HUMAN-IN-THE-LOOP OF CYBER PHYSICAL AGRICULTURAL ROBOTIC SYSTEMS

by

Maitreya Sreeram

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

School of Industrial Engineering

West Lafayette, Indiana

December 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Shimon Y. Nof, Chair

School of Industrial Engineering

Dr. Brandon Pitts

School of Industrial Engineering

Dr. Andrew Liu

School of Industrial Engineering

Approved by:

Dr. Abhijit Deshmukh

Dedicated to my family and friends

ACKNOWLEDGMENTS

The research presented here 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” with Co-PIs A. Bechar (ARO, Israel); Y. Tao (UMD); and S.Y. Nof. It is also supported by 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” with PI K. Ramani; Co-PIs S.Y. Nof; A. Quinn and T. Redick.

First and foremost, I would like to give my thanks to Dr. Shimon Y. Nof, my thesis advisor and mentor for his continuous support and guidance throughout my time at Purdue. Performing research with him has provided me valuable experiences and invaluable life lessons, and he has pushed me to strive for the very best. I would also like to acknowledge my committee members, Dr. Andrew Liu and Dr. Brandon Pitts for their support. From PRISM center, I would like to offer my thanks to my fellow researchers: Dr. Oak, Dr. Win, Ashwin and Billy for the time we spent together, and for their timely and insightful advice in all matters.

Finally, I would like to thank my family: my mother Jayashree, my father Sreeram and my brother Chinmaya- for their never-ending belief and faith in me. This would not be possible without their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The onset of Industry 4.0 has provided considerable benefits to Intelligent Cyber-Physical Systems (ICPS), with technologies such as internet of things, wireless sensing, cognitive computing and artificial intelligence to improve automation and control. However, with increasing automation, the “human” element in industrial systems is often overlooked for the sake of standardization. While automation aims to redirect the workload of human to standardized and programmable entities, humans possess qualities such as cognitive awareness, perception and intuition which cannot be automated (or programmatically replicated) but can provide automated systems with much needed robustness and sustainability, especially in unstructured and dynamic environments. Incorporating tangible human skills and knowledge within industrial environments is an essential function of “Human-in-the-loop” (HITL) Systems, a term for systems powerfully augmented by different qualities of human agents. The primary challenge, however, lies in the realistic modelling and application of these qualities; an accurate human model must be developed, integrated and tested within different cyber-physical workflows to 1) validate the assumed advantages, investments and 2) ensure optimized collaboration between entities. Agricultural Robotic Systems (ARS) are an example of such cyber-physical systems (CPS) which, in order to reduce reliance on traditional human-intensive approaches, leverage sensor networks, autonomous robotics and vision systems and for the early detection of diseases in greenhouse plants. Complete elimination of humans from such environments can prove sub-optimal given that greenhouses present a host of dynamic conditions and interactions which cannot be explicitly defined or managed automatically. Supported by efficient algorithms for sampling, routing and search, HITL augmentation into ARS can provide improved detection capabilities, system performance and stability, while also reducing the workload of humans as compared to traditional methods. This

research thus studies the modelling and integration of humans into the loop of ARS, using simulation techniques and employing intelligent protocols for optimized interactions. Human qualities are modelled in human "classes" within an event-based, discrete time simulation developed in Python. A logic controller based on collaborative intelligence (HUB-CI) efficiently dictates workflow logic, owing to the multi-agent and multi-algorithm nature of the system. Two integration hierarchies are simulated to study different types of integrations of HITL: Sequential, and Shared Integration. System performance metrics such as costs, number of tasks and classification accuracy are measured and compared for different collaboration protocols within each hierarchy, to verify the impact of chosen sampling and search algorithms. The experiments performed show the statistically significant advantages of HUB-CI based protocol over traditional protocols in terms of collaborative task performance and disease detectability, thus justifying added investment due to the inclusion of HITL. The results also discuss the competitive factors between both integrations, laying out the relative advantages and disadvantages and the scope for further research. Improving human modelling and expanding the range of human activities within the loop can help to improve the practicality and accuracy of the simulation in replicating an HITL-ARS. Finally, the research also discusses the development of a user-interface software based on ARS methodologies to test the system in the real-world.

1. INTRODUCTION

1.1 Background

Multi-Agent systems are systems composed of multiple interacting computing elements known as agents. Agents are physical or virtual entities with two major abilities: 1) Capable of some extent of autonomous behavior, and 2) Capable of interacting with other agents. The interactions are not limited to just the sharing of data, but rather a robust flow of information and decisions. Precision Agriculture (PA) is one such Multi-Agent Cyber-Physical System, which is an agricultural management system tasked with the monitoring of plant status based on technologies such as spectral imaging collected by specialized acquisition systems (Rad, Hancu, Takacs, & Olteanu, 2015). As an integrated information and production-based farming system, PA technologies also incorporate recent advances in modern agriculture such as agricultural robotics and wireless sensor networks, providing evidence for lower production costs, increased farming efficiency and reduced impacts (Koutsos & Menexes, 2019). Agricultural Robots have been used for tasks such as monitoring of crops, plant protection and weed control, fruit harvesting and mapping (Bechar & Vigneault, 2016). Mobile robots provide a stable medium to sample plants in controlled environments without risking exposure to externalities. Agricultural plants encounter a vast set of stresses and anomalies, given the unstructured nature of environmental factors such as temperature, humidity, air flow and pests. Such factors can be effectively measured using state-of-the-art sensor networks (Gongal, Karkee, & Lewis, 2015). It is important to keep a track of environmental factors since they play a significant role in plant growth and can often be the driving factor of disease or infection propagation in plants (Dik, Polyakova, Chelovechkova, Moskvin, & Nikiforova, 2018). Effective control strategies are required to ensure appropriate detection and localization of plant stresses if detected, to prevent irreversible and irreparable damages.

Collaborative Intelligence (CI) is a measure to gauge the efficiency and effectiveness of collaboration between diverse agents in a distributed network (Devadasan, Zhong, & Nof, 2013). Supported by protocols from Collaborative Control Theory (CCT) (Nof, 2007), CI algorithms and tools have been developed and employed on HUB-CI (HUB for Collaborative Intelligence), a platform which enables cross-domain collaboration between agents such as humans, robots and computational machines. Given the dynamic mix of agents in agricultural systems, it is important to optimize the interactions between them to ensure efficient sharing of information and decisions.

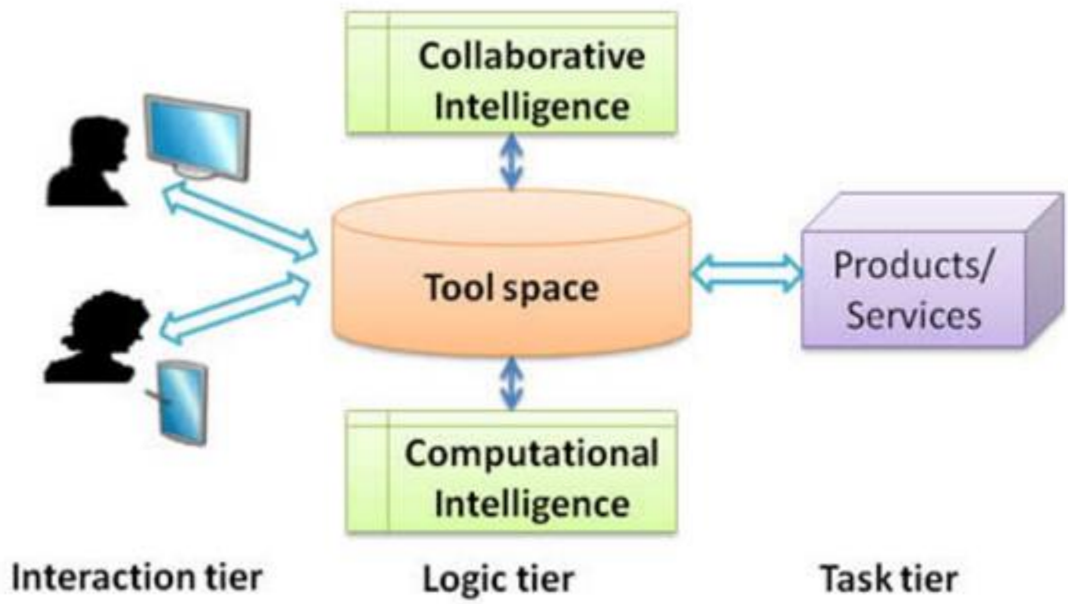


Figure 1: HUB-CI Infrastructure (Courtesy Zhong, Wachs, & Nof, 2013)

Human-in-the-loop (HITL) systems are systems in which human agents play active role in providing real-time decision support and information sharing (Emmanouilidis et al., 2019a). Automation of agricultural processes has resulted in reduction of manual labor, lower production costs, improved control of the production environment for crops and reduced yield losses (Sarangi, Umadikar, & Kar, 2016). However, automation can prove disadvantageous in unstructured terrains where systems have to deal with emergent and random events. In such cases it is useful to leverage the versatility and cognitive perception of human agents in dealing with unpredictability. Humans can complete some decision making and tasks more accurately than current autonomous systems (Bringes, Lin, Sun, & Alqasemi, 2013). HITL Systems are thus hybrid by design, combining the cognitive capability of humans with the physical advantages of automated systems.

This research thus reports the development of Human-in-the-loop workflows for an Cyber-Physical Agricultural Robotic System which deals with the early identification and localization of biotic and abiotic stresses in plants.

1.2 Abbreviations

Table 1: List of Abbreviations

Abbreviation	Explanation
ARS	Agricultural Robotic System
AS	Adaptive Search
BARD	Binational Agricultural Research and Development Fund
CCT	Collaborative Control Theory
CI	Collaborative Intelligence
CPS	Cyber Physical System
DSS	Decision Support System
HA	Human Agent
HE	Human Expert
HITL	Human-in-the-loop
HRC	Human-Robot Collaboration
HRI	Human-Robot Interaction
HUB-CI	HUB based Collaborative Intelligence
ICPS	Intelligent CPS
IoS	Internet of Services
IoT	Internet of Things
KPI	Key Performance Indicator
MDR	Missed Detection Ratio
OHL	Optimal HITL Level
PA	Precision Agriculture
RA	Robotic Agent
RSR	Redundant Sampling Ratio

1.3 Research Objectives

With increasing attention to food security and reliability, there is a need to study and develop robust agricultural systems capable of maximizing productivity from the chosen set of agents. The primary goal of this research is to simulate and understand the impact of integrating humans in the cyber-physical loop to improve the productivity of agricultural systems, especially in the timely detection of plant diseases. The objectives of this research include:

1. Creating a robust multi-agent system of Human, Cyber and Machine agents capable of communicating and collaborating with each other for early disease detection and prevention in crops

2. Modeling and simulating human operators as intelligent agents capable of integration within information and decision flows; Explore different Human-in-the-loop integration architectures and find the critical balance of interfacing between different agents to maximize system productivity
3. Developing an interactive Human-in-the-loop Simulator based on research outcomes and understandings

1.4 Research Questions

Based on the Research problems, the following research questions have been formulated:

1. What simulation models and technologies can we use to improve the productivity of current Agricultural Robotic Systems?
2. Cyber-Supported Collaboration:
    a. How do we use Cyber-Supported Collaboration for Agricultural Robotic Systems?
    b. Which collaboration protocols can we employ between agents to provide the optimal performance parameters?
3. Human-Machine Interactions:
    a. How do we configure the system for Human-Machine Interactions? What considerations do we make while modelling system agents?
    b. How do different Human-in-the-loop integrations affect the system? What are the different advantages and disadvantages to these integrations?

1.5 Research Significance

The significance of the research problem and formulated research questions are as follows:

1. Rather than remove humans from automated systems, this research aims to highlight the significant advantages of humans in the loop, with respect to human decision making and cognitive abilities.
2. Establishing a collaborative platform to manage interactions and communications between a diverse set of agents (robots, humans and sensors), including flow of information and decisions for different workflows.
3. Understanding the critical balance of control in such systems, based on system performance for different configurations and agent availabilities

These issues aim to provide evidence to support the existence, development and sustainability of HITL-ARS.

2. LITERATURE REVIEW

2.1 Cyber-Physical Systems

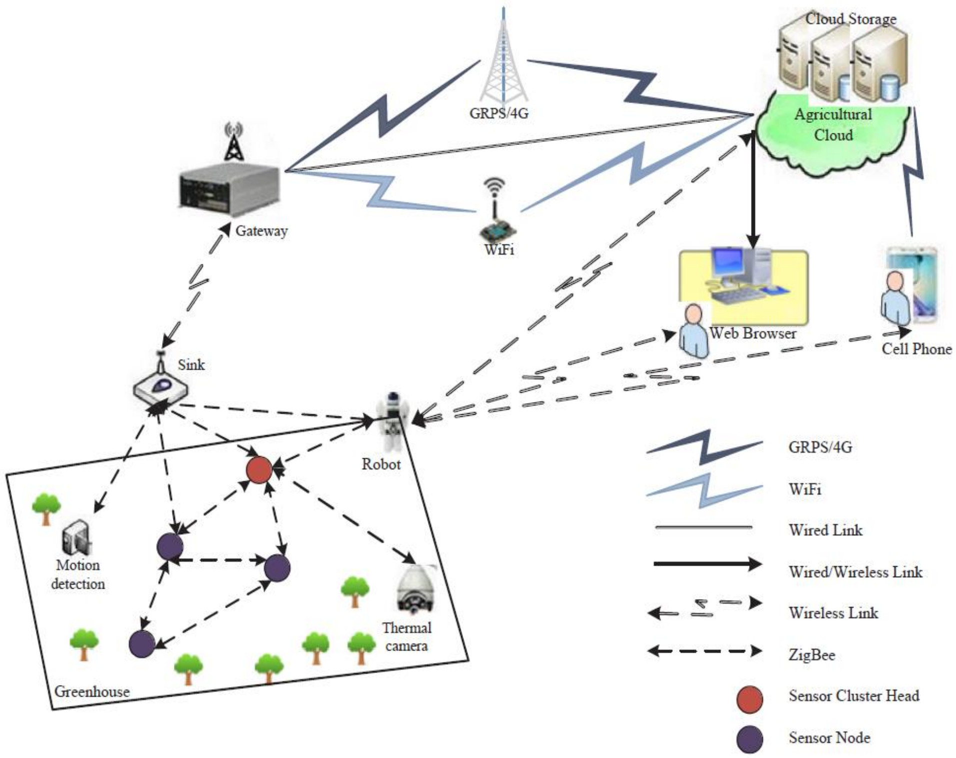


Figure 2: Framework of Agricultural CPS (Courtesy: Guo, Dusadeeringsikul, & Nof, 2018)

Cyber-Physical Systems are traditionally defined as integrations of computation with physical processes, where embedded computing devices and networks integrated with feedback loops monitor, control and support physical process (Lee, 2010). CPS integrates the dynamics of the physical processes with those of the software and networking, providing abstractions and modeling, design, and analysis techniques for the integrated whole. CPS applications encompass a wide range of fields including medical device systems, automotive systems, environmental control, avionics, manufacturing automation, distributed robotic systems and communication technologies (Khaitan & McCalley, 2015). CPS are an important paradigm of Industry 4.0, which also includes technologies such as distributed/decentralized networks, intelligent robotic systems, human-robot interactions, smart wearable devices, virtual and augmented reality. Internet of Things (IoT) and Internet of Services (IoS) have vastly improved the effectiveness of CPS in real-time decision support and administration, virtualization of physical components and data collection. Intelligent Cyber-Physical Systems (ICPS) represent the next generation of advanced networked systems, presenting distributed intelligence.

The current Industrial Revolution driven by CPS and IoT is expected to have a major impact on the future of agriculture as well, as there is a natural relation between industry and agriculture (Blanchet, Rinn, Von Thaden, & De Thieulloy, 2014). Agricultural applications can be modelled as a Cyber-Physical System (CPS), owing to the multi-dimensional nature of agents and communications involved, from physical agents such as human operators, mobile platforms and cyber agents such as the sensors, computational algorithms and protocols. Precision Agriculture is considered an application of an Agricultural Cyber Physical System (CPS) consisting of three layers: physical, network and decision layer (Rad et al., 2015). According to an EU Agricultural and Rural Development Study (Dumitrache, Sacala, Moisescu, & Caramihai, 2017), "Precision Agriculture (PA) is a whole-farm management approach using information technology, satellite positioning (GNSS) data, remote sensing and proximal data gathering. These technologies have the goal of optimizing returns on inputs whilst potentially reducing environmental impacts." Over the last decade, there has been a rise in the number of non-invasive remote sensing techniques which are sensitive, consistent, standard, high-throughput and cost effective. The most popular non-invasive techniques are: fluorescence spectroscopy, Visible/Near-Infrared (VNIR) spectroscopy, fluorescence imaging, and hyperspectral imaging (Sankaran, Mishra, Ehsani, &

Davis, 2010). In precision agriculture, remote sensing used widely to inspect and map variability in open fields is not suitable for protected crops.

2.2 Sensor Networks

One of the new trends emergent in the field of agriculture is the rise of sensor networks, synonymous with the development in precision agriculture. A sensor network has three basic functions: 1) Sensing 2) Communication 3) Computation using hardware, software and different algorithms (Aqeel-Ur-Rehman, Abbasi, Islam, & Shaikh, 2014). PA focuses on using software tools to observe, monitor and control agricultural practices (Romanov, Galelyuka, & Sarakhan, 2016) in production networks. Important aspects of PA include monitoring of soil, crop, climate attributes; Decision Support Systems (DSS) for preventive or prescriptive action on issues such as disease propagation and quarantine. Sensors are used to collect and integrate information of physical and environmental attributes within controlled environments (such as greenhouses). They serve as an efficient system of data acquisition over a wide range of attributes, some including temperature, moisture, water/air flow, salinity. Monitoring soil characteristics such as nutrient and water levels are paramount for optimal growth of crops, while temperature and humidity need to be measured and controlled to prevent suboptimal conditions, incubation of pathogens and pests. Within the ARS, the sensor network is also used to increase the information required to make decisions about certain inferences, since external factors can be a pressing issue in the onset of stresses in plants.

2.3 Agricultural Robotics and Simulation

Robots in agriculture have been conceptualized from the early 80s: namely, automated machinery such as unmanned combine harvesters, pesticide spraying robots, fruit harvesters, and weed control robots (Kawamura & Namikawa, 1988). Automated and robotic systems are can enable seamless, low-risk data collection and interaction in agricultural environments for physical tasks such as sensing, imaging and scanning. According to (Libin et al., 2008), an agricultural robot exhibits the following features: (1) operation with fragile and complex objects; (2) operation in unstructured environments; (3) complexity of overall task; (4) cost of production, use and maintenance. Because of the advancement in sensors and robotic technology, the idea of applying robots in unstructured environments such as fields and assistance of other agents such as human operators and sensors has become more feasible (Belforte, Deboli, Gay, Piccarolo, & Ricauda Aimonino, 2006). Monta & Namba, (2003) discuss a three-dimensional sensing system for agricultural robots to capture precise external information about different fruit plants. As automation and robots are good at repetitive tasks and able to work for more extended periods, they are generally responsible for routine operations such as irrigation, harvesting, inspection, and to help or reduce the reliance on farmers for tedious and arduous tasks (Reid, Zhang, Noguchi, & Dickson, 2000). Also, in extreme weather conditions, robots with smart technology can work with reliable results (Pedersen, Fountas, Have, & Blackmore, 2006). Currently, there is a need for effective early disease detection techniques to control plant diseases for food security and sustainability of Agro-ecosystem. In cases of disease or infection tracking and scanning in plants, having humans interacting with the plants can create safety and health concerns; it is more plausible to use a robot to perform the scanning and detection in such cases. Hence, automation and robots are beginning to proliferate within agriculture systems.

