Probabilistic modelling and safety assurance of an agriculture robot providing light-treatment

Mustafa Adam[1], Kangfeng Ye[2], David A. Anisi[1,3], Ana Cavalcanti[2], Jim Woodcock[2], and Robert Morris[4]

***Abstract*— Continued adoption of agricultural robots postulates the farmer's trust in the reliability, robustness and safety of the new technology. This motivates our work on safety assurance of agricultural robots, particularly their ability to detect, track and avoid obstacles and humans. This paper considers a probabilistic modelling and risk analysis framework for use in the early development phases. Starting off with hazard identification and a risk assessment matrix, the behaviour of the mobile robot platform, sensor and perception system, and any humans present are captured using three state machines. An auto-generated probabilistic model is then solved and analysed using the probabilistic model checker PRISM. The result provides unique insight into fundamental development and engineering aspects by quantifying the effect of the risk mitigation actions and risk reduction associated with distinct design concepts. These include implications of adopting a higher performance and more expensive Object Detection System or opting for a more elaborate warning system to increase human awareness. Although this paper mainly focuses on the initial concept-development phase, the proposed safety-assurance framework can also be used during implementation, and subsequent deployment and operation phases.**

I. Introduction

The use of robots in agricultural tasks can: (a) improve efficiency and productivity, (b) counter the shortage of seasonal workers, and (c) cater for laborious and possibly dangerous tasks to protect humans from hazardous situations, such as spraying, mowing, pruning and light treatment as depicted in Fig. 1. Technological advances in sensing, actuation, and machine learning have allowed more agricultural tasks to be carried out by Robotic Autonomous Systems (RAS).

One area of agriculture where RAS are especially useful is plant treatment. Many plant-pathogenic microorganisms, including fungi, damage the yield [1]. Strawberries, particularly, are a high-value crop whose plants are subject to rapid degradation in taste and yield once affected. With a growing interest in reducing, or even forbidding, use of chemicals and fungicides, farmers are ready to consider alternatives. Research and real-life usage have shown that ultraviolet (UV) radiation efficiently reduces disease development in many species, including strawberries [2]. As UVC light is harmful to the human eye and is highly repetitive, and the light treatment is performed during the night and does not require physical interaction with the plants, it is ideal for automation. UVC light-treatment services to control powdery mildew using RAS are now commercially available.

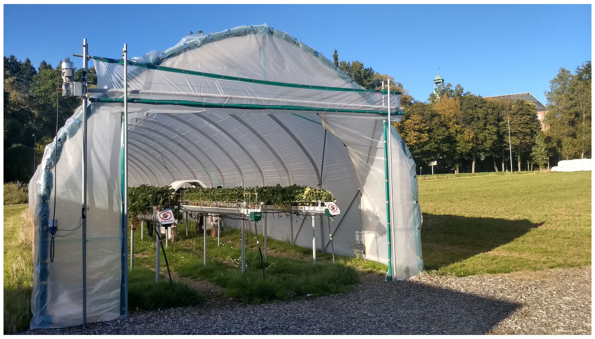


Fig. 1: Robotic UVC light-treatment in poly-tunnel at NMBU

From historical data, it is evident that the agricultural domain is more willing and capable of adopting new solutions and technology as compared to other segments. This, however, postulates the farmer's trust in the new technology's safety [3]. Therefore, assurance of agricultural robots is paramount to the continued adoption of novel robotic solutions. In this setting, the robot's ability to detect, track and avoid obstacles and humans is particularly interesting.

Safety is defined as the absence of unacceptable risk, and relevant standards require systematic identification, measurement and monitoring of these risks [4], [5]. To this end, any sound safety-engineering practice starts with the identification of all foreseeable *hazards* that can cause harm to people, the environment or business. The associated *risks* are then calculated by multiplying the severity of the hazards with their probability of occurrence. A *risk-assessment matrix*, where the probability and severity of the hazards are depicted as rows and columns, is a fundamental tool in all hazard-based safety-engineering practices.

