithms can produce relative weights such that the weighted sensor data has a reduced MSE compared to the one of the baseline scenario, with a valid statistical significance.

3.2 Parameters of the Experiments

The algorithm is validated through a series of simulations. The simulated scenarios are formulated with the following notation:

$$S_\eta = \{n, \delta, \varepsilon\} \quad (21)$$

Each scenario (S_η) has a specified parameter set (Table 3): number of sensors (n), inherent error (δ) between RO and HO, and inherent conflict between sensors (ε).

Table 3. Parameters of the Experiment Design

Parameters	Value
Number of sensors (n)	Small: 2 sensors
	Medium: 5 sensors
	Large: 10 sensors
	Very Large: 20 sensors
Inherent sensor error rate (δ)	Small: 10% deviation with HO data
	Large: 30% deviation with HO data
Conflict rate between sensors (ε)	Small: 5% deviation between sensors
	Large: 10% deviation between sensors

3.3 Simulation Results

Sixteen scenarios are generated based on the level of relevant factors. For every scenario, the algorithm is replicated 20 times. In each iteration, the MSE of weighted-sensor and baseline cases are recorded, and used to calculate CEPR.

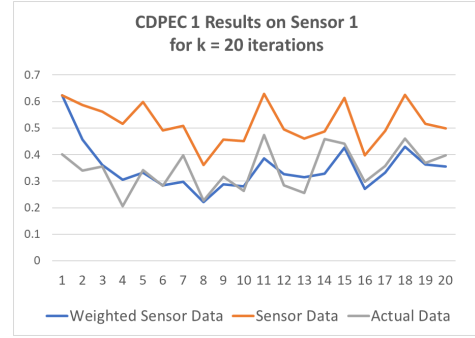


Figure 2. CDPEC 1 Results on Sensor 1

Results are plotted to visually demonstrate the weighted sensor data's ability to match actual data better than raw, non-weighted sensor data (Figure 2). Weighted sensor data, non-weighted sensor data, and the actual data for the case when CDPEC 2 is implemented are compared (Figure 3). The quantitative results of the experiments, in terms of CEPR, are presented in Table 4.

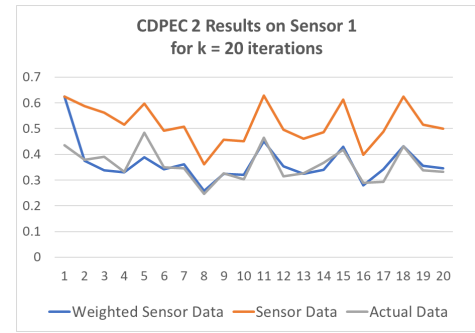


Figure 3. CDPEC 2 Results on Sensor 1

Table 4. Experiment Results of CDPEC 1 and CDPEC 2 Algorithms

Scenario $S_h = \{n, \delta, \xi\}$	CEPR for CDPEC 1		CEPR for CDPEC 2	
	Mean	Std.Dev	Mean	Std.Dev
$S_1 = \{2,0.1,0.05\}$	47.92%	11.91%	76.94%	7.38%
$S_2 = \{2,0.1,0.1\}$	-27.57%	31.58%	50.75%	13.63%
$S_3 = \{2,0.3,0.05\}$	84.12%	2.53%	90.05%	0.99%
$S_4 = \{2,0.3,0.1\}$	65.03%	9.28%	85.59%	2.08%
$S_5 = \{5,0.1,0.05\}$	50.57%	25.52%	83.64%	3.24%
$S_6 = \{5,0.1,0.1\}$	-16.50%	57.74%	61.50%	11.07%
$S_7 = \{5,0.3,0.05\}$	84.47%	2.95%	96.82%	0.49%
$S_8 = \{5,0.3,0.1\}$	74.27%	6.12%	92.50%	1.60%
$S_9 = \{10,0.1,0.05\}$	39.33%	14.86%	89.10%	1.51%
$S_{10} = \{10,0.1,0.1\}$	-16.98%	41.50%	68.39%	3.57%
$S_{11} = \{10,0.3,0.05\}$	88.69%	2.98%	97.13%	0.81%
$S_{12} = \{10,0.3,0.1\}$	77.78%	5.27%	93.21%	1.08%
$S_{13} = \{20,0.1,0.05\}$	31.57%	17.16%	87.18%	2.22%
$S_{14} = \{20,0.1,0.1\}$	-32.77%	27.23%	58.46%	5.79%
$S_{15} = \{20,0.3,0.05\}$	85.33%	5.35%	98.37%	0.47%
$S_{16} = \{20,0.3,0.1\}$	68.35%	5.60%	94.03%	0.88%

4. Analysis

Statistical test is conducted between the MSE of the algorithm's results and the one of the baseline scenario. To maintain the consistency of the distribution, the t-test uses only the MSE measured by one sensor on one object for every scenario, which is randomly selected. The null hypothesis is that the mean of MSE values of algorithm results equals the mean of MSE values of the baseline case. The t-test results are presented in Table 5.