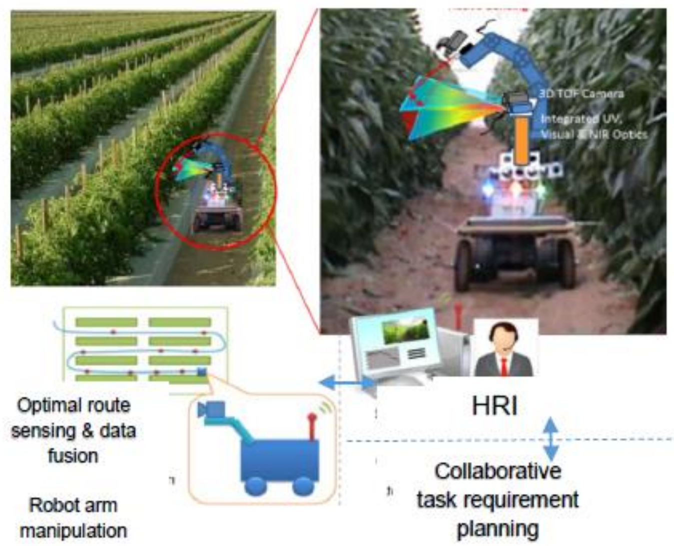


Figure 3: Agricultural robot in the field (BARD project report, 2016)

Advances in simulation technologies and virtual environments, affordable computational devices have cascaded the growth of agricultural robotics. Digitized collection of plant and environment data, integration with sensors and control technologies have enabled simulation platforms to virtualize a large range of agricultural scenarios at low cost and high scalability (Shamshiri et al., 2018). For example, fruit harvesting using robots require a large number of experimentations due to the delicate nature of the produce, susceptibility to environmental conditions and sufficient quality requirements (Bechar & Vigneault, 2016). Physical experimentation with different robots and sensors in an actual field is thus not always possible due to time constraints, malfunctioning equipment and physical labor. Simulation thus provides an affordable method to verify different configurations and mechanisms of such robotic systems, with the primary objective of fusing real-world and virtual data to derive new knowledge. Advantages of agricultural simulation include 1) reduced costs and shortened testing time, 2) easier diagnostics and debugging, 3) versatility of configurations, 4) avoiding risks and hazards to humans and environment. Shamshiri et al. (2018) discuss and compare the performance of various simulation platforms that have been adapted for agricultural robotics, including Webots, Gazebo, Actin, MORSE, V-REP and OpenHRP3. Even though thorough research has been conducted using these softwares for agricultural robots, the research in integrating humans in the loop remains in its nascent stages, and hence it is necessary to develop simulations to understand the nature of these integrations.

2.4 Human-in-the-loop Systems

Recent years have seen large developments in the fields of robotics and autonomous systems. While the complexity of performable tasks by robots has increased, there still exists many tasks where robots show a considerable gap in performance when compared to humans. Complex tasks such as object detection under uncertainty (Branson et al., 2010), terrain navigation require properly designed collaboration between humans and robots. Such systems are termed "human-in-the-loop (HITL)": multi-agent systems with optimized human-robot interaction and collaborative intelligence. Previous literature on this topic includes human-robot interactions for target recognition (Bechar, Meyer, & Edan, 2007), human decision modeling (Stewart, Cao, Nedic, Tomlin, & Leonard, 2012), robots querying humans for help (Holzinger, Plass, Katharina, Gloria, & Cris, 2019), assistive technologies (Bringes et al., 2013). Current approaches work asynchronously in connection with Human Agents (HAs) who are expected to provide input in terms of decisions and data interpretations. Leveraging human cognition and perception can improve the versatility of complex tasks, help in reducing errors and conflict with dynamic environments, and drive productivity. Bechar & Edan, (2003) also reported the advantages of Human-in-the-Loop Systems for melon detection tasks, which the HITL on average showing higher detection rates and lower detection times as compared to fully autonomous and manual systems.

Human performance is not perfect, and errors can be modelled probabilistically along with HA fatigue levels (Winkelhaus, Sgarbossa, Calzavara, & Grosse, 2018a). Fatigue is generally considered a closet term for a variety of factors affecting human awareness, attention, error-proneness, boredom, workload and performance – both physically and mentally, and across all levels of expertise. Known as a contributor to human error generation (Jaber & Neumann, 2010), errors caused by fatigue are observed in a wide range of industries such as aviation, healthcare, automotive. Modelling fatigue appropriately is an important factor in HITL systems, as we do not want to exceed the permissible limits of Human-Robot Interaction (HRI) and risk increasing chance of error and loss of productivity, along with safety constraints for HAs (Munir, S., Stankovic, J. a, Liang, C.-J. M., & Lin, 2013). The limits of HRI (or HITL Level) can be defined by the amount of time the HAs interface with the system; limited by time, physical and mental constraints, increasing interactions often result in reduced performance, increased chance of error. This is an important measure in high-risk systems such as machine operation, maintenance, emergency protocols – and it needs to be ensured that the HAs interface for the optimal amount of time, as long as their benefits can overbear their disadvantages. This research considers the cognitive effects of fatigue with increasing workloads, specifically for classification tasks.

With increasingly complex workflows, testing different HITL configurations can be done affordably and extensively using simulations. Industry standard simulators for driving, aviation and military purpose are fairly well established, including matured user-interface design and psychology studies (Driggs-Campbell, Bellegarda, Shia, Sastry, & Bajcsy, 2014). Agricultural simulators such as V-REP, Gazebo and ARGoS, though efficient in visualization, computation and cross-platform connectivity – are traditionally focused on hardware meshing and computation.

There does not seem to be a lot of research on studying the “human” augmentation into CPS, especially when human inputs can provide timely decision support and error detection. For agricultural systems, HAs are usually trained inspectors tasked with sampling, inference and physical movement between locations. Appropriate task taxonomies should be studied to understand the range of tasks which can be performed by the HAs in the CPS context. In order to do so, it is necessary to accomplish the following: 1) Develop a robust HA model with appropriate task scope for integration into the system and 2) Explore different integration workflows. Bechar, Edan, & Meyer, (2006) develop different levels of Human-Robot Collaboration (HRC) based on the four degrees of autonomy (Sheridan & Verplank, 1978), with each level corresponding to a unique HRC workflow to elucidate the different possible integrations of humans in the loop. This research attempts to deploy realistic HA models within simulations of agricultural tasks.

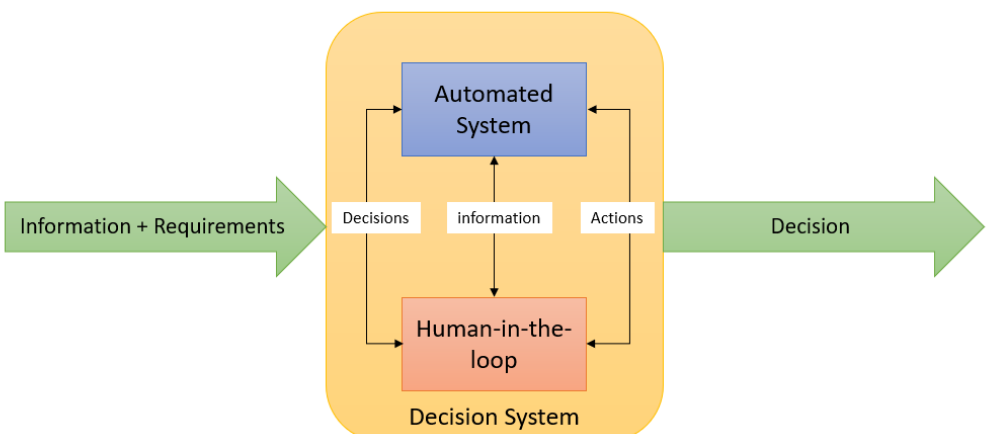


Figure 4: Human-in-the-loop System

Based on precision agriculture, sensor networks, agricultural robotics and HITL, an agricultural robotic system (ARS), described in (Guo et al., 2018) is a multi-agent cyber-physical system comprising of humans, an autonomous robotic cart and a sensor system. Human Agents (HAs), such as farmers, agricultural experts, software experts are tasked with real time decision support while the purpose of the mobile robot (Robotic Agent (RA)) is to maneuver to specific locations within a greenhouse to inspect and scan plant samples with the help of the sensor system. ARS aims to combine the strength of multi-agents such as humans, robots, and sensors to improve overall system performance. Since there are many collaborating entities, it becomes necessary to gauge the collaborative intelligence of the system and ensure that inter-agent interactions are efficiently managed for the common goal.

2.5 Collaborative Intelligence

Collaboration is the sharing of information, resources and responsibilities between distributed agents to plan, implement and achieve individual and common goals of the combined system (Zhong, Levalle, Moghaddam, & Nof, 2015). In environments where information and decision flows are varying continuously, efficient collaboration can ensure that the right information is presented to the appropriate agents at the right time.

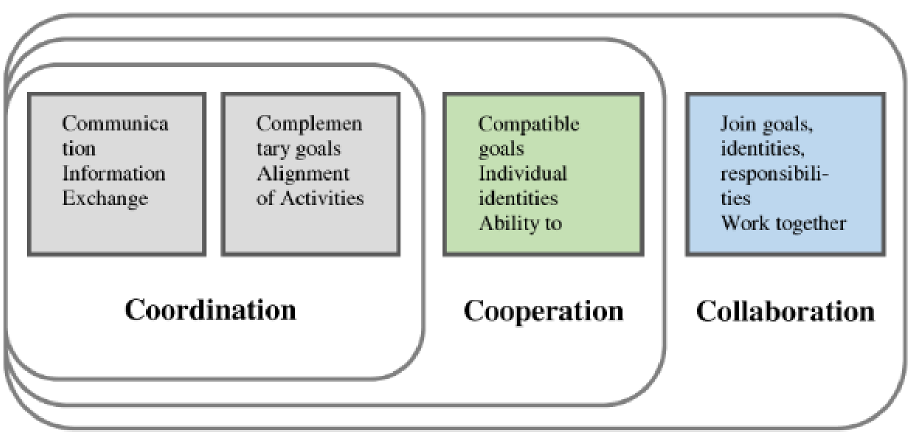


Figure 5: Coordination, Cooperation and Collaboration (Velásquez & Nof, 2009)

Given complex systems, individual agents must control information in a robust manner to ensure simplified working of the system. Collaborative Intelligence (CI) is a measure of an agent's ability to interpret new information, share resources and information with other peers to resolve new local and global problems in a dynamic environment. Most modern industrial systems are largely modular and involve some form of Collaborative Intelligence. For example, managing order fulfillment (Bhargava, Levalle, & Nof, 2016), task assignment in multi-robot systems (Zhang, Zhong, & Nof, 2015) – both these examples leverage CI to optimally match resource and task agents. Increasingly internetworked and layered systems have necessitated the design of cyber-supported collaboration, which extends the definition of collaboration to computational and autonomous agents. CI is an essential tenet of Collaborative Control Theory (CCT) (Nof, 2007), a theory developed to understand, design and optimize collaboration support systems, collaboration protocols and algorithms. CCT defines several key principles to augment and enables organizations to achieve their goals. From the context of e-Factories, or e-Systems, the principles of CCT are pivotal in streamlining interactions. They have been actively applied in diverse use-cases including heterogeneous multi-robot systems (Zhang et al., 2015), facility sensor networks, virtual manufacturing (Moghaddam & Nof, 2017) and multi-enterprise collaboration. It can be argued that the ARS is a similar heterogeneous multi-agent system, and hence the CCT principle can provide vital insight into workflow management.

This research studies cyber-supported collaboration for an agricultural context, where efficient collaboration between robotic, software and human agents can enable early detection of infections and stress in plants and initiate appropriate preventive measures to prevent further spread. Using the tenets of Collaboration Control Theory, Collaboration Intelligence is leveraged to obtain improved workflows.

2.6 HUB-CI

There have been several collaborative tools designed for knowledge and information sharing (Durugbo, 2016). Inspired by HUBZero (Mclennan & Kennell, 2010), a tool developed by researchers at Purdue University to facilitate online collaboration between researchers to develop simulation tools, HUB is an online platform to connect and provide users with computational support tools. HUB-based CI is the algorithm logic used to develop and strengthen the functioning of the HUB in managing not just agent interactions, but protocols and computational algorithms (Devadasan et al., 2013). The main objective of HUB-CI is to enable agents to identify the following: (1) what information to contribute; (2) which tools to use; (3) with whom to collaborate. The key innovation of HUB-CI is in its ability to enable and facilitate physical and virtual collaboration between several groups of human participants, along with relevant cyber-physical agents (Zhong, Wachs, & Nof, 2014) while pre-existing HUBs were limited to virtual interactions between agents (Mclennan & Kennell, 2010). This is of particular importance to HITL Systems, where decision and information flows are shared virtually (between robots and sensors, or between algorithms and agents) and physically (between humans and robots). HUB-CI logic can be emulated for ARS to provide cyber-supported collaboration, thus dictating the task sequences of the system. As shown in Figure 6, HUB-CI manages and governs cross-domain interactions, including inputs from human experts, sensors, computational devices, robots and knowledge bases.

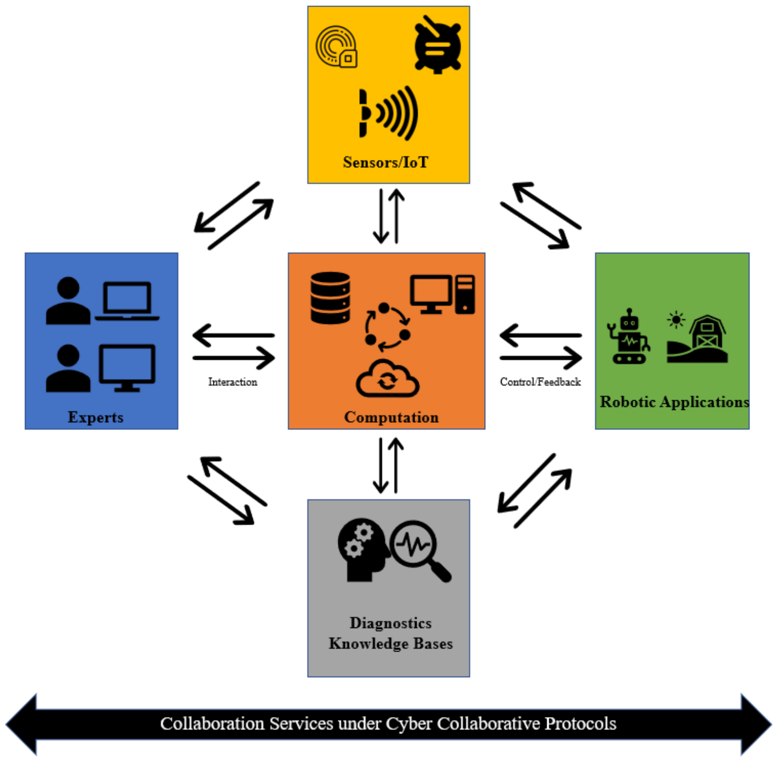


Figure 6: HUB-CI Domains

This research reports the use of HUB-CI logic to develop a simulation platform for the communication and interaction of different agents from an Agricultural Robotic System. Its logic governs as a central controller responsible for the distribution of information and decisions between human, robot and sensor agents, in an optimized and harmonized manner, and it manages information from three central algorithms: Sampling/Routing, Adaptive Search and Human-in-the-loop, which are explained in further in Section 3. HUB-CI logic is also used to develop a physical simulation platform (software) to study the ARS Human-Robot workflows, which is explained in Section 3.5.

2.7 Summary of Literature Review

Table 2: Summary of Literature Review

Research Question	Literature reviewed
1. What simulation models and technologies can we use to improve the productivity of current Agricultural Robotic Systems?	(Bechar & Vigneault, 2016; Driggs-Campbell et al., 2014; Guo et al., 2018; Ramin Shamshiri et al., 2018; Staranowicz & Mariottini, 2011)
2. Cyber-Supported Collaboration: - How do we use Cyber-Supported Collaboration for Agricultural Robotic Systems? - Which collaboration protocols can we employ between agents to provide the optimal performance parameters?	(Dusadeerungsikul & Nof, 2019; Munir, S., Stankovic, J. a, Liang, C.-J. M., & Lin, 2013; Nof, 2007; Zhong et al., 2015, 2013)
3. Human-Machine Interactions: - How do we configure the system for Human-Machine Interactions? What considerations do we make while modelling system agents? - How do different Human-in-the-loop integrations affect the system? What are the different advantages and disadvantages to these integrations?	(Bringes et al., 2013; Chipalkatty, 2012; Emmanouilidis et al., 2019b; Jouppi, Schweigert, Sun, Su, & Srinivasan, 2015; Munir, S., Stankovic, J. a, Liang, C.-J. M., & Lin, 2013; Winkelhaus, Sgarbossa, Calzavara, & Grosse, 2018b)

The following list summarizes the learnings from the literature review

- Cyber-Physical Systems are complex integrations of computational, physical and virtual agents, communicating not only with each other but with different algorithms which optimize the working of the system. Agricultural CPS, specifically Precision Agriculture involve advanced sensing technologies, data collection and enhanced machinery to optimize agricultural production and account for variabilities.
- Automated machinery and robots in agriculture are used primarily to reduce human efforts and standardize agricultural processes, now aided by the advancement of sensors and robotic technology. Robots are efficient at repetitive tasks, showcasing precise and

consistent task performances and can hence replace humans in tasks involving manual labor.

- Autonomous systems reduce manual labor, but they show considerable gaps in task performance when compared to humans for certain tasks. Humans qualities such as cognition decision-making and perception cannot be reproduced by autonomous systems yet, and these qualities can augment autonomous systems which much needed versatility in dealing with dynamic and unstructured conditions. However, factors such as fatigue and fatigue-induced error in humans limit the potential of humans in such systems, and thus it is necessary to develop robust human models which incorporate all these relative advantages and disadvantages.
- With increasing diversity in agents, it is also necessary to consider the complexity of interactions, and the collaborative performance of the system. The system must be designed to optimize interactions between agents and algorithms, and for this purpose, HUB-CI logic is used to control workflows with collaborative intelligence. Efficient collaboration ensures that the right information, including inputs and decisions, reaches the right agents or algorithms at the right times.
- Improving simulation technologies and affordable computational devices have allowed the testing and validation of diverse cyber-physical systems, combined with real-world and virtual data. Simulation provides a platform to test out different configurations, parameters and architectures before physical deployment, thus saving on costs, time and providing safety. This research employs simulation techniques to test different human models, integrations for ARS.

- While prior research has focused on the computational/hardware side of CPS, this research focuses on a human-centric point of view: how can we effectively create virtual models of humans for integration, and how does different integrations affect the working of the system.

3. METHODOLOGY

3.1 Task Description

A cyber physical agricultural robotic system (ARS), described in (Dusadeerungsikul & Nof, 2019) is a multi-agent system comprising of humans (experts), an autonomous robotic cart, and a sensor system. HAs are tasked with real time decision support, while the purpose of the mobile robot (RA) is to effectively maneuver to specific locations within a greenhouse, to inspect and scan plant samples with the help of the sensor system. The robotic cart is mounted with remote sensing equipment and vision systems to inspect plant attributes for the early detection of stress, potentially causing diseases. Early stress detection is attempted to enable precise response and disease prevention. It is infeasible to sample every single plant in the greenhouse, hence, a sampling strategy is required to sample specific plants, which represent the local area, based on chosen heuristics and data obtained from sensors. Once a sample list is created by the software agents, the information is conveyed to the robotic cart, which then maneuvers to the location for scanning and stress/disease inference. Based on scientifically established disease behaviors and their pathological relation to environmental parameters, it is possible to predict the direction of spread of certain diseases (Dusadeerungsikul & Nof, 2019). Adaptive Search Algorithm described in Section 3.3 applies this information with current data obtained from sensing equipment, to search in the vicinity of an infected plant and localize the extent of the disease potential spread.

Information collected from the location can help predict the status of the plant, which can be conveyed to the HAs for further verification and initiation of preventive measures. It must be noted that decision support is required, since the robot motion occurs in an unstructured and unpredictable environment, and any potential for error and conflict must be mitigated in real time.

This challenge is one of the reasons humans are required in the loop: Human cognition and versatility can be leveraged to improve dynamic responses of such systems. In many cases, the visual data obtained from the robotic cart can provide unpredictable data for software algorithms, which can be augmented by HA knowledge bases to improve the disease predictive strength of the integrated robot-human system. HAs are considered to be able to leverage their cognitive capability to make inferences on plant status (infected or not infected). These actions and decisions enhance the versatility of the system, as it is leveraging the knowledge base of both the autonomous vision and scanning system, as well as the expertise of human agents such as farming experts, agricultural scientists. For e.g., in some cases the signs of disease are observable and hence can be diagnosed immediately by the HA to also initiate countermeasures to prevent the spread of the disease. In order to test the inclusion of humans in the loop, a simulation-based approach is considered for the provided task description. By simulating the ARS, the scope of human control can be varied with different configurations, which can be tested and evaluated.

Thus, the cyber physical ARS is a multi-agent system of collaborating agents and algorithms, with each agent communicating information and decisions based on the underlying algorithms responsible for administering the workflow. The integration and scope of HITL simulations within the ARS to measure the impact on KPIs and to detect the optimal levels of collaboration are analyzed. The objective is to detect the early onset of diseases in greenhouse plants using the best combination of given agents

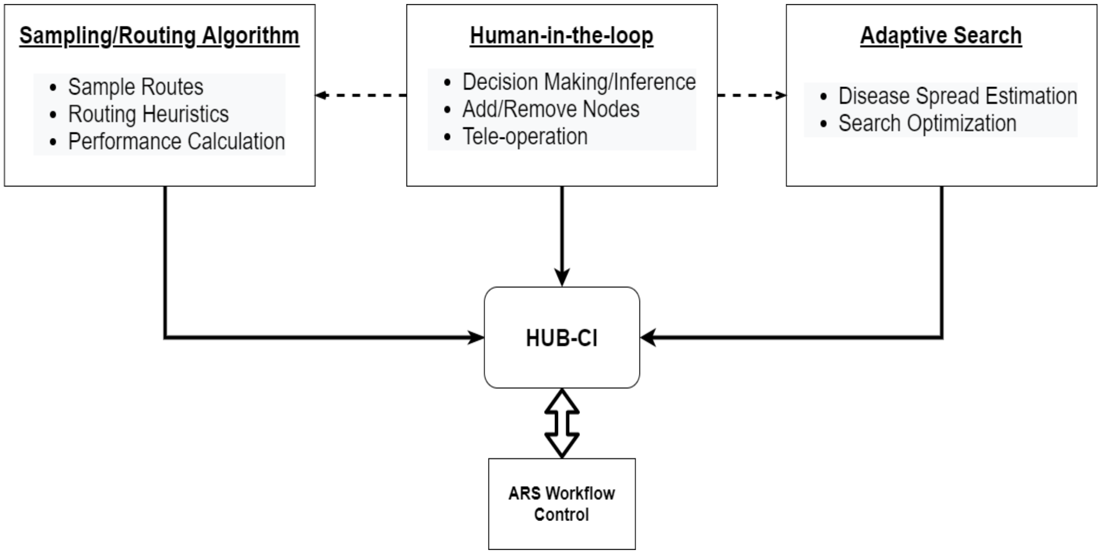


Figure 7: HUB-CI for ARS Workflow

3.2 Sampling and Routing Strategies

Modern stress monitoring involves physically visiting and scanning plants using non-destructive technologies such as hyperspectral or multispectral imaging. Traditional stress monitoring systems in greenhouses, however, are largely manual: a trained inspector scouts the plants on foot within the plot, usually sampling a few locations at each plot, and the sampling is usually visual. The sampling locations are determined arbitrarily, following a random heuristic. Constrained by large plots, limited human resources, time restraints and high costs, sampling rates even for trained operators are low, while visual sampling might prove insufficient for the early detection of diseases. Usage of robotic agents (RAs) can help farmers execute wiser decisions, improve sampling rates by shifting the bottleneck from the inspectors, but due to added costs it essential to employ sampling strategies or algorithms which try and approximate probable locations, thus improving detection while reducing redundant samples.

Biotic (diseases, infections) or abiotic (overfertilization, insufficient water etc.) stresses are common sources of non-ideal conditions in commercial plants. The spread of stresses or diseases in commercial crops often can be pinpointed to a few infected plants, which can occur from either insect vectors (Fereres & Raccah, 2015) (such as aphids, whiteflies), or non-insect vectors (such as contaminated seeds, soil etc.). These initially infected plants can act as an inoculum source for propagation of the disease within the plant structures, thus making it imperative to detect these stresses as early as possible. To estimate optimal sampling spots in a structured plant grid, a sampling algorithm is employed which uses pre-determined disease parameters such as initial chance of disease diagnosis, and chance of plant-to-plant propagation to estimate high-risk locations and add them to the sampling queue.

Once the list of sampling locations is determined, the robot must visit these locations in an efficient manner. This is a variant of the Travelling Salesman Problem (TSP), generate the shortest path between the set of sampling nodes. Since the plants are structured as a grid, the L1 Norm, or the Manhattan/Taxicab distances are considered for the routing calculations.

$$\boldsymbol{L1\ Norm}: \left|\left|\boldsymbol{z}\right|\right|_{\boldsymbol{1}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} |\boldsymbol{z_i}| \qquad (\boldsymbol{1})$$

A greedy algorithm, called Nearest Neighbor algorithm, is a constructive heuristic used to find shortest path between the locations. As the locations are limited and well ordered (not sparsely distributed), the NN algorithm works efficiently to create the shortest path. The static nature of the sampling list and route is supplemented by the Adaptive Search algorithm, a search tool described in Section 3.3 works in tandem with the sampling and routing algorithms to add critical nodes into the path in real time based on the plant inference, ensuring that the sampling remains dynamic.

An example of a sampling run is shown in Figure 8. The plants are shown in the green highlight, while the yellow highlighted spots are the locations to be sampled in this run. "t" represents the time taken by the robot to move between locations, while "s" and "s`" represent the time taken to complete a scan by the robot and human expert, respectively. The red lines represent the tasks which can be sampled by the human for this run (not all locations can be sampled – based on availability of human experts).

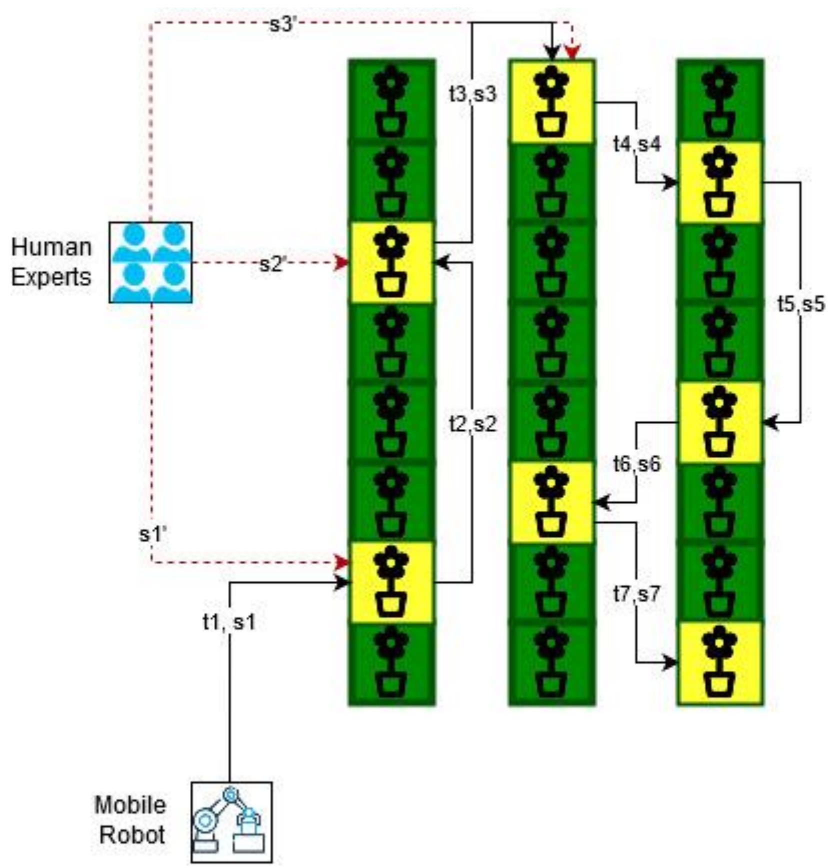


Figure 8: Sampling/Routing algorithm

3.3 Adaptive Search

Gongal et al., (2015) establish the need of preventive and predictive maintenance in a timely manner of agricultural crops. As mentioned earlier, a few infected plants are often the cause of further disease propagation in structured plant grids. For certain diseases, the infections or stresses usually propagate in scientifically predictable directions, often influenced by external factors such as sunlight and airflow. For example, if a plant is infected with Botrytis, it spreads masses of gray diseased spores with may be picked up and carried on air currents and transported to healthy plants, which become infected. Plants affected by Phytophthora are susceptible to the disease via root-to-root spread, while diseases such as Tobacco mosaic virus (TMV) spread easily via direct contact between plants, or even by workers who have been in contact with the virus. It is necessary to consider disease propagation while sampling plants. Appendix A has some preliminary research in detecting disease type based on the direction of propagation using Neural Networks.

The Adaptive Search algorithm, as reported in (Dusadeerungsikul & Nof, 2019) is an sensing algorithm used to indicate the extent or severity of a disease in plants in a greenhouse. It has been developed to extend the sampling space to include locations with high risk of stress, based on the inference from a completed sampling. It is dynamic in nature, and depends on the real-time conditions: hence, it cannot be inherently designed in the sampling algorithms. Depending on factors such as the type of disease detected, type of plant sampled and environmental conditions, the search region for AS varies. To elaborate for this research, if a plant is found to have a disease based on the sampling data, AS is initiated to add the locations of plants located in the cardinal directions to the plant, or combinations of the directions depending on the external factors. In this article, we consider an adaptation of AS where the plants positioned North and South of the plant in question are added to the sampling list. By initiating and performing AS, HAs can also understand the severity of the damage caused by the disease, and hence develop a mitigation strategy to prevent further spread. AS ensures that if a disease is detected, appropriate quarantining is undertaken to measure the extent of the spread from the disease, as seen in Figure 9.

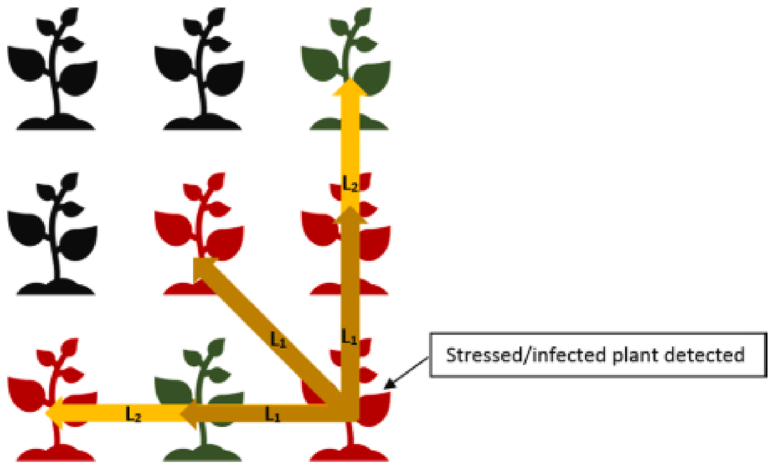


Figure 9: Adaptive Search Algorithm (Dusadeerungsikul & Nof, 2019)

3.4 HITL Interaction Modelling

3.4.1 Human Agent Modelling

The major challenges in integrating HAs include understanding the range of applications of HITL control, or namely the different taxonomies based on the controls that HAs employ, modeling human behavior of various types and identifying the optimal modeling schemes for each type (Munir et al., 2013). When developing HA models, it is necessary to make appropriate assumptions and distinctions between HAs and RAs.

Robotic Agents (RA) (and Autonomous Agents) perform well in industrial environments which are structured and have predictable events (Bechar et al., 2006). They showcase significant stability in repetitive tasks without decrease in performance, which is not the case with HAs, where repetitive tasks are possible causes for physical and mental fatigue, both of which have been reported to reduce individual performance (Winkelhaus et al., 2018a). Errors caused by fatigue have been observed and studied in aviation, transportation, manufacturing – often risking human safety – and this are one of the primary reasons why HITL Simulators are required (Jaber, Givi, & Neumann, 2013). However, as per Bechar et al., humans have advanced cognitive capabilities, can easily adapt to changing environments and acute perceptive skill, which current robots lack. Thus, the modeling of HAs remains a complex field with abundant assumptions. With regards to the challenge of modeling human behavior with ARS, the following attributes were considered and designed:

1. *Task performance*: For target recognition tasks, humans have superior recognition capabilities, due to their cognitive knowledge bases and better perception of physicality (depth, height, shape, color) (Bechar & Edan, 2003). Previous literature in comparing human and robot task performance (Genaldy, Duggal, & Mital, 1990; PAUL & NOF, 1979) suggest that robot on average took more time than humans to complete industrial tasks. There is not a lot of literature on the comparison of human and robot performance for scanning tasks, but it can be assumed that HAs can perform the tasks at a relatively better rate than the corresponding RAs. This relative advantage is achieved at a higher cost, since robotic costs are largely installation and maintenance; they can do tasks at lower costs than humans due to high degree of repeatability, lack of risky exposure. The RAs were developed to reduce physical load and provide automated and generic control, but they lack

versatility and cognition that HAs possess. For the scope of this research, the task performance parameters such as task time, classification accuracy and distance between plants are estimated from internal data populated from the BARD Grant and (Nair, Bechar, Tao, & Nof, 2019).

2. *Fatigue Models*: The concept of fatigue has been widely studied in the fields of ergonomics, physiology, psychology and sports. Fatigue can occur both physically and mentally, with both resulting in longer task times and high instances of human error (Jaber & Neumann, 2010). Ma et al., (2009) suggest that fatigue accumulates over time as more work is performed, and for this research, we assume the fatigue function to be linearly related to the number of tasks performed (similar models have been suggested by (Jaber & Neumann, 2010; Kaneko & Sakamoto, 2001; Soo et al., 2009). The corresponding error rate is also linearly dependent on the fatigue level of the HA. HAs are thus prone to relatively higher error rates and inferior task performance (longer time taken to service task) under increased task load based on pre-defined thresholds. The estimates of error rate for both robot and human are estimated from previous research (Nair et al., 2019) done within the BARD project. This research limits the fatigue to the cognitive domain, since HAs are performing cognitive classification tasks within the loop.

3. *Limited availability*: Unlike RAs, physical and mental constraints enforce timeliness constraints on HAs. They cannot be available throughout the functioning of the system, and thus HAs are assumed to be available for task assignment only for fixed time periods. These time periods, known as **HITL Level** of the system, correspond to the proportion of the total system time that HAs are available. When HAs are not available, the RAs continue system functioning individually without human support. By also varying the availability of

HAs, different configurations can be simulated, to locate the optimal configuration parameters for the ARS.

Table 3: Comparison of Agent Activities

Robotic/Autonomous Agent	Human Agent
Performance does not decrease with repetition	Performance decreases with repetition (time taken to execute task increases)
Fixed probability of error	Variable probability of error (depends on level of fatigue)
Higher scan/inference task time	Lower scan/inference task time
Lower cost	Higher cost

Schirner et al., (2013) also address the question of workflow control in such systems: which agent should control the system at a given time period; how do they delegate tasks and avoid decision conflicts – through a shared control/governance workflow which makes it possible for HAs and RAs to act in direct support of each other when required. To locate optimal modeling schemes and understand the spectrum of HITL control, we define and associate **task control** with the agent currently controlling the system: for e.g., if the robot performs and validates a scan on a certain plant, task control is said to be with the RA; if the HA provides the validation, task control is maintained by the HA. In ARS, the agent that provides the final decision inference on the status of the plant maintains task control. Task control is dynamic and switches between agents based on their availability. For a specific task, it can either be solely maintained by a single agent (Robot/Human scans individually) or shared by multiple agents (e.g. Robot can request Human for help). Based on these differences, two workflow types are simulated and compared: Sequential, and Shared Integration.

3.4.2 Sequential Integration

In this type of integration, Task Control is maintained solely by either of the agents, but not shared (not both). The term sequential is used to denote the sequential change in task control, between the agents performing the scanning tasks i.e. the robot or the human agent. The human agents are available for a fixed time period, and within this period are completely utilized to provide scanning inferences for all tasks. Figure 10(L) shows the flow of task control. Priority of scan task is given to the HA, if available, and otherwise solely maintained by the agent performing the task: if both RA and HA are available, the HA will be given task control. Since this type of integration requires available HAs to perform all tasks, it is useful to consider the effects of fatigue on task performance and error. It might be predicted that with increasing availability, the effect of fatigue will be more pronounced, causing a decrease in task performance. Thus, it also becomes necessary to find a theoretical optimum time period, which is denoted as **Optimal HITL Level (OHL)** for which the HAs can perform the tasks, since HAs perform a higher number of tasks they will incur higher instances of failures and penalty costs.

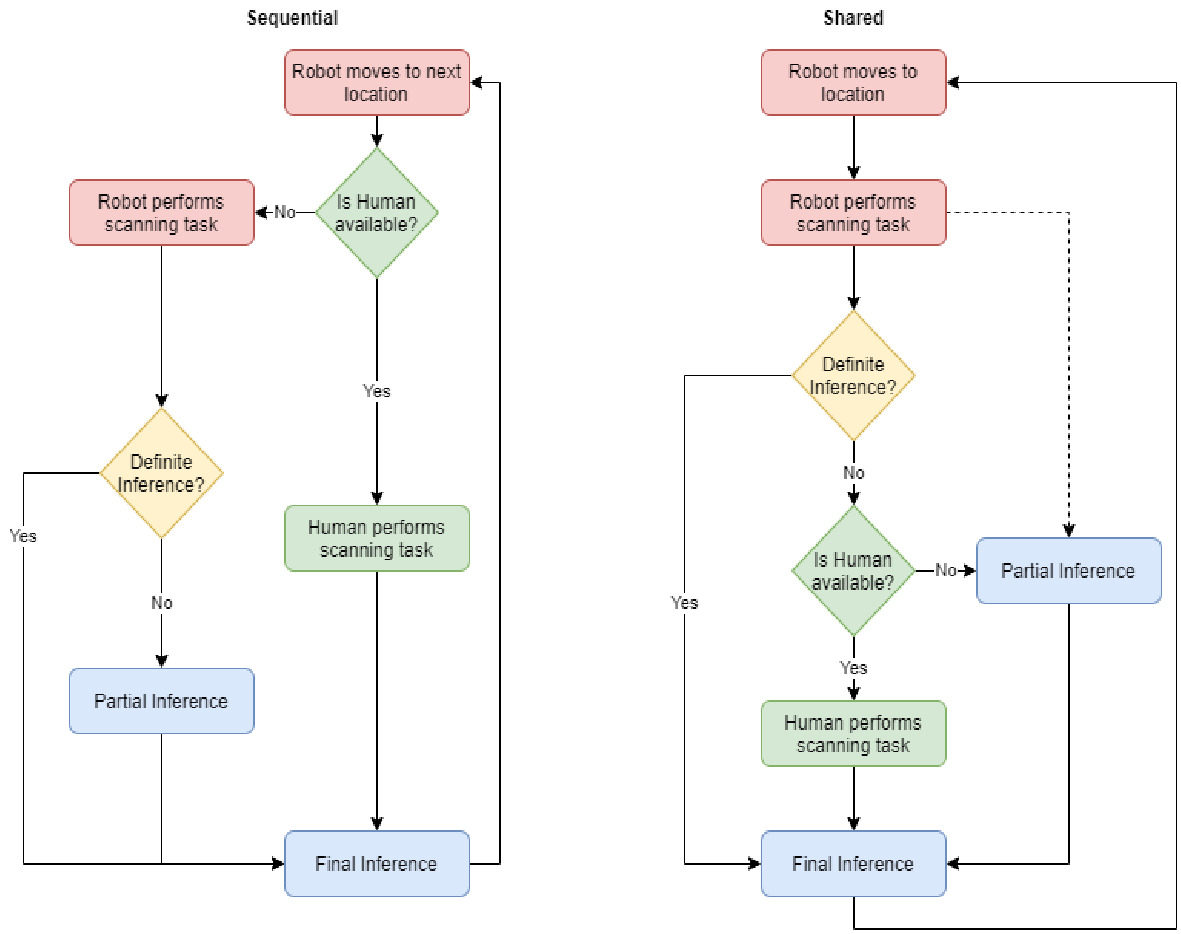


Figure 10: Sequential (L) vs Shared (R) Integration Workflows

3.4.3 Shared Integration

As opposed to Sequential Integration, Task Control in the case of Shared Integration is collaboratively distributed, with the ability of agents to collaborate with other agents when required. Agent availability does not correspond to task assignment, especially in the case of human agents (in sequential integration, an available HA has to take up task control). Task Control is now multi-layered depending on the complexity of the scan task: it is not necessarily individual, and multiple agents can control the same task. For e.g., if a result of the robot scan does not provide definite results, it can request help from the human agent when available; the human agent can provide their final inference based on trained expertise. By promoting shared task control, it is necessary to have adequate collaborative interactions in place to enable inter-agent communication for indefinite cases. As in the case of Sequential Integration, HAs are available for a fixed time period, and it is necessary to understand how the OHL is affected in this case. Figure 10(R) shows the workflow for shared integration, and also establishes the differences in both workflows.