Based on the results, CDPEC 1 performs well in most scenarios, except the ones when the conflict rate is relatively high and the error rate is relatively low. In these cases, the means of CEPR of CDPEC 1 are negative, which means that the algorithm results have higher MSE values compared to the ones of the baseline scenario.

Table 5. Statistical Significance Test

Scenario $S_\eta = \{n, \delta, \varepsilon\}$	P-value of independent t-test ($\alpha = 0.05$)	
	CDPEC 1	CDPEC 2
S1	0.0046*	7.43×10^{-5} *
S2	0.064	0.072
S3	7.6×10^{-15} *	4.31×10^{-12} *
S4	3.41×10^{-7} *	1.88×10^{-11} *
S5	0.023*	1.09×10^{-6} *
S6	0.977	0.12
S7	3.01×10^{-15} *	3.55×10^{-16} *
S8	1.18×10^{-8} *	3.12×10^{-12} *
S9	5.32×10^{-5} *	1.49×10^{-5} *
S10	0.336	0.042
S11	1.36×10^{-22} *	3.54×10^{-14} *
S12	8.05×10^{-5} *	1.52×10^{-11} *
S13	1.29×10^{-4} *	1.2×10^{-4} *
S14	0.426	0.051
S15	1.80×10^{-12} *	2.07×10^{-12} *
S16	3.04×10^{-7} *	2.76×10^{-8} *

* Statistically significant at ($\alpha = 0.05$)

Upon further analysis, it is found that the combination of small error rate and high conflict rate marginally inflated the difference between the observations and actual data, such that it has a random pattern. The algorithm often failed to find local optima in these cases as the data mining of the previous iteration is practically inaccurate to predict a random pattern.

The results indicate that CDPEC 2 can reduce a significant portion of errors and conflicts in every scenario. In most cases, the reductions are statistically significant, with the null hypothesis being rejected. In some cases, however, where the conflict rate is high and the error rate is low, the results are not statistically significant. Even in these cases, CDPEC 2 still maintains a relatively acceptable level of CEPR, whereas CDPEC 1 fails.

5. Discussions

Experimental results demonstrate the validity of these algorithms to a certain extent, as both CDPEC 1 and CDPEC 2 can significantly reduce faults by improving the conflict and error prevention rate (CEPR) within ARS for a given greenhouse.

It is again evident that a collaborative "brain model," such as HUB-CI, is needed for improved cyber-physical interactions. It is also evident that different protocols and algorithms, with their relative trade-offs, can impact differently the extent and value of the gained benefits.

The first sub-model of the algorithms focuses on sampling HO data, modelled as a MILP problem. It applies, as the data are normalized and expressed on a scale relative to the known deviation of the sensor data. In more general cases, however, where input data are neither scaled nor normalized, MILP may not be as effective. Then, metaheuristic optimization by evolutionary or swarm algorithms appears to be potentially useful. Second, recent developments in industrial internet and wearables may lead to high volume and variety of agricultural crops' data. While this research models human operator as a relatively more accurate yet slower agent, advances of industrial internet wearables may speed up data collection by human agents. To accommodate such data exchange, novel protocols and extension of ARS architecture will be necessary.

6. Conclusion

This research aims to detect early and prevent potential errors and conflicts within an agricultural robotic system (ARS) with collaborating agents. HUB-CI/ARS, a cyber hub for collaborative intelligence gathering and exchange among ARS agents, was designed to operate under various cyber-collaborative protocols and algorithms: Collaborative Control Protocol in Cyber-

Physical System (CCP-CPS) (Dusadeerungsikul, Sreeram et al., 2019), CCP-CPS with routing and searching algorithms (Dusadeerungsikul, Nof et al., 2019) and a Human-in-the-Loop integrated ARS (Sreeram & Nof, 2021). Two collaborative detection and prevention of errors and conflicts (CDPEC) algorithms are proposed, developed, and validated. On average, for the same scenarios, CDPEC 1 reduces potential faults to 66.4% compared to the baseline; CDPEC 2 reduces potential faults to 86.9%. CDPEC 2 enables 30.9% higher effectiveness relative to CDPEC 1.

Future researchers can extend this study by developing a data pre-processing procedure, capable of processing a variety of non-standardized input data. Furthermore, in order to improve its monitoring and detection capabilities, ARS robots

could be modelled with a variety of novel types of sensors, which enable a better data acquisition.