3.4.4 Performance Metrics

Two types of performance metrics are calculated, based on costs and routing parameters. The variables for difference calculations are shown in Table 4.

Table 4: Performance Variables

α = *Distance normalization constant*
β = *Time normalization constant*
γ = *Penalty constant*
C_{move} = *Motion cost*
$C_{penalty}$ = *Penalty cost*
C_{scan} = *Scan cost*
C_{tot} = *Total cost*
D = *Total number of infected plants detected per run*
e_j = *error instance at* j^{th} *task*
i = *index of previous task completed*
j = *index of task being completed*
FN = *False Negatives per run*
FP = *False Positives per run*
MDR = *Missed Detection Ratio*
RSR = *Redundant Sampling Ratio*
S = *Total number of sampled plants per run*
$T_{infected}$ = *Total number of infected plants per run*
TN = *True Negatives per run*
TP = *True Positives per run*
t_j = *Scan time of* j^{th} *task*
x_i, y_i = *Coordinates of* i^{th} *task*

Cost Calculation

For each integration type, the overall costs incurred in the system due to the agents and their interactions are captured, as shown in Equation (2). Three reasons of cost occurrence are considered: motion, scanning and error penalties. Motion costs reflect the efficiency of the routing algorithms and are incurred solely by the RAs which traverse between different plant locations. Scanning and error cost parameters vary based on different agent-specific assumptions, since both RAs and HAs can perform these tasks. It is considered that HAs can complete a task at a faster rate relative to a robot agent, since they can leverage cognitive knowledge bases. Limited by physical constraints, the probability of error for the human agent depends on the fatigue level, which increases with the number of tasks performed by the human agent. Fatigue also effects task performance, defined by the time required for the agent to complete the chosen tasks (scanning/decision inference); with increased fatigue, task performance decreases (task time increases). The probability of error for the robot agent, however, remains constant throughout the simulation, since fatigue for robotic agents is not considered. Penalty cost is incurred at every error instance. The costs for different HITL Levels are captured for comparison across levels for optimal operation. As mentioned earlier, the coefficient values are derived from internal data estimates of the BARD grant (ARO).

$$C_{tot}^{j} = C_{move}\,(i,j) + C_{scan}\,(j) + C_{penalty}(e_j) \tag{2}$$

$$C_{move}(i,j) = \alpha(|x_j - x_i| + |y_j - y_i|) \tag{3}$$

$$C_{scan}(j) = \beta t_j \tag{4}$$

$$C_{penalty}(e_j) = \gamma C_{scan}(j) * e_j \tag{5}$$

$$e_j = \begin{cases} 1, & error\ occurs\ at\ j^{th}\ task \\ 0, & otherwise \end{cases} \tag{6}$$

To measure the effectiveness of the sampling, routing and Adaptive Search techniques, the following metrics are also developed. Each metric will also be affected by the HITL level, since we expect the number of tasks performed to vary with the HITL level.

Routing and Sampling Metrics

To measure the effectiveness of the sampling, routing and Adaptive Search techniques, the following metrics are developed and observed during the experiments for each complete run of the simulation. It is expected that HITL Level will have an effect on these metrics, since it will affect the number of tasks being performed in each run.

Total Sampled/Task Performance (S): Number of tasks performed by the HITL ARS for a single run of the simulation. Based on the inclusion and availability of HAs, this metric is a good indicator of whether the proposed workflow can sample more plants within the same time constraints.

Missed Detection Ratio (MDR): Percentage of missed infections which the system might have missed/cannot capture within the given run. This metric also showcases the performance of Adaptive Search as an effective search localization tool.

$$MDR = \frac{T_{infected} - D}{T_{infected}} \tag{7}$$

Redundant Sampling Ratio (RSR): Percentage of sampled locations which were unnecessarily sampled. Based on the sampling model for disease, this metric reflects the accuracy of the model in predicting the possible locations of the infected plants

$$RSR = \frac{S - D}{S} \tag{8}$$

Classification Observations: Since both RAs and HAs have a probability of error, it becomes necessary to track the classification accuracy of the system (overall, and for each agent individually). For each run, the sampling model estimates whether a plant has an infection (positive outcome), either inherently or from propagation, and compares this outcome to the outcome of the scanning task, which depends on the agent with task control. Thus, this gives rise to the following classifications:

- **True Positives (TP):** Sampling and scanning indicate positive outcome. Indicates proportion of correctly identified diseases.

$$TP(\%) = \frac{TP}{S} * 100 \tag{9}$$

- **False Positive (FP):** Sampling indicates positive, but scanning indicates negative outcome. Indicates proportion of incorrectly identified diseases.

$$FP(\%) = \frac{FP}{S} * 100 \tag{10}$$

- **False Negative (FN):** Sampling indicates negative, but scanning indicates positive outcome. Indicates proportion of incorrectly rejected diseases.

$$FN(\%) = \frac{FN}{S} * 100 \tag{11}$$

- **True Negative (TN):** Sampling and scanning indicate negative outcome. Indicates proportion of correctly rejected diseases.

$$TN(\%) = \frac{TN}{S} * 100 \tag{12}$$

3.5 HUB-CI ARS Software

Previous sections discuss simulation of different HITL integrations for the ARS, to study the impact of human involvement. This section reports the development of a prototype user interface based on the principles of HUB-CI for ARS. The HUB-CI-ARS controller software is developed to enable Human-Robot Collaboration within the ARS. It is built to communicate in real-time with an agricultural robot, via ROS, and also with Human Experts for dynamic inputs. The strength of the software lies in the ability to integrate human agents' decisions into the workflow to optimize the sampling and decision-support algorithms. The key features of the software are as follows:

1. Optimize/re-optimize sampling strategy for the robot based on the simulated propagation model
2. Initiate and use different modes of Adaptive Search as a heuristic tool for disease detection
3. Provide Human-in-the-loop (HITL) Collaboration through human decision integration in the system workflow
4. Simulate the entire experiment for different parameter settings using **Simulated Mode** to mimic different sampling and routing strategies
5. Collaborate and communicate with the robot using **Real Mode** (via ROS) in real-time

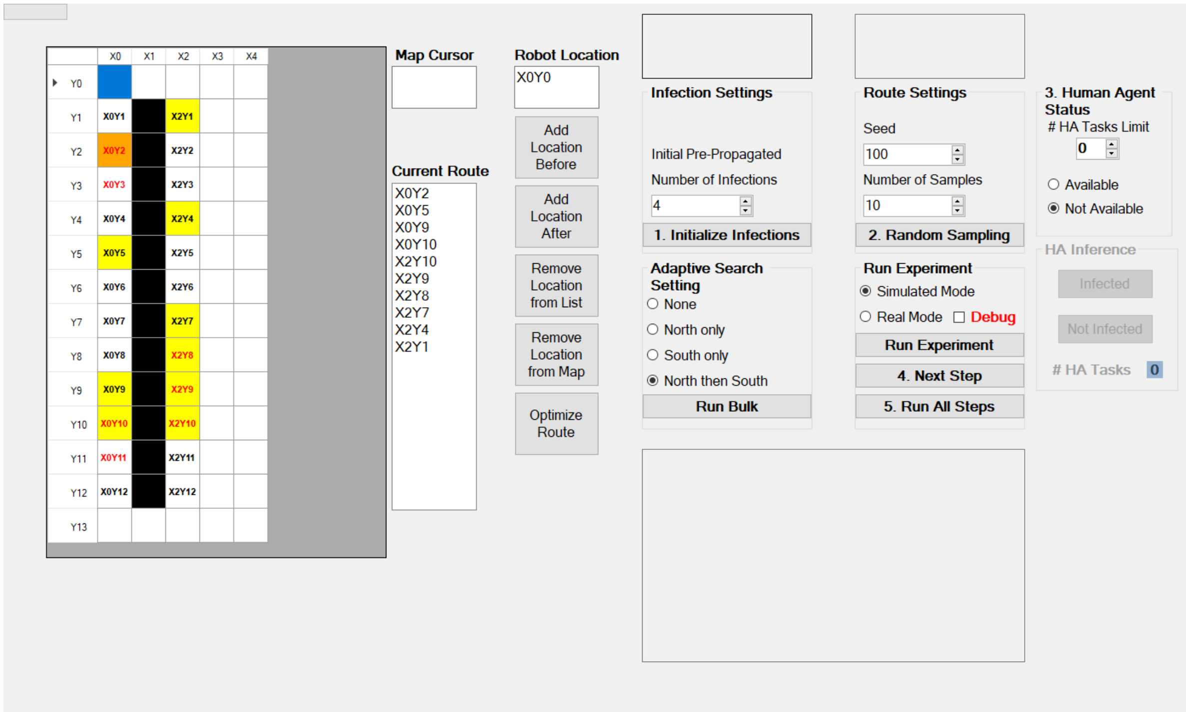


Figure 11: Snapshot of HUB-CI-ARS GUI (BARD technical report, 2020)

Figure 11 shows a screenshot of the interface. The software features and implementation are discussed in the following sections.

3.5.1 Software features

HUB-CI manages information from three primary algorithms: Sampling/Routing, Adaptive Search and HITL. The task description for the software is the same as described in Section 3.1. The software implementation of each feature is described below:

Routing

In order to optimize sampling and search of diseases, HUB-CI-ARS supports the following (Figure 12 shows the plant grid):

1. A grid of plants is provided in the display. Each plant has a location associated with it, and the use of highlight and colored text describe the current and future states of the sampling. The plant grid can be structured based on the actual placement of plants in the greenhouse.
2. A probabilistic disease propagation model to estimate spread of a disease within the plants (for simulation basis only). Probable infected places are displayed with red text color.
    a. Each plant has an inherent (native) probability to be infected
    b. Each plant can spread a disease (if infected natively) with a pre-defined probability
    c. Based on the above values, for each run of the simulation, the probabilities are reassigned, and the infected locations estimated
3. A sampling algorithm that determines the set of plant to be visited, based on the propagation model. The locations highlighted in yellow belong to the current sampling run, while the location highlighted in orange denotes the next location to be visited by the robot. This model interacts with the Human-in-the-loop model, since users can add and remove nodes from the sampling list based on expertise and prior knowledge
4. A routing algorithm to minimize traversing time using shortest path heuristics (Floyd-Warshall algorithm). Since users can add or remove nodes during the operation of the software, the Floyd-Warshall algorithm is employed so that the route dynamically optimized (based on L1 norm)

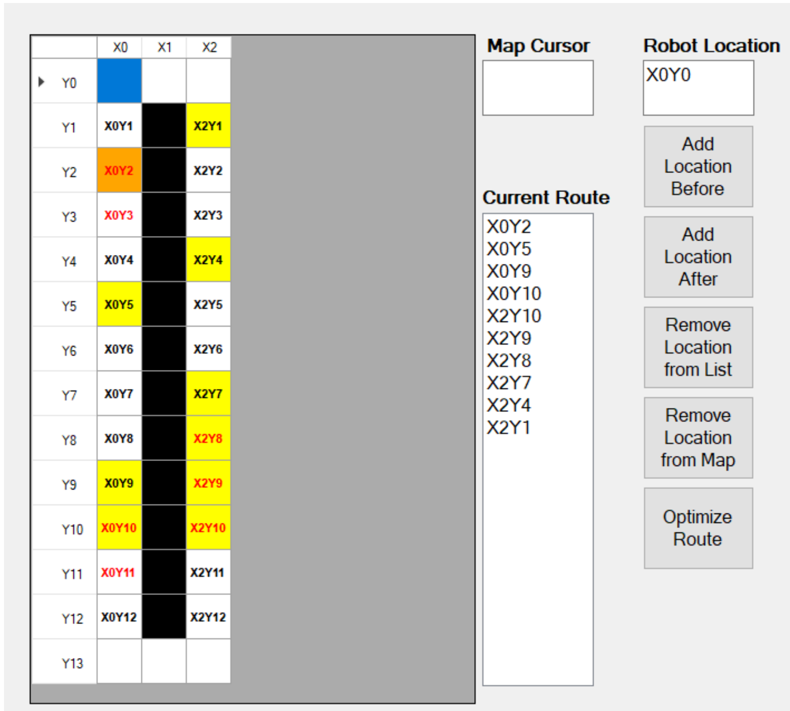


Figure 12: Routing and mapping (BARD project report, 2019)

Adaptive Search

In the HUB-CI software, the adaptive search algorithm is integrated at the red square in Figure 13.

1. Four options of adaptive search are available, namely:
    a. None
    b. North: If disease located, search north
    c. South: If disease located, search south
    d. North then South: If disease located, search north, then south
2. HAs can select one out of four options according to the stress or disease characteristics.
3. Based on the current inference, Adaptive Search is either initiated automatically or ignored.

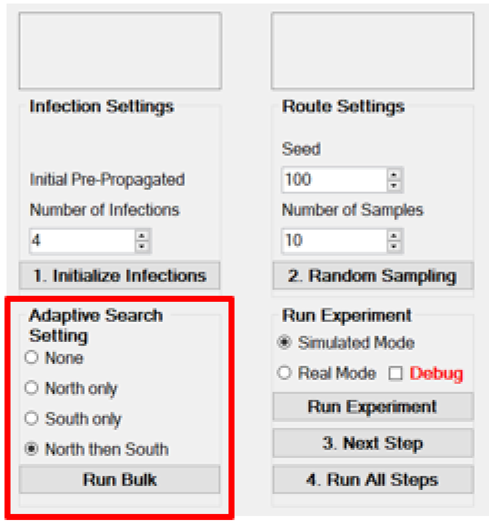


Figure 13: Adaptive Search Algorithm (BARD project report, 2019)

Human-in-the-loop

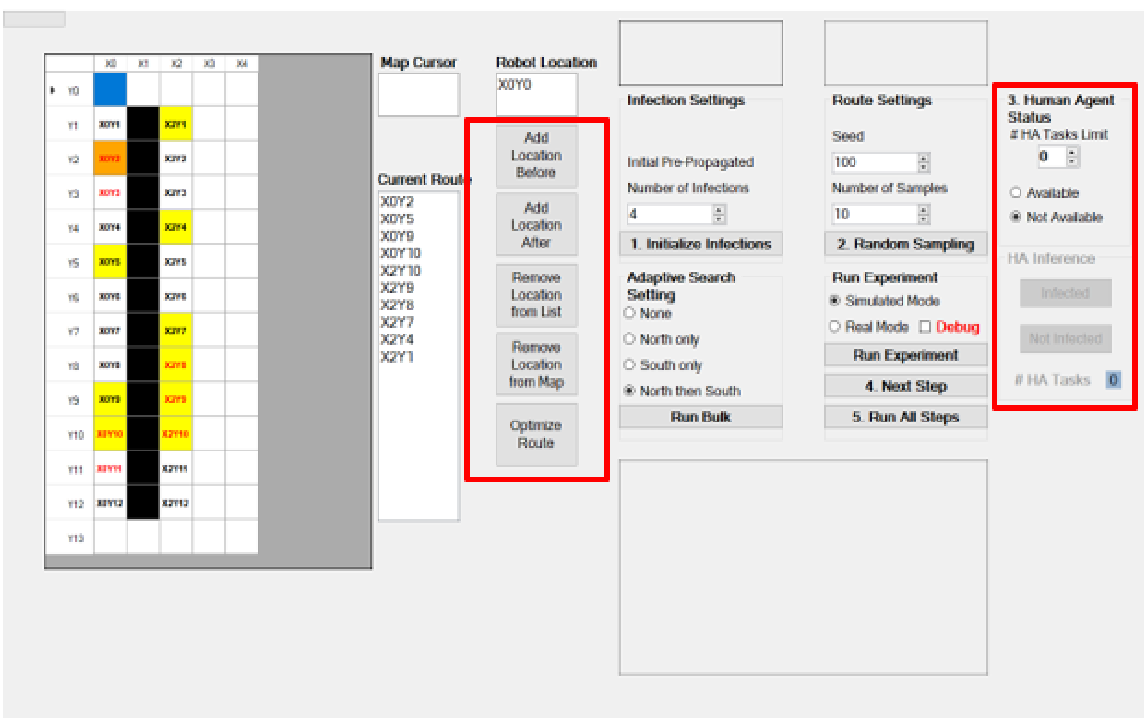


Figure 14: HITL Elements (BARD technical report, 2020)

HUB-CI-ARS enables sequential human-in-the-loop integration in the following ways:

1. Human Operators (HOs) can ADD, REMOVE locations and REOPTIMIZE the sampling set of locations to be visited by the robot
2. Domain expert HOs such as infection scientists can also provide decision inferences into selected cases (locations). HUB-CI-ARS enables direct human inference, where HOs can view the scanned data and determine the output.
3. HOs can interface for a limited number of tasks, since they cannot be present throughout the operation. The checkbox ensures that the variability in the availability of human agents is overcome, since domain expert HOs can update their availability in real time.
4. Once the task limit is reached, HOs can no longer intervene in the inference tasks and control is transferred back to the automated scanning and vision system
5. HOs can also set the appropriate Adaptive Search setting before initiating the simulation run.

3.5.2 Software Connectivity

The software has been developed in C#, which is a platform independent compiled programming language. HUB-CI ARS determines the next location to be visited by the robot, based on initially set parameters. Since the communication to the robotic environment is established via ROS (which is Linux only), it uses socket programming and automatically establish a two-way connection with a local Linux PC (ROS Publisher) which contains the ROS nodes real-time data communication. Data is shared via a publisher-subscriber system native to ROS (using ROS Messages and Topics), and data shared includes next location to be visited, current location of the robot. The ROS Publisher sends the next location information to the robot. An overview is provided below:

1. The HUB-CI ARS Client (C#) establishes a connection (two way) with the local ROS Publisher/Server and sends the next location coordinates to be visited (via socket programming) once Real Mode is initiated.
2. The ROS Publisher/Server (which is a publisher node) publishes (one way) the coordinates to the Robot ROS-Subscriber on the desired ROS topic
3. Robot ROS-Subscriber sends the coordinates to the ARS Robot, which sends the location confirmation to the Robot ROS-Publisher on reaching the location
4. Robot ROS Publisher publishes confirmation data to the local ROS Subscriber/Server on the desired ROS topic
5. ROS Subscriber/Server sends confirmation data to HUB-CI ARS Client (via socket programming)

Once the confirmation data is received, HUB-CI updates its internal agents and the proceeds to share the next location to ARO, on the basis on the inference from current location.

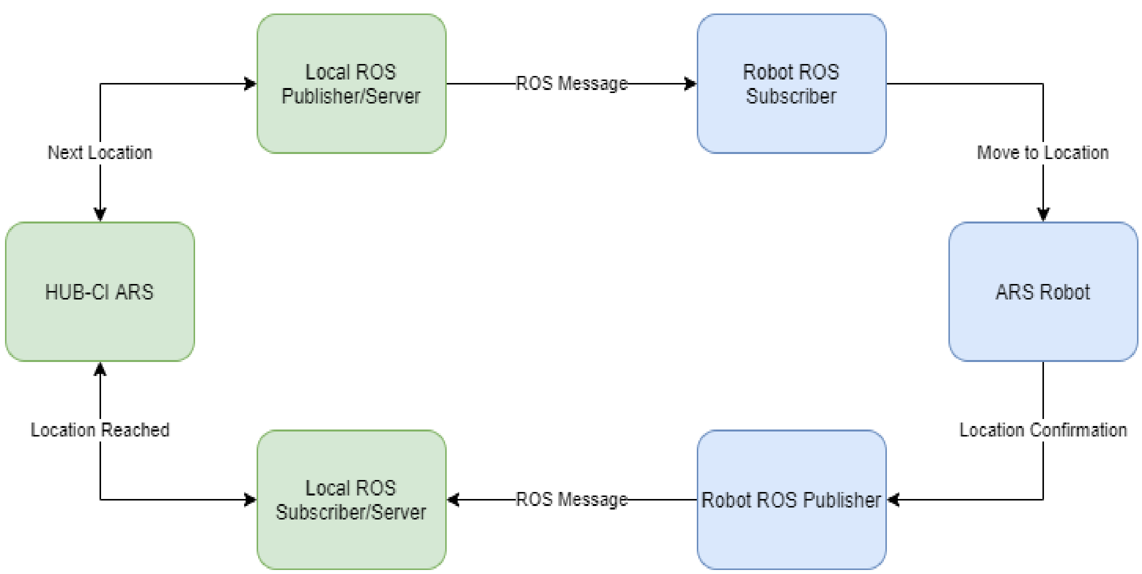


Figure 15: HUB-CI ARS Connectivity (BARD technical report, 2020)

3.5.3 Running the software

Setting Initial Conditions

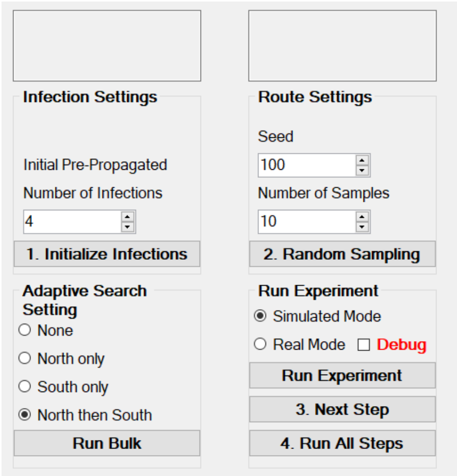


Figure 16: Setting initial conditions (BARD project report, 2019)

1. **Infection Settings:** Initial number of infections (probability of infection). HO can set the number based on historical data. Required to generate sampling list.
2. **Route Settings:** Initial number of samples. HO can choose the optimal number of sampling required for that run. Generates sample list for the current run.
3. **Human Agent Status:** Number of tasks to be performed by the Human Agent. This is based on the HITL integration study
4. **Adaptive Search Setting**: Choosing which type of adaptive search sampling we require based on domain knowledge and historical data.
5. **ADD/REMOVE/OPTIMIZE**: Once the sampling list and AS directions are generated, the HO can also add/remove and optimize the new list based on expertise.

The parameters can be set based on these initial conditions. Changing the seed changes the sampling list, while changing the adaptive search setting changes the final number of samples required.

Run Mode

Once the initial parameters are set and the route is optimized, we can run the experiment. There are two different run modes depending on the purpose of the experiment.

Simulated Mode

In Simulated mode, we can run virtual simulations of the system by using assumptions to predict the final inference of the plant status. Connection to the robot and robot inferences are virtually assumed. Within simulated mode, we have the following options to run the simulation:

1. **Step-by-step:** By using the "**NEXT STEP"** button, we can follow the experiment for each consecutive node in the sampling route. The **CURRENT ROUTE** will tell us which node is next and it will also mention if Adaptive Search is active based on the sampling results.
2. **Run All Steps:** Run all steps at once to view the performance metrics for that run. This is usually used for performance evaluation and is a quick way to measure if the inclusion of certain node selection protocols by the HO is effective or not.
3. **Run Bulk:** In this mode, we perform the experiment for a large number of runs to average out the results and provide a more standard metric formulation.

Real Mode

This mode is used to communicate with the actual robot in real time, sending location information via ROS publishers and waiting for the robot to send final location confirmation.

1. Within Real Mode, the experiment can only be run Step-by-Step.

2. When Real Mode is initiated, the software first establishes connection with the PRISM ROS Publisher via socket connections. Real Mode cannot be initiated if this connection does not exist.
3. On the PRISM ROS Publisher, the ROS node must be active and ready connect to the HUB-CI Client and publish the data.
4. Once Real Mode is set and the connection is established, HUB-CI will share the next location in real time, either decided by the sampling algorithm or the available HO. Once the robot receives the location, it will move to the location and send a confirmation once it has reached the destination.
5. Optimally, the robot will also send the scanning inference to be updated in HUB-CI ARS. If the task is being completed by the HO, then the robot will send real-time camera data so that the HO can make an inference and update the software manually.
6. HOs can also add or remove nodes during any point of the operation. They can choose to reoptimize the routing queue once they add/remove a node. These operations occur in real-time along with the decision inferences (within permissible number of tasks)

4. EXPERIMENTS

4.1 Phase 1: HITL Simulations

In order to understand the dynamics of each integration, two simulation experiments are designed to analyze and evaluate respective metrics across different HITL Levels. Simulation models have been used extensively as a low-cost solution to test different environments. Various simulation tools such as V-REP, Gazebo, AgROS and Webots (Shamshiri et al., 2018) are extensively used to test agricultural robotics. AgROS is such a tool used specifically as a decision support tool that enables evaluation of robots in 3D unstructured environments emulating real-world agricultural fields (Tsolakis, Bechtsis, & Bochtis, 2019). However, these tools are specifically dedicated for the integration of agricultural robotics into real-world environments, and none of them are specifically designed to evaluate the integration of HAs into such robotic systems. Thus, the simulation reported in this research is developed using an object-oriented programming (OOP) approach in Python, and it follows a discrete-time, event-based architecture. An OOP based approach allows us to dynamically model humans in the simulation by providing flexibility in defining attributes such as fatigue and chance of error.

The objective of these experiments is to identify critical configurations from different HITL integrations which can enhance collaboration, examine the effect of HITL Level (or time spent by the HA interfacing with the system) on key metrics such as cost, total tasks performed in the system in order to create hybrid and optimal ARS workflows.

4.1.1 Experiment Design

The simulation is designed for the task description provided in Section 3.1. Within the simulation, an event is defined as any of the motion or scanning tasks occurring as part of the task sequence. While the time required for motion tasks depends on the distance between the current and next location of the robot, the time for scanning tasks is vary based on the agent performing the scan. As stated in the assumptions, HAs can perform scans at a faster rate than RAs, and hence the average scan time for a HA is lesser than the scan time for a RA. The cost relations are provided in Section 3.4.4.

A single run in the simulation consists of 100 time steps, and the number of tasks (events) completed in the given time frame for a fixed HITL Level are monitored. For e.g., if the HITL Level is 30%, HAs are available and can perform tasks (such as scanning, decision inference) for 30 timesteps of the overall 100 timesteps. The nature of task control depends on the HITL integration chosen. By varying the amount of time the human interacts with the system (within these 100 time steps), the HITL Level is varied and the performance metrics for each case are collected. When the HAs are available (depending on the HITL Level), they perform tasks within the system with active fatigue models to monitor the fatigue and error levels after each task. For each fixed HITL level, the experiment is repeated 1000 times for different initial seeds and the results are averaged out for statistical significance. Different collaboration protocols are considered, to compare the performance of traditional agricultural practices to the proposed HUB-CI protocol – with the aim validating the potential of sampling, routing and AS algorithms in modern agricultural systems. The following simulated collaboration protocols are considered:

1. Random (baseline): The set of locations to be visited are generated at random for a chosen HITL Level. This corresponds to traditional sampling methods used in agricultural currently.
    a. Without Adaptive Search
    b. With Adaptive Search
2. HUB-CI: The set of locations to be visited are based on the sampling algorithm, with Adaptive Search (simulating the cyber collaborative protocol of HUB-CI) for a chosen HITL Level.

Based on the above description, the following 2 experiments are performed.

Experiment 1: Simulating Sequential Integration for the different simulated collaboration protocols and different HITL Levels

Experiment 2: Simulating Shared Integration for the different simulated collaboration protocols and different HITL Levels

4.2 Phase 2: HUB-CI ARS Software

Phase 2 tests the working of the HUB-CI ARS software, for both simulated and real mode. By setting different initial conditions, multiple sample runs can be executed to test the efficacy of the primary algorithms. Also, the real-time communication with the robot can be tested, with the robot sending a confirmation for accurate receipt and arrival to the desired sampling location. This experiment also expands the range of tasks performed by the HAs, since human input for sampling is not included (as it cannot be simulated) in Phase 1. HAs can provide decision inference, and the limit to the number of tasks performed by the HAs can be set based on the inferences from Phase 1.

5. RESULTS

5.1 Phase 1: HITL Simulator Results

5.1.1 Experiment 1: Sequential Integration

Sequential Integration relies on completely utilizing the available agents to complete the required tasks. True and false classification observation (Section 3.4.4) graphs for each HITL Level in Experiment 1 are shown in Figure 17. The parabolic nature of the graphs suggest the existence of point of inflection in each graph, coinciding approximately at the same HITL Level. This is the first evidence of an of an optimal HITL Level (OHL) for this integration, since at these points of inflection the classification accuracy of the system seems to be optimal. To elaborate, at this OHL (approximately 45%-55% HITL Level), the true positives and negatives both are maximum, while both false positives and negatives are at a minimum. A nonlinear regression analysis performed on the classification observations suggests that the plots (Figure 17) follow a polynomial curve of order = 2 (the multiple R-Squared value for TP, TN, FP, FN curves are 0.95, 0.94, 0.88, 0.92 respectively for an order 2 polynomial fit).

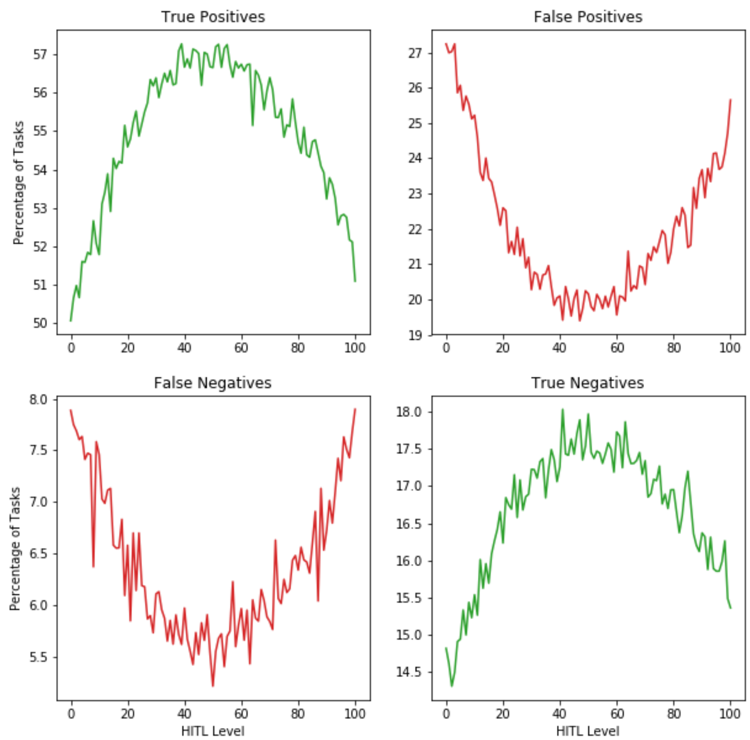


Figure 17: True and false classification observations for Experiments 1: (a) True Positives, (b) False Positives, (c) False Negatives, (d) True Negatives for HITL Level

In Figure 18, the task performance (a) and costs (b) (section 3.4.4) for different simulation schemes over different HITL Levels are plotted. From Figure 18(a), the difference is task performance (tasks sampled/performed) is visually evident, and the HUB-CI protocol outperforms the Random protocol (with and without AS). This can be attributed to the compounded impact of the three central algorithms of HUB-CI: Sampling, AS and HITL, which augment each other to improve the number of tasks performed. Comparing task performance across HITL Levels, the Wilcoxon signed rank test between the HUB-CI protocol(M = 15.18, S = 0.37) and Random (baseline, with AS) protocol (M = 13.08, S = 0.49) indicates the relative superiority of the sampling algorithm (W = 10201, $p < 2.2E\text{-}16$). A similar test between Random (baseline, with AS) protocol (M = 13.08, S = 0.49) and Random (baseline, without AS) protocol (M = 10.52, S = 0.38) verifies the productivity increase due to Adaptive Search (W = 10201, p < 2.2E-16). To attribute improvements due to HITL, we compare the HUB-CI protocol performance for two fixed HITL Levels: 45% (M = 15.44, S = 0.07) and 0% (M = 14.06, S = 0.06); the results (t(196.65) = 143.56, p < 2.2E-16) provide strong evidence to showcase the benefits of HITL augmentation.

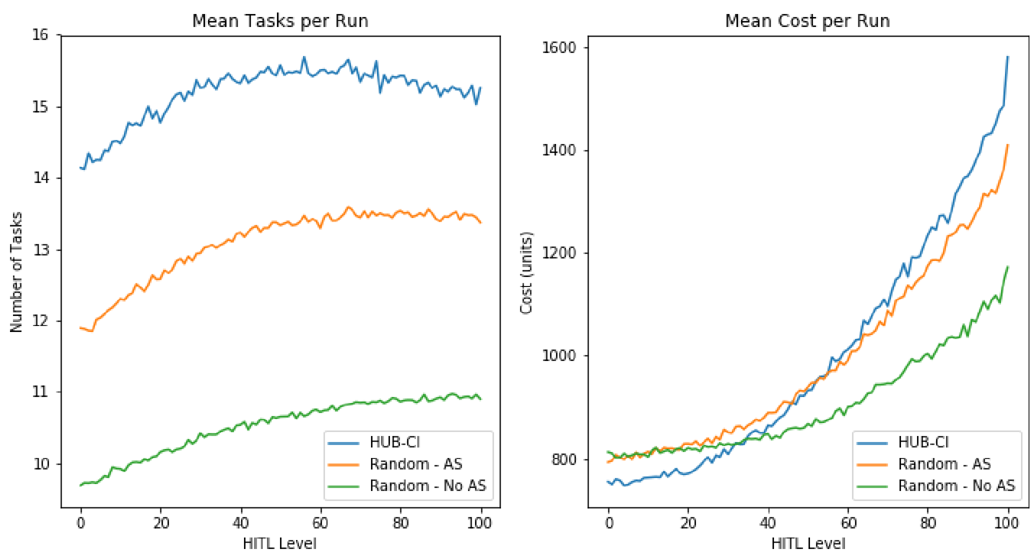


Figure 18: (a)Task(L) and (b)Cost Performance(R) for different simulated collaboration protocols

Figure 18(b) shows the costs of each collaboration protocol for different HITL Levels. A one-way ANOVA test (F (2) = 9.026, p = 0.0001) determined that the difference between costs are statistically significant. The cost trend seems to increase polynomially (order = 3, Multiple R-Squared = 0.99) with increasing HITL Level, but since motion and scanning costs are directly related to the number of tasks performed (concave downward trend as shown in Figure 18(a)), this increase can be attributed to penalty costs occurring due to higher instances of errors, and also increased costs from lowered task servicing due to fatigue. An interesting point to note is that at the OHL (45% HITL Level), the costs for each protocol remains similar, but the task performance at the same level shows large differences, indicating the advantages of the HUB-CI protocol. Table 5 shows the performance metrics at the OHL to elaborate the previous statement.

Table 5: Performance Metrics for HITL Level = 45

Protocol	**Number of Tasks**		**Cost**		**RSR**		**MDR**	
	Mean	*Std Dev*	*Mean*	*Std Dev*	*Mean*	*Std Dev*	*Mean*	*Std Dev*
HUB-CI	**15.44**	0.07	897.82	6.75	**43.28**	0.32	**33.10**	0.52
Random (with AS)	13.27	0.04	911.17	4.75	59.63	0.53	60.04	0.58
Random (without AS)	10.458	0.03	**851.65**	4.86	64.66	0.55	72.99	0.38
*Difference**	16.35%*		1.4%*		33.06%*		44.87%*	

*Note: Difference between HUB-CI and Random (with AS); statistically significant ($p < 0.0001$)

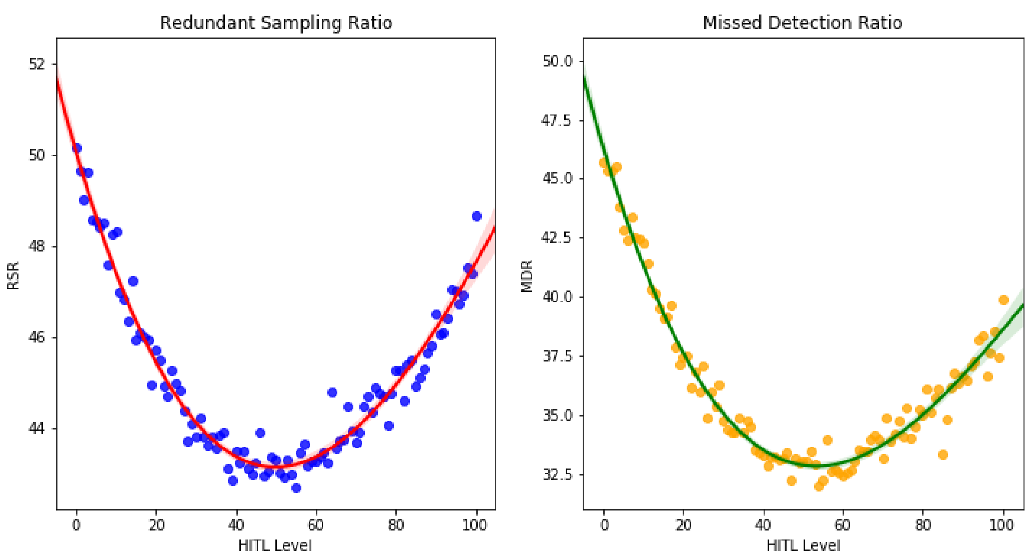


Figure 19: RSR and MDR for HUB-CI

Redundant Sampling Ratio (RSR) and Missed Detection Ratio (MDR) for different HITL Levels are plotted in Figure 19. The concave nature of both plots provides more evidence to support the existence of the OHL, with the local minima of both plots coinciding approximately and the formerly stated OHL (45%). At this level, the benefits of HITL are maximized and beyond this level negative effects due to human error are prevalent, as reflected in the decrease in overall system accuracy and performance. In Figure 20, the MRD and RSR for different collaboration protocols is over different HITL Levels is plotted. It is clearly evident that the HUB-CI protocol is more efficient and productive when compared to traditional sampling methods.