Acknowledgements

The research reported in this paper is supported by the Production, Robotics, and Integration Software for Manufacturing & Management (PRISM) Center at Purdue University. The research on ARS is supported by BARD project Grant# IS-4886-16R, “Development of a Robotic Inspection System for Early Identification and Locating of Biotic and Abiotic Stresses in Greenhouse Crops”; and NSF project Grant# 1839971, “FW-HTF: Collaborative Research: Pre-Skilling Workers, Understanding Labor Force Implications and Designing Future Factory Human-Robot Workflows Using a Physical Simulation Platform”.

REFERENCES

- Ajidarma, P. (2017). *Multi-Sensor Fault Tolerant Learning Algorithm in an Agricultural Robotic System* Purdue University. Available at: <<https://docs.lib.purdue.edu/dissertations/AAI10685624/>>.
- Baille, A., Kittas, C. & Katsoulas, N. (2001). Influence of whitening on greenhouse microclimate and crop energy partitioning, *Agricultural and Forest Meteorology*, 107(4), 293–306. DOI: 10.1016/S0168-1923(01)00216-7
- Bak, T. & Jakobsen, H. (2004). Agricultural Robotic Platform with Four Wheel Steering for Weed Detection, *Biosystems Engineering*, 87(2), 125–136. DOI: 10.1016/j.biosystemseng.2003.10.009
- Bautista-Gallego, J., Rodríguez-Gómez, F., Barrio, E., Querol, A., Garrido-Fernández, A. & Arroyo-Lopez, F. N. (2011). Exploring the yeast biodiversity of green table olive industrial fermentations for technological applications, *International Journal of Food Microbiology*, 147(2), 89–96. DOI: 10.1016/j.ijfoodmicro.2011.03.013
- Bechar, A. & Eben-Chaime, M. (2014). Hand-held computers to increase accuracy and productivity in agricultural work study, *International Journal of Productivity and Performance Management*, 63(2), 194–208. DOI: 10.1108/IJPPM-03-2013-0040
- Bechar, A. & Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components, *Biosystems Engineering*, 149, 94–111. DOI: 10.1016/j.biosystemseng.2016.06.014
- Bloch, V., Bechar, A. & Degani, A. (2017). Development of an environment characterization methodology for optimal design of an agricultural robot, *Industrial Robot: An International Journal*, 44(1), 94–103. DOI: 10.1108/IR-03-2016-0113
- Bruno, G. & Antonelli, D. (2018). Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells, *The International Journal of Advanced Manufacturing Technology*, 98, 2415–2427. DOI: 10.1007/s00170-018-2400-4
- Chen, X. & Nof, S. (2012a). Constraint-based conflict and error management, *Engineering Optimization - ENG OPTIMIZ*, 44(7), 821–841. DOI: 10.1080/0305215X.2011.613466
- Chen, X. W. & Nof, S. Y. (2012b). Agent-based error prevention algorithms, *Expert Systems with Applications*, 39(1), 280–287. DOI: 10.1016/j.eswa.2011.07.018
- Chen, X. W. & Nof, S. Y. (2012c). Conflict and error prevention and detection in complex networks, *Automatica*, 48(5), 770–778. DOI: 10.1016/j.automatica.2012.02.030
- Dusadeerungsikul, P. O. & Nof, S. Y. (2019). A collaborative control protocol for agricultural robot routing with online adaptation, *Computers and Industrial Engineering*, 135, 456–466. DOI: 10.1016/j.cie.2019.06.037
- Dusadeerungsikul, P. O., Nof, S. Y., Bechar, A. & Tao, Y. (2019). Collaborative control protocol for agricultural cyber-physical system, *Procedia Manufacturing*, 39, 235–242. DOI: 10.1016/j.promfg.2020.01.330
- Dusadeerungsikul, P. O., Sreeram, M., He, X., Nair, A., Ramani, K., Quinn, A. J. & Nof, S. Y. (2019). Collaboration requirement planning protocol for hub-Ci in factories of the future, *Procedia Manufacturing*, 39, 218–225. DOI: 10.1016/j.promfg.2020.01.327