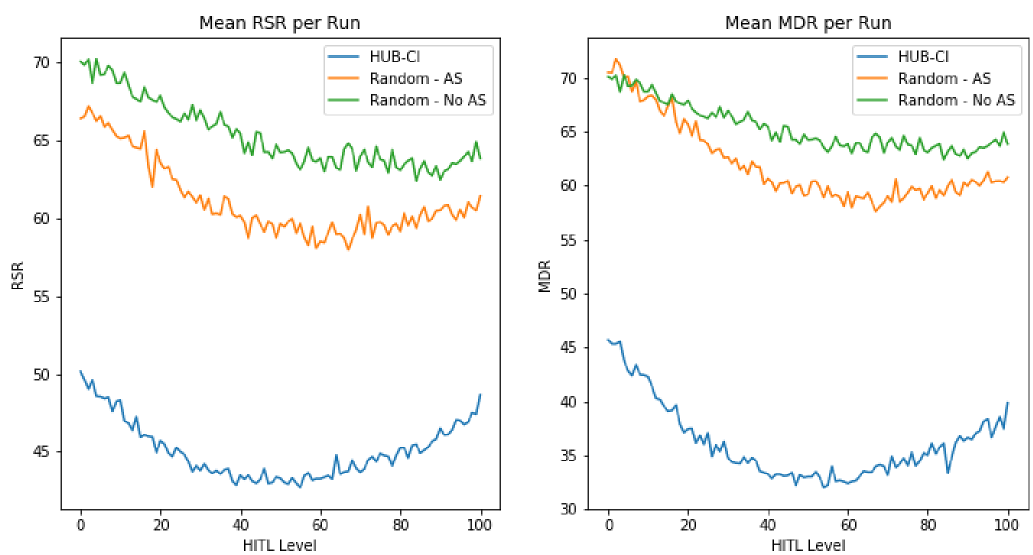


Figure 20: RSR and MDR Comparison over a range of HITL Levels

5.1.2 Experiment 2: Shared Integration of HITL

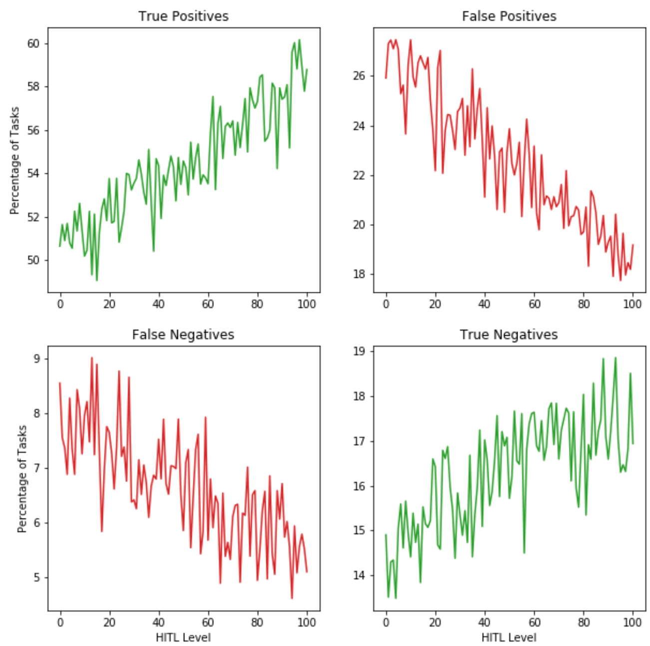


Figure 21: True and false classification observations for Experiments 1: (a) True Positives, (b) False Positives, (c) False Negatives, (d) True Negatives for HITL Level

Experiment 1 tests the implications of maximizing agent utilization; it however establishes that increasing tasks can contribute to a rise in costs and lowering of performance, attributed to fatigue and higher instances of error in HAs. Experiment 1 also establishes the superiority of HUB-CI as an effective protocol, when compared to the other collaboration protocols. In Experiment 2, the collaborating agents share task control and agent utilization prioritizes necessity over availability. Tasks are shared only when the RA cannot provide definite results, and when the HA is available. The true and false classification in Figure 21, unlike Sequential Integration do not indicate the existence of an Optimal HITL Level for this integration (order = 1, Multiple R-Squared = 0.92). Shared task control ensures that HAs only get tasks when necessary, and this reduces the impact of errors on task classification. For e.g., in Figure 21(a), true positives increase with increasing

HITL Level: increased HITL Level does not necessarily mean that the HAs perform more tasks, but that they are more available to provide decision inference on tasks which could not be classified initially (by the RAs).

It is expected that the impact of increased costs due to lowered task performance (with fatigue) and penalty costs due to error in HAs will be less prevalent, since now they augment the workflow in only critical cases. Figure 22 plots the task and cost performance for each collaboration protocol, and it is evident that HUB-CI protocol is relatively the most productive (Difference (M) = 2.1 tasks). The decrease in mean tasks per run can be due to the increased availability of HAs in the loop, consequently increasing cumulative scan time. The cost performance provides an important observation: as opposed to sequential integration, the costs for HUB-CI protocol (M = 738.28, S = 5.27) decrease slightly with increasing HITL Level, and the differences between protocols are statistically significant ($F(2) = 57.62$, $p < 2.2E\text{-}16$). Initial costs can be attributed to robot error (penalty costs), since HAs are not available to provide decision inference at lower HITL Level. It is interesting to note that costs for both Random (AS) (M = 776.95, S = 11.86) and Random (No AS) (M = 780.09, S = 14.39) remain similar, and a Wilcoxon signed rank test confirms that mean costs are not statistically different ($W = 4379$, $p = 0.08$). The costs for HUB-CI remain statistically different – suggesting that the employment of sampling/routing algorithms is necessary to drive costs, while AS might not be as effective. However, task performance (Figure 22(a)) does not paint the same picture, and the distinction between Random (AS) and Random (no AS) in terms of tasks is clearly evident. A possible explanation for this can be that the offset in costs due to increased tasks is countered by lower motion costs incurred due to AS, as AS adds locations of plants close to diseased plants, while without AS the locations are randomized.

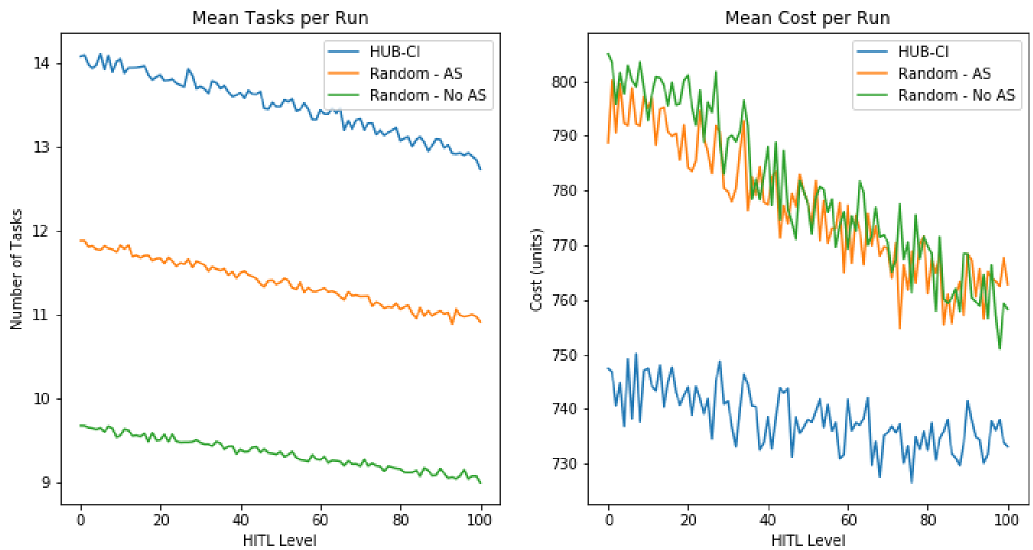


Figure 22: a) Task (L) and (b)Cost Performance (R) for different simulation protocols

The lack of impact due to fatigue is visible in the RSR and MDR metrics from Figure 23, which decrease monotonically with increasing HITL Level. The optimal (lowest) values for both RSR and MDR both lie at HITL% = 100, thus also stating that the positive impact of HA knowledge and scanning is not as impactful when compared to sequential integration. Figure 24 cements HUB-CI simulation protocol as a better sampling, routing and search protocol.

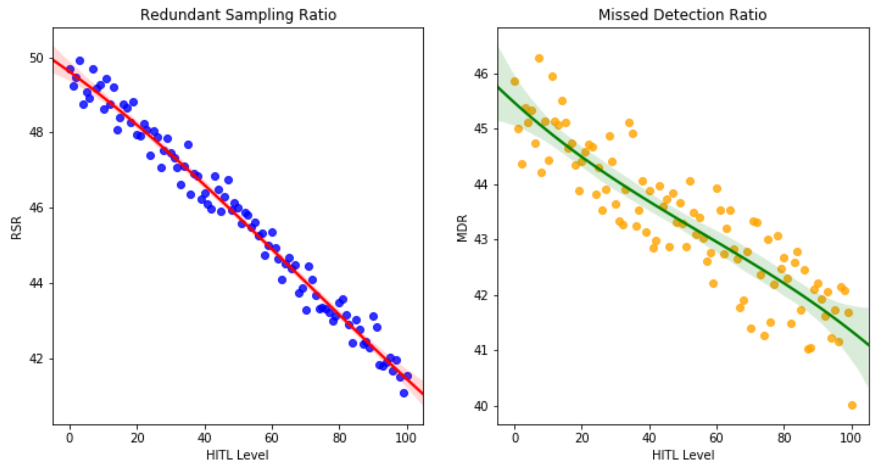


Figure 23: RSR and MDR for HUB-CI

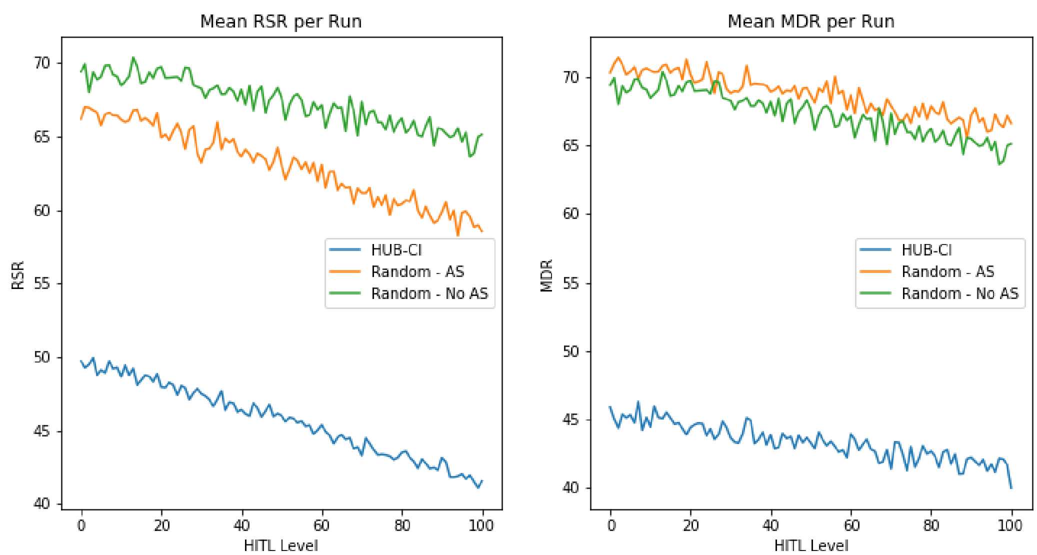


Figure 24: RSR and MDR Comparison

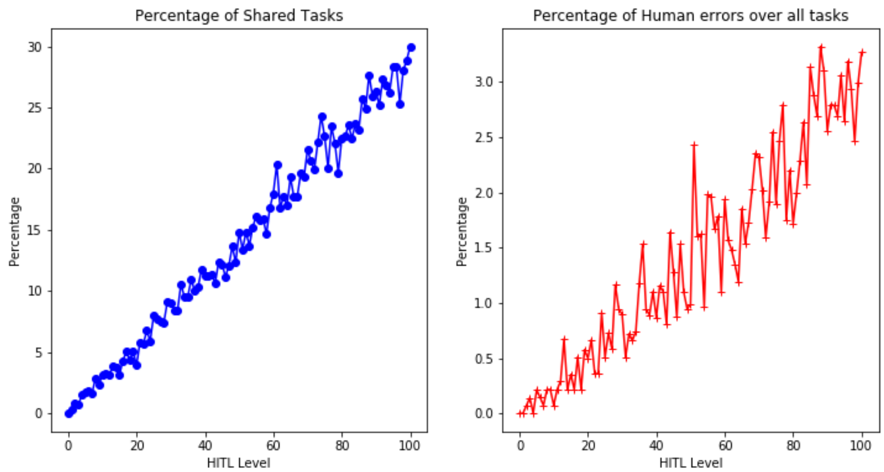


Figure 25: Factors contributing to increasing costs

Studying the statistics of shared task control, Figure 25 plots the percentage of shared tasks and the comparative percentage of average human error associated with different HITL Levels. HA errors increase but are constrained, which is reflected in the low overall costs of Shared Integration, with minimal contributions from HA penalty costs. This is explained further in Section 5.1.3

Table 6: Statistical tests performed

Null Hypothesis	Statistical test	Interpretation
Sequential Task Performance: HUB-CI = Random (with AS)	Wilcoxon signed rank test: W=10201, $p < 2.2E-16$	Sampling algorithm provides statistically significant improvement
Sequential Task Performance: Random (with AS) = Random (without AS)	Wilcoxon signed rank test: W = 10201, $p < 2.2E-16$	Adaptive Search provides statistically significant improvement
Sequential Task Performance: HUB-CI (45% HITL Level) = HUB-CI (0% HITL Level)	Welch two-sample t-test: $t(196.65) = 143.56$, $p < 2.2E-16$	HITL integration provides statistically significant improvement
Sequential Costs: HUB-CI = Random (with AS) = Random (without AS)	One-way ANOVA: $F(2) = 9.026$, $p = 0.0001$	Different in costs are statistically significant
Shared Costs: Random (with AS) = Random (without AS)	Wilcoxon signed rank test: W = 4379, $p = 0.08$	Don't reject null hypothesis. Difference in costs are not statistically different
Shared Costs: HUB-CI = Random (with AS) = Random (without AS)	One-way ANOVA: $F(2) = 57.62$, $p < 2.2E-16$	Different in costs are statistically significant

5.1.3 Cost Comparisons

An agent-based cost comparison was also performed to understand the nature of costs in each integration. From Figure 26, it is evident that with increasing HITL Level, the cost in Sequential Integration increases at a higher rate as compared to Shared Integration, except initially where the Sequential Integration shows lower cost per task. This is supported by Figure 27, where an agent-based cost-breakdown is provided. The stacked bar charts show that for Experiment 1, the contribution of HAs to the total cost dominates (due to increased penalty costs, lower task performance) over high HITL Levels, overshadowing the RA costs involved. This is not the case for Experiment 2, even though HA cost does increase, it remains nominal compared to the RA costs, suggesting that this integration is more cost efficient and does ensure that HAs are tasked appropriately to lessen the degree of penalty and performance costs.

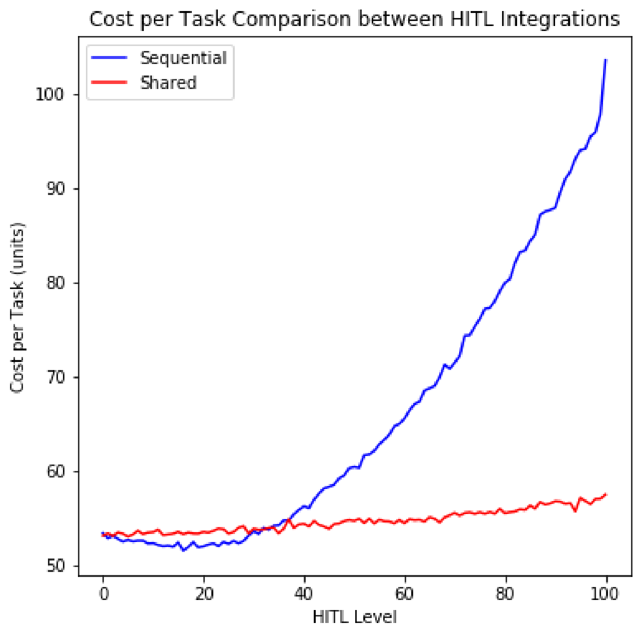


Figure 26: Cost Differences

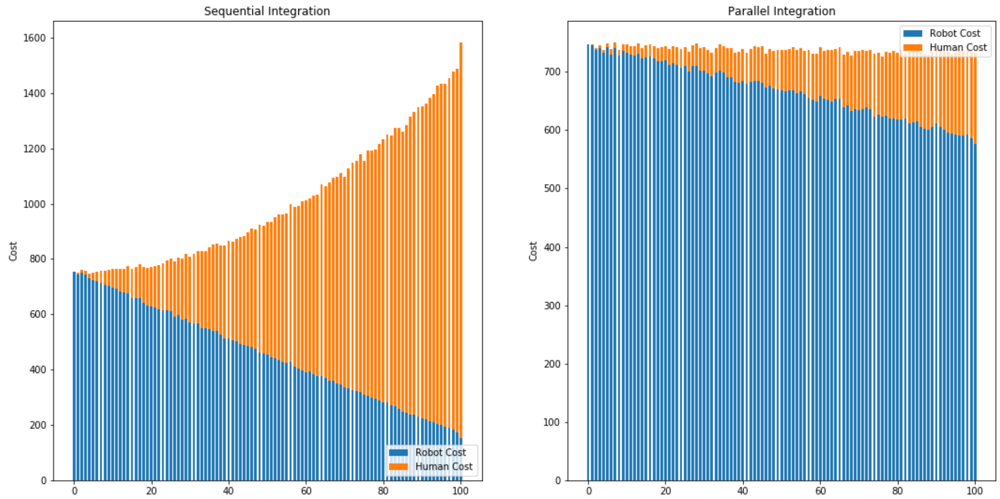


Figure 27: Cost breakdown

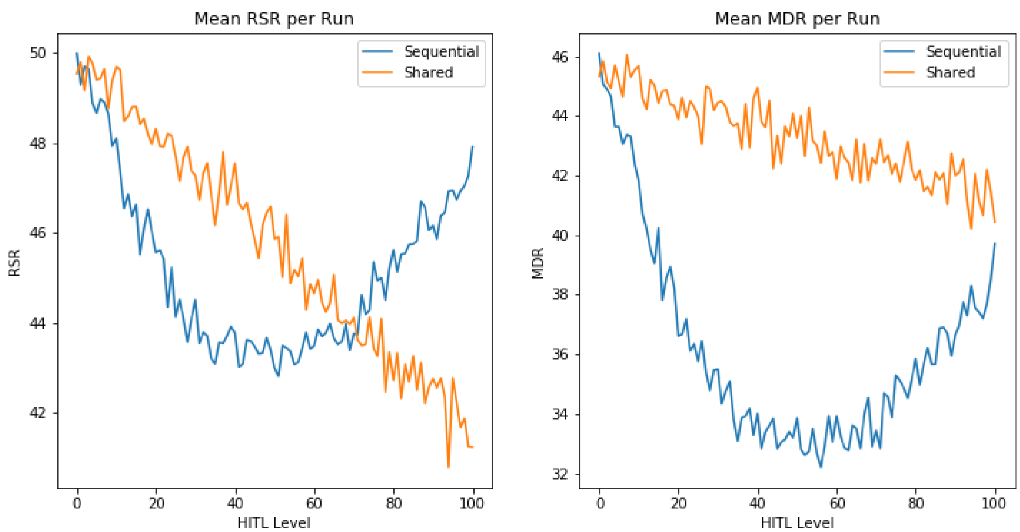


Figure 28: RSR and MDR comparison across integrations

The comparison of RSR and MDR across integrations also indicates the following: at lower HITL levels, sequential integration tends to provide improved metrics. The best case MDR for shared integration (in Figure 28, at HITL Level = 100% - 40.01%) still underperforms when compared to Sequential Integration, which had a lower best case MRD (at OHL - 32.02%). Best case RSR for shared integration (at HITL Level = 100 – 41.08%) performs marginally better than sequential integration (at OHL – 42.70%).

5.2 Phase 2: ARS HUB-CI GUI

The results of Phase 1 govern certain critical parameters in Phase 2, especially in determining the number of HITL tasks to be included. By default, the ARS HUB-CI software is based on the Sequential Integration. Based on the chosen modes of operation, an ARS can be simulated, or run with actual physical agents. Figure 29 provides a snapshot of real mode, where real-time communication is established with the robot. The console windows in both images establish the validity of the connection, and initiate sharing of the next location to be visited by the robot based on current inferences.

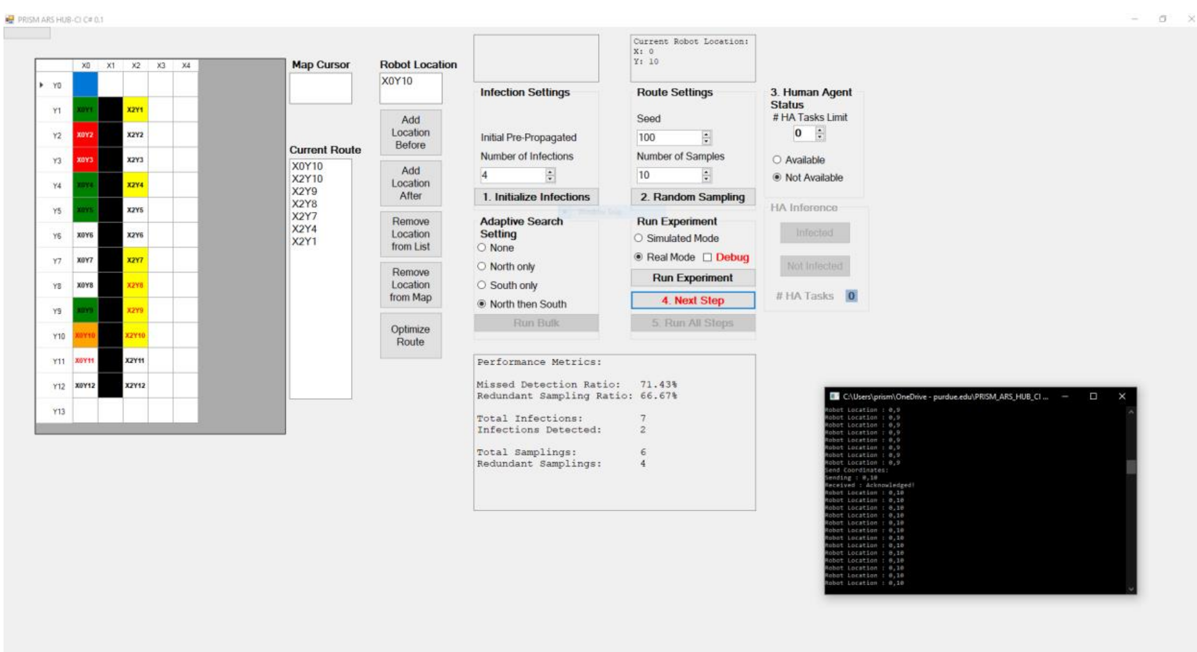


Figure 29: Snapshot of software when real mode is initiated (BARD technical report, 2020)

In Figure 29 and 30, the command prompt window is responsible for communication with the PRISM ROS Server and shows the connection status in real-time.

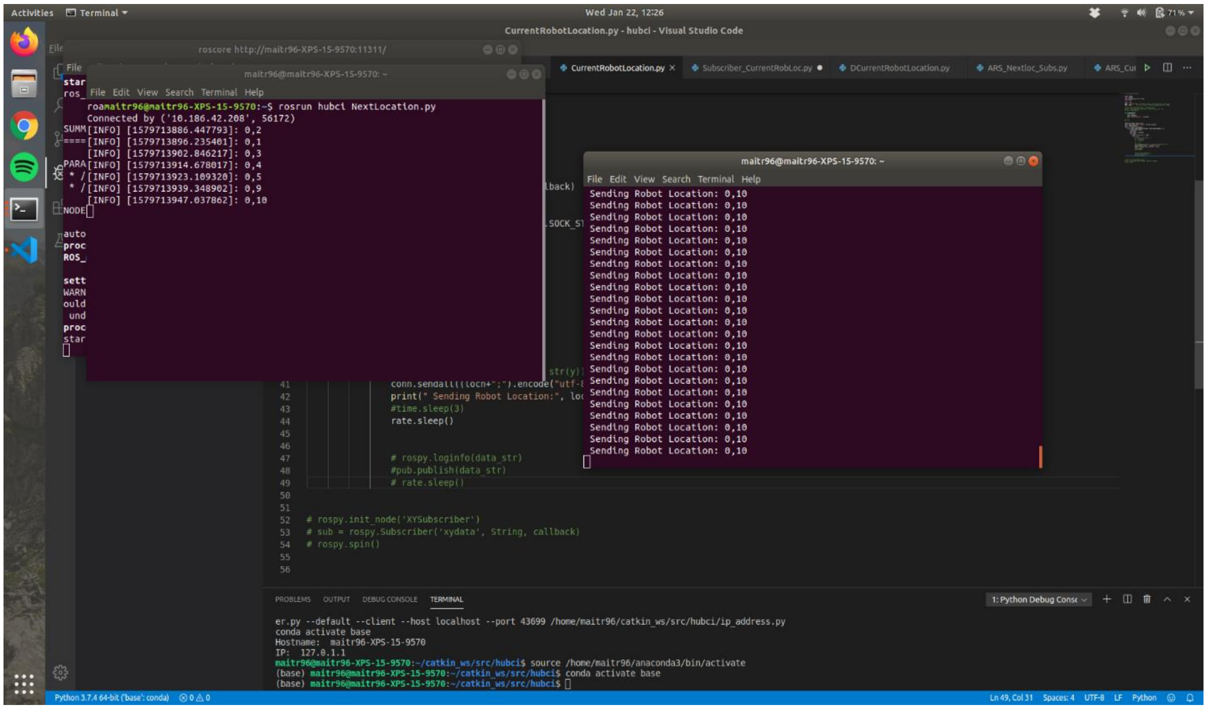


Figure 30: Snapshot of the PRISM ROS Server (BARD technical report, 2020)

Terminal windows show the connection status ((L)- between HUB-CI client and PRISM ROS Server) and the real-time location sharing ((R) – between PRISM ROS Server and ARO ROS Subscriber). These figures confirm the ability of the software to create cross-platform communication and information structures between software and robotic agents.

Based on the information received from the three central algorithms, the next location to be visited is confirmed.

The performance metrics for each run are shown in a window in real time (Figure 31). Based on the chosen sampling strategy and efficacy of HITL, the system performance is reflected in the metrics.

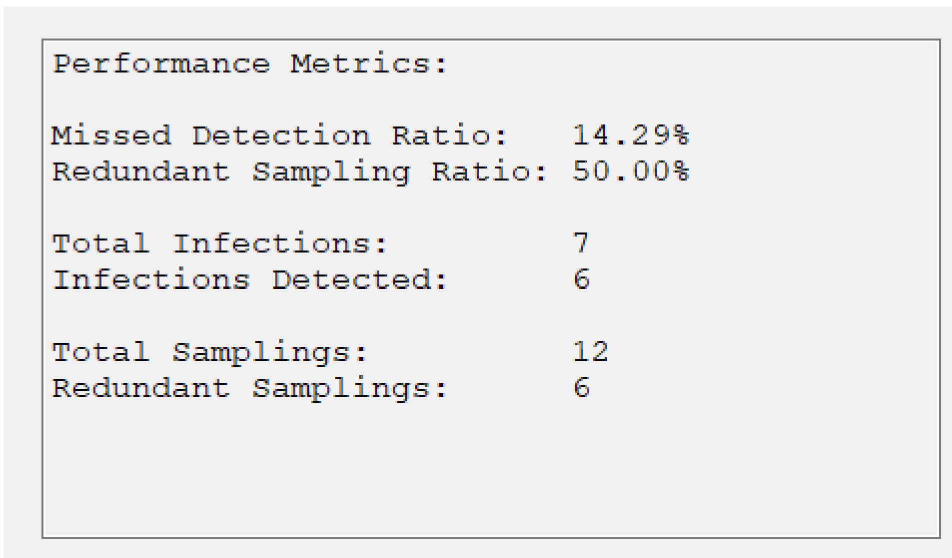


Figure 31: Real-time performance metrics (BARD project report, 2019)

6. CONCLUSIONS

This research discusses Human-in-the-loop systems within the scope of Agricultural Robotic Systems. While an automated loop is advantageous in repetitive and well-defined environments, there is a need for humans in the loop, given the unstructured nature of the environment and their ability to leverage specific knowledge bases for cognitive decision-making and versatility in handling dynamic events. The objective is twofold; 1) to create a precise HA model for accurate integration into the loop, and 2) to explore different possible integrations of humans within the cyber-physical loop, with the target of early detection of diseases in greenhouse plants. HAs can physically intervene with the system (HITL Level), providing improved task productivity and detection ability to the system. The availability of HAs are modelled through active time-constraints called HITL Level of the simulation, which signifies the time-period for which a HA is available to perform tasks. Agent-level errors and defects are considered to make the simulation more reflective of real-world scenarios, an example being the fatigue and error models considered for the development of HAs.

To manage communication and interaction between diverse agents (robots, humans) and algorithms (Sampling, Adaptive Search and HITL), a simulation based on HUB-CI, a collaboration management platform is developed. The simulation monitors KPI's of the system in relation to sampling and routing efficiency, task classification and productivity, and overall costs. By comparing the system performance for different simulation protocols, the devised experiments provide insight into the augmentation provided by the three primary algorithms, Sampling/Routing, Adaptive Search and HITL for two HITL hierarchies: Sequential, and Shared HITL Integrations, while also eliciting the positives and negatives of each integration. Both Integrations establish the relative superiority of the HUB-CI simulation protocol in providing improved performance metrics. By providing evidence supporting the use of intelligent algorithms, the simulation can justify the investment in implementing these algorithms in place of traditional agricultural methods. Simulation different collaboration protocols also emphasizes the extent to which existing protocols can be improved, thus also entailing research into more intelligent methods.

In Sequential Integration, the priority of task management is agent utilization, and task control is maintained individually based on availability. If HAs are available, they are necessarily provided task control and are responsible for task performance. Based on the established workflow, the performance metrics calculated inferred the existence of an Optimal HITL Level (OHL), where the productivity gains from HAs are at a maximum, which has been shown in the experiment results. At this OHL, metrics such as tasks performed, task classifications are at a relative maximum, while Redundant Sampling Ratio, Missed Detection Ratio are at a minimum. However, the main drawback of this integration is the cost management, since by compulsorily pushing tasks to HAs, increased costs due to fatigue and error penalties lead to an exponential increase in system costs. As shown in the cost comparison, the cost contribution due to HAs overshadows the cost from the RAs for higher HITL Levels, thus indicating the cascading effects of fatigue and error. Also, statistical tests indicate the efficacy of sampling and AS as effective algorithms when compared to traditional methods, with the HUB-CI protocol providing higher task performance at relatively lower cost.

For Shared Integration, task control is shared between RAs and HAs, conditional on the necessity of sharing. HAs are inserted into the workflow (subject to availability) only when collaboration is absolutely essential to provide an inference, and otherwise they are not required to perform tasks. Since task control is now shared, the number of tasks performed can be expected to decrease, since both agents now contribute to contentious tasks (as opposed to one in Sequential Integration). The experiment results also do not provide evidence of an Optimal HITL Level (OHL) for this integration, since HAs are not driven to complete utilization (and thus maximum productivity benefits). However, the positives of this integration are visible in the cost performance, which remains largely stable over the different HITL Levels; on comparing to Sequential Integration we can infer that this is due to the lack of productivity and penalty losses occurring from fatigue and induced errors. The sampling algorithm also proves to be statistically effective when it comes to tasks and costs, while AS does not provide similar cost reductions (though it does provide task performance improvements).

Table 7: HUB-CI vs Traditional performance comparison for different integrations

Integration	**Cost**		**Task Performance**		**RSR**		**MDR**	
	w/o AS	**AS**	**w/o AS**	**AS**	**w/o AS**	**AS**	**w/o AS**	**AS**
Sequential	10%	0.8%	44%	15%	-30%	-26%	-50%	-41%
Shared	-5%	-5%	44%	18%	-32%	-27%	-44%	-37%

By establishing the critical factors of both HITL integrations, the advantages of the HUB-CI protocol for ARS are elaborated (Table 7). The use of simulation enables the validation of different configurations and collaborative protocols and provides impetus to test them out in the real world. In order to test out the HUB-CI workflow in real-world scenarios, the HUB-CI ARS software is developed. It enables a unified platform which connects different components of the ARS, enables real-time communication and decision-making from both robots and humans, and enables dynamic response to external conditions of the system. By maintaining a collaborative platform, the software encapsulates cyber-supported collaboration within a diverse, multi-agent system. The working of the software enables human experts to direct the workflow in certain cases, providing an easy and intuitive user interface to provide inputs, and see the results in real time. Such software tools can be deployed for effective agricultural management in greenhouses, not just limited to disease detection but also for extended uses such as irrigation and spraying of pesticides/fungicides. By providing measurable and quantifiable benefits over traditional methods, the design of HUB-CI ARS can be expanded to cover a wider range of agents, algorithms and protocols.

6.1 Addressing the research questions

The following table describes how this research address the formulated research questions

Table 8: Research Questions

Objective	Contribution
1. What simulation models and technologies can we use to improve the productivity of current Agricultural Robotic Systems?	Describing the requirement of HITL for ARS, Studying different HITL integrations via simulation (Sections 3.1-3.4) Development of the HUB-CI ARS software in Section 3.5 as a central controller for ARS
2. Cyber-Supported Collaboration: - How do we use Cyber-Supported Collaboration for Agricultural Robotic Systems? - Which collaboration protocols can we employ between agents to provide the optimal performance parameters?	-Use of HUB-CI based protocol managing sampling, routing, Adaptive Search and HITL protocols. -HUB-CI ARS software connecting to robots via ROS and Humans in real time for decision support, navigation
3. Human-Machine Interactions: - How do we configure the system for Human-Machine Interactions? What considerations do we make while modelling system agents? - How do different Human-in-the-loop integrations affect the system? What are the different advantages and disadvantages to these integrations?	-Robust Human Model including fatigue, availability and task performance for simulation -Different HITL Integrations to study the positives and negatives for different workflows -Varying HITL Level across integrations to for optimal

6.2 Limitations and future research

1. Human models can improve from well researched fatigue and task performance models. The expanded effect of fatigue beyond task performance and error scope and including physical fatigue should be studied to greater extent. Not a lot of extensive research has been done into how fatigue works in terms of tasks such as scanning, data interpretation – and its exact relation to task error. Improved models can certainly dictate the configuration of optimal human robot collaboration
2. The sampling and routing heuristics are efficient for smaller grids with limited number of plants – for larger grids choosing more computationally efficient heuristics will prove more optimal. Also, research in techniques such as Bayes Networks, Grid Sampling can provide useful insight for agricultural sampling
3. HUB-CI ARS has not been rigorously tested with online scanning system – and the communication methods are developed for inter-university communication and hence not optimal for commercial deployment. ROS messages limit the amount of information being shared, and other passages must be explored for communication
4. The scope of human tasks can be expanded beyond scanning and adding/removing nodes. While these tasks might be the extent for the current ARS, with increasingly complex systems, human cognition and perception can be leveraged to improve critical decision making and workflow management.
5. The parameters chosen for the HITL integrations studies such a task time, probabilities of error (for robot and human) can be improved by more empirical data on ARS task studies.

Future research can be targeted towards hybrid interactions consisting of elements from both types of integrations. For e.g., a hybrid workflow where a priority-based time-out protocol can be used to determine the amount of time the HA has to dynamically interface in the system, based on task characteristics. It is also important to improve HA models to make them more realistic towards decision making, sampling and also expanding the range of performable tasks in the loop. It will also be interesting to look into learning model along with fatigue models for the human agents, since repetitive tasks also improve the retentivity and skill acquisition of humans.

Rigorous testing must also take place under real world conditions to validate the scope and parameters assumed in this research. HITL systems can establish the future of human machine systems and advance the definition of human agents in tune with futuristic developments. The HUB-CI ARS software can also be expanded to include real-time camera visualization, connection to sophisticated scanning systems, sensors networks to enable all-round control and collaborative management of the system.

APPENDIX A

Neural Networks in Prediction of Agricultural Patterns (Source: Developed by PRISM ARS Research Team)

Early prediction of crop yields in plants such as rice, corn, and wheat has been an interesting area of research for agricultural meteorologists since they provide significant economic value. Development of systems that can predict crop patterns based on environmental data has ventured into the application of Artificial Intelligence (AI), Artificial Neural Networks (ANNs), and Fuzzy Systems (Anitha & Chakravarthy, 2018). ANNs are mathematical models that have been utilized in data mining. Fundamentally, ANNs are an interconnected network of nodes, parallel to the vast network of neurons in the human brain. The connection between two neurons carries the weights in which the information is implicitly coded. Neural networks have recently gained popularity as an alternative to regression models to characterize pattern-based models. In precision agriculture, ANNs have been used traditionally to characterize and recognize different diseases from hyperspectral or multispectral images of healthy plant leaves (Behmann, Steinrücken, & Plümer, 2014). However, in order to have accurate preventive maintenance in place, we need a system capable of providing us with the extent of the spread of disease and, based on this propagation (and environmental conditions), an accurate prediction of the type of disease. This research attempts to employ ANNs to predict patterns in plant infection propagation, estimating the type of infection based on a probability distribution of disease spread.

This research (performed by the PRISM ARS research team) assumes that the direction of disease propagation is known and calculated beforehand. Based on this knowledge the Adaptive Search Setting is chosen to maximize coverage of the infected plans. To provide more insight into this assumption, this section is dedicated to the experimental prediction of the infection direction using Artificial Neural Networks (ANNs). The research questions targeted in this section are as follows:

1. How do we design an Artificial Neural Network to predict disease direction in greenhouse crops?
2. What are the optimal parameters for the model in order to minimize costs?

In this section, the three experiments are described. The three experiments were designed to investigate the impact of different parameters in the ANNs, which will lead to the difference in total cost. The main objective is to minimizing Total (normalizing) cost (**TC_N**), which is composed of two main costs; computational cost (**C_c**) and error cost (**C_e**). The following paragraph will discuss TCN and its components.

Performance Metrics

- **Total (normalizing) cost (TC_N)**

TC_N composed of Computational cost (C_c) and Error cost (C_e). However, as C_c and C_e are measured in a different scale, cost normalization is necessary. The following formula, equation (1), is used for computing **TC_N**.

$$TC_N = \left[\frac{C_c}{\max(C_C)}\right] + \left[\frac{C_e}{\max(C_e)}\right] \tag{1}$$

- **Computational cost (C_c)**

C_c represents the cost of computing the result which can be considered as computational time. Ideally, the longer the time of the training, the better the result of the prediction. The computational time can be represented in (2).

$$C_c = \alpha T \tag{2}$$

Where α represents the cost per unit time of computing, and T represents the time of computing.

- **Error cost (C_e)**

C_e is the cost for error that happens in the prediction. A larger number of L and N will lead to a lower error in prediction. The error cost can be calculated as shown in (3).

$$C_e = \beta E \tag{3}$$

Where β represents the cost per error and E represents the number of prediction errors.

Experiment design

To minimize the effect of initial weights, each setting will run ten times. The three experiments are:

Experiment 1: Fixed number of nodes per layer and varied number of layers.

Experiment 1 studied the impact of the number of (hidden) layers on the performance and time of the network. In the setting, ten runs per setting were conducted for the networks with the number of hidden layers ranging between 2 to 20. Each layer is set to have ten nodes for the ease of comparison. Computation time and performance function (MSE) were taken to compare the performance improvement by adding hidden layers in the network.

Experiment 2: Fixed number of layers and varied number of nodes per layer

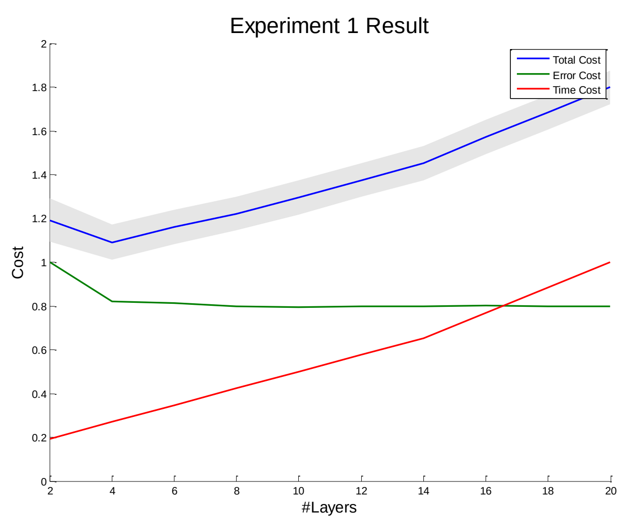
The second experiment has an objective to investigate the impact of the number of nodes on the ANNs performance and computational time. In the baseline setting, ten runs per setting were conducted for a network with three hidden layers. Each layer is set to have 5 to 50 nodes. Computation time and performance function were taken to compare the performance improvement by adding nodes in the fixed depth network.

Results

Experiment 1: Layers Effect of Prediction Performance

Experiment 1 was performed to observe the effect of the number of layers of the neural network on the time and performance indicators. The results of Experiment 1 are plotted in Figure 32, which measures the time, error, and total cost for the experiment with increasing layers of the ANNs, given that each layer contains the same number of nodes (N = 10). From Figure 32, it is evident that the total cost achieves a global minimum when L = 4 (3 hidden layers), beyond which the total cost rises monotonically. The reason is that the error cost achieves steady state after three hidden layers, while the time cost increases as the number of layers increase. Thus, three layers are chosen as the optimal number of hidden layers for this experiment. It should be noted that the same activation (LOGSIG) and training function (Gradient Descent) were used for each case.