- Filip, F. G., Zamfirescu, C. B. & Ciurea, C. (2017). *Computer-Supported Collaborative Decision-Making*, 1st edition. Springer International Publishing. Available at: <<http://lib.ugent.be/catalog/ebk01:371000000918172>>
- Guo, P., Dusadeerungsikul, P. & Nof, S. Y. (2018). Agricultural cyber physical system collaboration for greenhouse stress management, *Computers and Electronics in Agriculture*, 150, 439–454. DOI: 10.1016/j.compag.2018.05.022
- Haibo, L., Dong, S., Zunmin, L. & Chuijie, Y. (2015). Study and Experiment on a Wheat Precision Seeding Robot, *Journal of Robotics*, 2015(2), 1–9. DOI: 10.1155/2015/696301
- van Henten, E. J., Hemming, J., Tuijl, B., Kornet, J., Meuleman, J., Bontsema, J. & van Os, E. (2002). An Autonomous Robot for Harvesting Cucumbers in Greenhouses, *Autonomous Robots*, 13, 241–258. DOI: 10.1023/A:1020568125418
- Hiremath, S. A., van der Heijden, G. W. A. M., van Evert, F. K., Stein, A. & Ter Braak, C. J. F. (2014). Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter, *Computers and Electronics in Agriculture*, 100, 41–50. DOI: 10.1016/j.compag.2013.10.005
- Lan, L., Le, X., Dong, H., Xie, J., Ying, Y. & Ping, J. (2020). One-step and large-scale fabrication of flexible and wearable humidity sensor based on laser-induced graphene for real-time tracking of plant transpiration at bio-interface, *Biosensors and Bioelectronics*, 165, 112360. DOI: 10.1016/j.bios.2020.112360
- Li, N., Remeikas, C., Xu, Y., Jayasuriya, S. & Ehsani, R. (2015). Task Assignment and Trajectory Planning Algorithm for a Class of Cooperative Agricultural Robots, *Journal of Dynamic Systems Measurement and Control*, 137(5), 51004. DOI: 10.1115/1.4028849
- Mcbratney, A., Whelan, B., Ancev, T. & Bouma, J. (2005). Future Directions of Precision Agriculture, *Precision Agriculture*, 6, 7–23. DOI: 10.1007/s11119-005-0681-8
- Moghaddam, M. & Nof, S. Y. (2018). Collaborative service-component integration in cloud manufacturing, *International Journal of Production Research*, 56(1–2), 677–691. DOI: 10.1080/00207543.2017.1374574
- Moghaddam, M., Nof, S. Y. & Menipaz, E. (2016). Design and administration of collaborative networked headquarters, *International Journal of Production Research*, 54(23), 7074–7090. DOI: 10.1080/00207543.2015.1125544
- Morio, Y., Shoji, T. & Murakami, K. (2016). Working motion templates for detecting agricultural worker behaviors, *Engineering in Agriculture, Environment and Food*, 9(4), 297–304. DOI: 10.1016/j.eaef.2016.03.002
- Nair, A. S., Bechar, A., Tao, Y. & Nof, S. Y. (2019). The HUB-CI Model for Telerobotics in Greenhouse Monitoring, *Procedia Manufacturing*, 39, 414–421. DOI: 10.1016/j.promfg.2020.01.385
- Nof, S. Y. (2007). Collaborative control theory for e-Work, e-Production, and e-Service, *Annual Reviews in Control*, 31(2), 281–292. DOI: 10.1016/j.arcontrol.2007.08.002
- Nof, S. Y. (ed.). (2009). *Springer Handbook of Automation*. Springer-Verlag Berlin Heidelberg.
- Nof, S. Y. (2017). Collaborative Control Theory and Decision Support Systems, *Computer Science Journal of Moldova*, 25(2), 115–144.
- Okayama, T. & Miyawaki, K. (2013). The “smart garden” system using augmented reality, *IFAC Proceedings Volumes*, 46(4), 307–310. DOI: 10.3182/20130327-3-jp-3017.00070
- Sharma, S., Jagyasi, B., Raval, J. & Patil, P. (2015). AgriAct: Agricultural Activity Training using multimedia and wearable sensing. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, (pp. 439–444). DOI: 10.1109/PERCOMW.2015.7134078
- Sreeram, M. & Nof, S. Y. (2021). Human-in-the-loop of Cyber Physical Agricultural Robotic Systems, *International Journal of Computers, Communications and Control*, 16(2). DOI: 10.25394/PGS.13303490.v1
- Xia, C., Wang, L., Chung, B.-K. & Lee, J.-M. (2015). In Situ 3D Segmentation of Individual Plant Leaves Using a RGB-D Camera for Agricultural Automation, *Sensors*, 15(8), 20463–20479. DOI: 10.3390/s150820463
- Zhang, Q. (2013). Opportunity of robotics in specialty crop production, *IFAC Proceedings Volumes*, 46(9), 38–39. DOI: 10.3182/20130327-3-JP-3017.00011
- Zhao, F., He, J., Li, X., Bai, Y., Ying, Y. & Ping, J. (2020). Smart plant-wearable biosensor for in-situ pesticide analysis, *Biosensors and Bioelectronics*, 170, 112636. DOI: 10.1016/j.bios.2020.112636
- Zhong, H., Ozsoy, E. & Nof, S. Y. (2016). Co-Insights framework for collaborative decision support and tacit knowledge transfer, *Expert Systems with Applications*, 45, 85–96. DOI: 10.1016/j.eswa.2015.09.036