Figure 32 also provides the confusion matrix of the best-case scenario for Experiment 1, with an output accuracy of 80.6%. One observation is the pattern target 3 has the lowest hit rate, which came from the similarity between patterns 3 and 5.



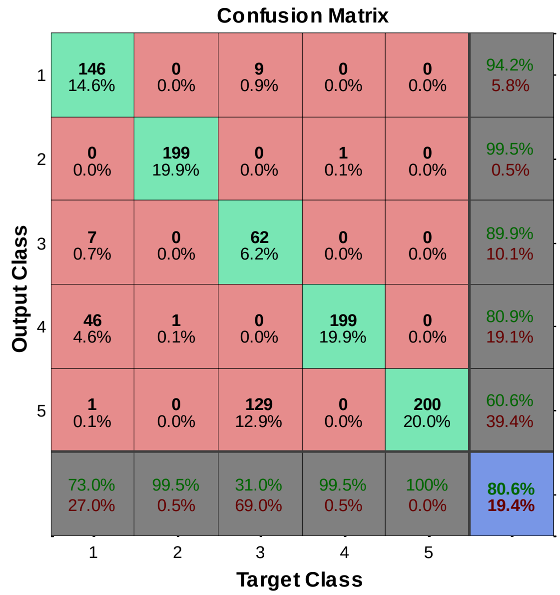


Figure 32: Cost Comparison and Confusion Matrix for Experiment 1 (Developed by PRISM ARS research team)

Experiment 2: Nodes Effect of Prediction Performance

In Experiment 2, the effect of the number of nodes on the cost and performance indicators is observed. The number of nodes per layer is varied while maintaining the same number of hidden layers for each case (hidden layers = 3), the results of which can be seen in Figure 33. From the plots, it can be inferred that for N = 10 the total cost is minimized, and on a further increase of N, the total cost increases. The reason is observed as the error cost saturates for N = 10, while the time cost increases as N increases. Thus, the output of this experiment dictates that the optimal number of nodes per layer should be limited to 10, and at these parameters, the total cost of computation and errors is minimum. As with the previous experiment, the same activation and training functions are used for each case.

Figure 33 also displays the confusion matrix for the best-case scenario, which has an output accuracy of 85.6%. The results indicate that 85.6% of the data patterns are recognized correctly. The hit rate for pattern 1 is the lowest because it has an overlapping pattern to others. (Pattern 1 is the multidirectional propagation, while other patterns such as patterns 4 and 5, which are the unidirectional propagation direction, have perfect pattern recognition.)

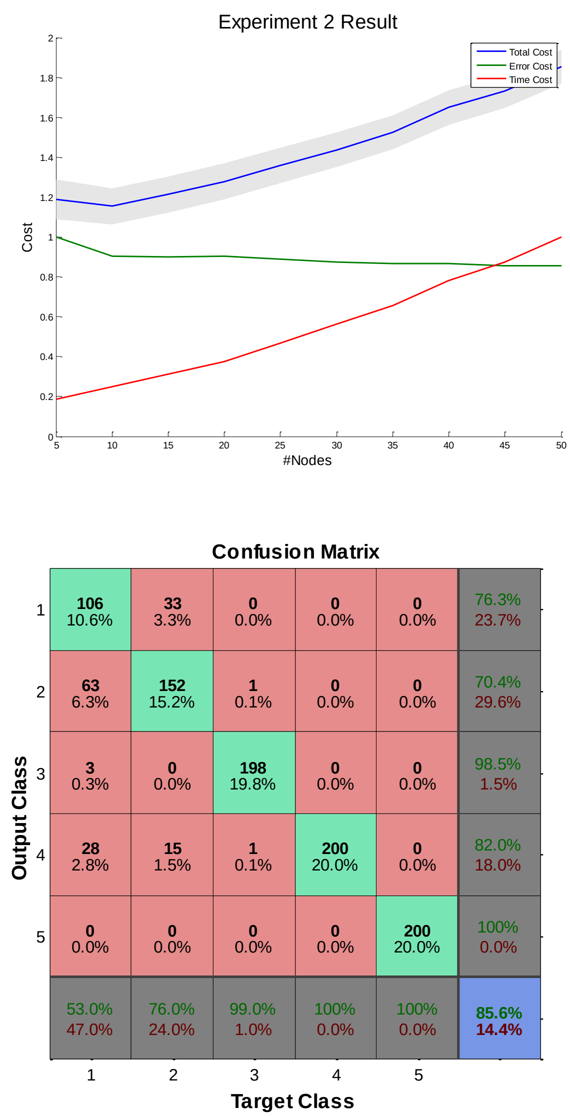


Figure 33: Cost Comparison and Confusion Matrix for Experiment 2 (Developed by PRISM ARS research team)

REFERENCES

Anitha, P., & Chakravarthy, T. (2018). Agricultural Crop Yield Prediction using Artificial Neural Network with Feed Forward Algorithm. *International Journal of Computer Sciences and Engineering*, *6*(11), 178–181. https://doi.org/10.26438/ijcse/v6i11.178181

Aqeel-Ur-Rehman, Abbasi, A. Z., Islam, N., & Shaikh, Z. A. (2014). A review of wireless sensors and networks' applications in agriculture. *Computer Standards and Interfaces*, *36*(2), 263–270. https://doi.org/10.1016/j.csi.2011.03.004

Bechar, A., & Edan, Y. (2003). Human-robot collaboration for improved target recognition of agricultural robots. *Industrial Robot*, *30*(5), 432–436. https://doi.org/10.1108/01439910310492194

Bechar, A., Edan, Y., & Meyer, J. (2006). Optimal collaboration in human-robot target recognition systems. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, *5*, 4243–4248. https://doi.org/10.1109/ICSMC.2006.384801

Bechar, A., Meyer, J., & Edan, Y. (2007). An objective function to evaluate performance of human-robot systems for target recognition tasks. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, *84105*, 967–972. https://doi.org/10.1109/ICSMC.2007.4414015

Bechar, A., & Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, *149*, 94–111. https://doi.org/10.1016/j.biosystemseng.2016.06.014

Behmann, J., Steinrücken, J., & Plümer, L. (2014). Detection of early plant stress responses in hyperspectral images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *93*, 98–111. https://doi.org/10.1016/j.isprsjprs.2014.03.016

Belforte, G., Deboli, R., Gay, P., Piccarolo, P., & Ricauda Aimonino, D. (2006). Robot Design and Testing for Greenhouse Applications. *Biosystems Engineering*, *95*(3), 309–321. https://doi.org/10.1016/j.biosystemseng.2006.07.004

Bhargava, R., Levalle, R. R., & Nof, S. Y. (2016). A best-matching protocol for order fulfillment in re-configurable supply networks. *Computers in Industry*, *82*, 160–169. https://doi.org/10.1016/j.compind.2016.07.001

Blanchet, M., Rinn, T., Von Thaden, G., & De Thieulloy, G. (2014). Industry 4.0: The new industrial revolution-How Europe will succeed. *Hg. v. Roland Berger Strategy Consultants GmbH. München. Abgerufen Am 11.05. 2014, Unter Http://Www. Rolandberger. Com/Media/Pdf/Roland_Berger_TAB_Industry_4_0_2014 0403. Pdf*.

Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., & Belongie, S. (2010). Visual Recognition with Humans in the Loop. In *Proceedings of the 11th European Conference on Computer Vision: Part IV* (pp. 438–451). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1888089.1888123

Bringes, C., Lin, Y., Sun, Y., & Alqasemi, R. (2013). Determining the benefit of human input in human-in-the-loop robotic systems. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 210–215. https://doi.org/10.1109/ROMAN.2013.6628447

Chipalkatty, R. (2012). Human-in-the-loop control for cooperative human-robot tasks. *PhD Dissertations*, *3533131*(May), 144. Retrieved from http://ezproxy.net.ucf.edu/login?url=http://search.proquest.com/docview/1221580706?accountid=10003%5Cnhttp://sfx.fcla.edu/ucf?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&genre=dissertations+&+theses&sid=ProQ:ProQuest+Dissertations+&+

Devadasan, P., Zhong, H., & Nof, S. Y. (2013). Collaborative intelligence in knowledge based service planning. *Expert Systems with Applications*, *40*(17), 6778–6787. https://doi.org/10.1016/j.eswa.2013.06.048

Dik, D., Polyakova, E., Chelovechkova, A., Moskvin, V., & Nikiforova, T. (2018). The System of Environment Control of Botanic Garden Greenhouses. *2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018*, 1–7. https://doi.org/10.1109/FarEastCon.2018.8602559

Driggs-Campbell, K., Bellegarda, G., Shia, V., Sastry, S. S., & Bajcsy, R. (2014). Experimental Design for Human-in-the-Loop Driving Simulations, 1–10. Retrieved from http://arxiv.org/abs/1401.5039

Dumitrache, I., Sacala, I. S., Moisescu, M. A., & Caramihai, S. I. (2017). A conceptual framework for modeling and design of Cyber-Physical Systems. *Studies in Informatics and Control*, *26*(3), 325–334. https://doi.org/10.24846/v26i3y201708

Durugbo, C. (2016). Collaborative networks: A systematic review and multi-level framework. *International Journal of Production Research*, *54*(12), 3749–3776. https://doi.org/10.1080/00207543.2015.1122249

Dusadeerungsikul, P. O., & Nof, S. Y. (2019). A collaborative control protocol for agricultural robot routing with online adaptation. *Computers and Industrial Engineering*, *135*(August 2018), 456–466. https://doi.org/10.1016/j.cie.2019.06.037

Emmanouilidis, C., Pistofidis, P., Bertoncelj, L., Katsouros, V., Fournaris, A., Koulamas, C., & Ruiz-Carcel, C. (2019a). Enabling the human in the loop: Linked data and knowledge in industrial cyber-physical systems-physical systems Internet of things Context information management Product lifecycle management Asset lifecycle management Maintenance Human in the loop. *Annual Reviews in Control*, *47*, 249–265. https://doi.org/10.1016/j.arcontrol.2019.03.004

Emmanouilidis, C., Pistofidis, P., Bertoncelj, L., Katsouros, V., Fournaris, A., Koulamas, C., & Ruiz-Carcel, C. (2019b). Enabling the human in the loop: Linked data and knowledge in industrial cyber-physical systems. *Annual Reviews in Control*, *47*, 249–265. https://doi.org/10.1016/j.arcontrol.2019.03.004

Fereres, A., & Raccah, B. (2015). Plant Virus Transmission by Insects. In *eLS* (pp. 1–12). American Cancer Society. https://doi.org/10.1002/9780470015902.a0000760.pub3

Genaldy, A. M., Duggal, J. S., & Mital, A. (1990). A comparison of robot and human performances for simple assembly tasks. *International Journal of Industrial Ergonomics*, *5*(1), 73–81. https://doi.org/https://doi.org/10.1016/0169-8141(90)90029-2

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, *116*, 8–19. https://doi.org/10.1016/j.compag.2015.05.021

Guo, P., Dusadeeringsikul, P., & Nof, S. Y. (2018). Agricultural cyber physical system collaboration for greenhouse stress management. *Computers and Electronics in Agriculture*, *150*(June 2017), 439–454. https://doi.org/10.1016/j.compag.2018.05.022

Holzinger, A., Plass, M., Katharina, M. K., Gloria, H., & Cris, C. (2019). Interactive machine learning : experimental evidence for the human in the algorithmic loop A case study on Ant Colony Optimization, 2401–2414.

Jaber, M. Y., Givi, Z. S., & Neumann, W. P. (2013). Incorporating human fatigue and recovery into the learning-forgetting process. *Applied Mathematical Modelling*, *37*(12–13), 7287–7299. https://doi.org/10.1016/j.apm.2013.02.028

Jaber, M. Y., & Neumann, W. P. (2010). Modelling worker fatigue and recovery in dual-resource constrained systems. *Computers and Industrial Engineering*, *59*(1), 75–84. https://doi.org/10.1016/j.cie.2010.03.001

Jouppi, M., Schweigert, S., Sun, J., Su, J., & Srinivasan, S. (2015). Robotic Manipulation with a Human in the Loop. Retrieved from http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-78.html

Kaneko, K., & Sakamoto, K. (2001). Spontaneous Blinks as a Criterion of Visual Fatigue during Prolonged Work on Visual Display Terminals. *Perceptual and Motor Skills*, *92*(1), 234–250. https://doi.org/10.2466/pms.2001.92.1.234

Kawamura, N., & Namikawa, K. (1988). Robots in agriculture. *Advanced Robotics*, *3*(4), 311–320. https://doi.org/10.1163/156855389X00253

Khaitan, S. K., & McCalley, J. D. (2015). Design techniques and applications of cyberphysical systems: A survey. *IEEE Systems Journal*, *9*(2), 350–365. https://doi.org/10.1109/JSYST.2014.2322503

Koutsos, T., & Menexes, G. (2019). Economic, Agronomic, and Environmental Benefits From the Adoption of Precision Agriculture Technologies. *International Journal of Agricultural and Environmental Information Systems*, *10*(1), 40–56. https://doi.org/10.4018/IJAEIS.2019010103

Lee, E. A. (2010). CPS foundations. *Proceedings - Design Automation Conference*, 737–742. https://doi.org/10.1145/1837274.1837462

Lentz, E. C., Project, C., Gardner, B. L., Rausser, G. C., Ruel, M. T., Assessment, F. S., … Nutrition, M. (2010). References and Notes 1., (February).

Libin, Z., Qinghua, Y., Guanjun, B., Yan, W., Liyong, Q., Feng, G., & Fang, X. (2008). Overview of research on agricultural robots in China. *International Journal of Agricultural and Biological Engineering*, *1*(1), 12–21. https://doi.org/10.3965/j.issn.1934-6344.2008.01.012-021

Ma, L., Chablat, D., Bennis, F., & Zhang, W. (2009). A new simple dynamic muscle fatigue model and its validation. *International Journal of Industrial Ergonomics*, *39*(1), 211–220. https://doi.org/https://doi.org/10.1016/j.ergon.2008.04.004

Mclennan, M., & Kennell, R. (2010). HUBzero: A platform for dissemination and collaboration in computational science and engineering. *Computing in Science and Engineering*, *12*(2), 48–52. https://doi.org/10.1109/MCSE.2010.41

Moghaddam, M., & Nof, S. Y. (2017). The collaborative factory of the future. *International Journal of Computer Integrated Manufacturing*, *30*(1), 23–43. https://doi.org/10.1080/0951192X.2015.1066034

Monta, M., & Namba, K. (2003). Three-dimensional sensing system for agricultural robots. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, *2*(Aim), 1216–1221. https://doi.org/10.1109/AIM.2003.1225516

Munir, S., Stankovic, J. a, Liang, C.-J. M., & Lin, S. (2013). Cyber Physical System Challenges for Human-in-the-Loop Control. *The 8th International Workshop on Feedback Computing*. https://doi.org/10.3969/j.issn.1674-4748.2015.31.001

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. https://doi.org/https://doi.org/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. https://doi.org/10.1016/j.arcontrol.2007.08.002

PAUL, R. P., & NOF, S. Y. (1979). Work methods measurement—a comparison between robot and human task performance. *International Journal of Production Research*, *17*(3), 277–303. https://doi.org/10.1080/00207547908919615

Pedersen, S. M., Fountas, S., Have, H., & Blackmore, B. S. (2006). Agricultural robots - System analysis and economic feasibility. *Precision Agriculture*, *7*(4), 295–308. https://doi.org/10.1007/s11119-006-9014-9

Rad, C.-R., Hancu, O., Takacs, I.-A., & Olteanu, G. (2015). Smart Monitoring of Potato Crop: A Cyber-Physical System Architecture Model in the Field of Precision Agriculture. *Agriculture and Agricultural Science Procedia*, *6*, 73–79. https://doi.org/10.1016/j.aaspro.2015.08.041

Ramin Shamshiri, R., A. Hameed, I., Pitonakova, L., Weltzien, C., K. Balasundram, S., J. Yule, I., … Chowdhary, G. (2018). Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison. *International Journal of Agricultural and Biological Engineering*, *11*(4), 12–20. https://doi.org/10.25165/j.ijabe.20181104.4032

Reid, J. F., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture*, *25*(1–2), 155–167. https://doi.org/10.1016/S0168-1699(99)00061-7

Romanov, V., Galelyuka, I., & Sarakhan, Y. (2016). Wireless sensor networks in agriculture. *2015 IEEE 7th International Conference on Intelligent Computing and Information Systems, ICICIS 2015*, 77–80. https://doi.org/10.1109/IntelCIS.2015.7397200

Sankaran, S., Mishra, A., Ehsani, R., & Davis, C. (2010). A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture*, *72*(1), 1–13. https://doi.org/10.1016/j.compag.2010.02.007

Sarangi, S., Umadikar, J., & Kar, S. (2016). Automation of Agriculture Support Systems using Wisekar: Case study of a crop-disease advisory service. *Computers and Electronics in Agriculture*, *122*, 200–210. https://doi.org/https://doi.org/10.1016/j.compag.2016.01.009

Schirner, G., Erdogmus, D., Chowdhury, K., & Padir, T. (2013). The future of human-in-the-loop cyber-physical systems. *Computer*, *46*(1), 36–45. https://doi.org/10.1109/MC.2013.31

Soo, Y., Nishino, M., Sugi, M., Yokoi, H., Arai, T., Kato, R., … Ota, J. (2009). Evaluation of Frequency Band Technique in Estimating Muscle Fatigue during Dynamic Contraction Task. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation* (pp. 4206–4211). IEEE Press.

Staranowicz, A., & Mariottini, G. L. (2011). A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2141622.2141689

Stewart, A., Cao, M., Nedic, A., Tomlin, D., & Leonard, N. (2012). Towards Human–Robot Teams: Model-Based Analysis of Human Decision Making in Two-Alternative Choice Tasks With Social Feedback. *Proceedings of the IEEE*, *100*(3), 751–775. https://doi.org/10.1109/JPROC.2011.2173815

Tsolakis, N., Bechtsis, D., & Bochtis, D. (2019). Agros: A robot operating system based emulation tool for agricultural robotics. *Agronomy*, *9*(7). https://doi.org/10.3390/agronomy9070403

Velásquez, J. D., & Nof, S. Y. (2009). Collaborative e-Work, e-Business, and e-Service. In *Springer handbook of automation* (pp. 1549–1576). Springer.

Winkelhaus, S., Sgarbossa, F., Calzavara, M., & Grosse, E. H. (2018a). The effects of human fatigue on learning in order picking: an explorative experimental investigation. *IFAC-PapersOnLine*, *51*(11), 832–837. https://doi.org/10.1016/j.ifacol.2018.08.442

Winkelhaus, S., Sgarbossa, F., Calzavara, M., & Grosse, E. H. (2018b). The effects of human fatigue on learning in order picking: an explorative experimental investigation, 832–837. https://doi.org/10.1016/j.ifacol.2018.08.442

Zhang, L., Zhong, H., & Nof, S. Y. (2015). Adaptive Fuzzy Collaborative Task Assignment for Heterogeneous Multirobot Systems. *International Journal of Intelligent Systems*, *30*(6), 731–762. https://doi.org/10.1002/int.21725

Zhong, H., Levalle, R. R., Moghaddam, M., & Nof, S. Y. (2015). Collaborative intelligence - Definition and measured impacts on Internetworked e-Work. *Management and Production Engineering Review*, *6*(1), 67–78. https://doi.org/10.1515/mper-2015-0009

Zhong, H., Wachs, J. P., & Nof, S. Y. (2013). *HUB-CI model for collaborative telerobotics in manufacturing*. *IFAC Proceedings Volumes (IFAC-PapersOnline)* (Vol. 46). IFAC. https://doi.org/10.3182/20130522-3-BR-4036.00059

Zhong, H., Wachs, J. P., & Nof, S. Y. (2014). Telerobot-enabled HUB-CI model for collaborative lifecycle management of design and prototyping. *Computers in Industry*, *65*(4), 550–562. https://doi.org/10.1016/j.compind.2013.12.011