A highly relevant safety standard for RAS in the agricultural domain is ISO 18497 [4], while IEC 61508 is a cross-industry functional-safety standard [5]. Other standards, such as ISO 31000 and ISO 21000, indicate how to perform risk assessment and mitigation actions. Identifying and managing hazards and functional failures are some of the main challenges, particularly when involving AI or learning-based components [6], [5] Several approaches are often used, such as FTA (fault-tree analysis), FMEA (failure mode and effect analysis) and its variants, model-based reasoning, qualitative reasoning, and assumption-based truth maintenance [7].

In extension to the prior art, this paper presents an ap-

[1] Robotics Group, Faculty of Science & Technology, Norwegian University of Life Sciences (NMBU), Norway {mustafa.adam, david.anisi}@nmbu.no

[2] Dept. of Computer Science, University of York, UK {kangfeng.ye, ana.cavalcanti, jim.woodcock}@york.ac.uk

[3] Dept. of Mechatronics, Faculty of Engineering and Science, University of Agder (UiA), Norway

[4] Saga Robotics AS, Norway rmorris@sagarobotics.com

proach for analysing the risks associated with UVC treatment during row transition. Starting with the hazard identification results in [8], we use a diagrammatic domain-specific notation, namely, RoboChart [9], to model the behaviour of the mobile robot platform, sensor, and perception system, as well as humans. Using RoboChart, we define three synchronised probabilistic state machines. From a RoboChart model, a probabilistic model can be auto-generated, then solved and analysed using the PRISM model checker [10]. Here, based on this, risk mitigation plans and several design concepts are proposed to control risk. In particular, results regarding the effect of ODS on reducing the risk below tolerable level are presented. In [9], [10], RoboChart is used to model existing RAS, including the robot platform, controller and mechanical components. This paper complements that work by utilising the same modelling and reasoning framework to capture and quantify risks during the early development phase.

The remainder of this paper is organised as follows. Next, we discuss related work. Section III describes the light-treatment use case. Section IV describes the adopted probabilistic model-checking methodology. Section V explains the RoboChart models and the safety-properties formalisation. The analysis results are in Section VI. Finally, conclusions and directions for future work are in Section VII.

II. Related work

Mayoral *et al.* [11], [12] also considers the safety of humans near agricultural robots. The sensor is an RGBD camera, and a neural network architecture (YOLOv4) is used to classify humans concerning their distance from the robot. This complementary work sets the stage for the Object Detection System (ODS) considered in this paper.

Regarding the robotic light treatment of strawberries, [8] presents a list of potential hazards and failure modes during UVC treatment identified using FMEA. Table I gives one example. There, the probability of a human getting injured by the UVC light when varying the occurrence of the failures is quantified. The results indicate that the riskiest scenario occurs when the robot is unaware of a human approaching from the side while transitioning between rows (Fig. 2).

Based on [8], [2], we study here the probability of the robot being unaware of the presence of a human during row transition until it is too late to act. Our focus is on the risk of human-robot encounters during row transition (F-G5 in Table I), but the fundamentals apply to other hazards and scenarios. For the current example, the field transition, in which the robot moves between different parts of the field (see Fig. 2), also presents risks we can address.

III. Light-Treatment Use-Case

The UVC treatment is a preventive measure to be applied weekly to control powdery mildew. With this frequency, there is a non-negligible risk of possible robot encounter with field workers, untrained people and visitors, kids, and animals [13]. A mitigation plan to reduce risks should consider the distance from the lamp and exposure time [14].

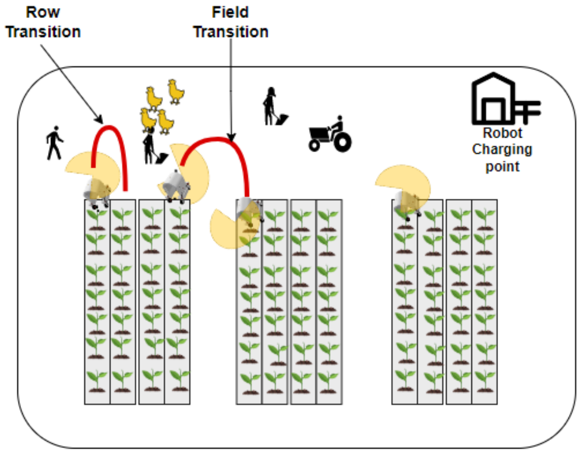


Fig. 2: Robot(s) performs farming operations. The robot can perform different tasks along the rows and then do a transition at the end of row. During rows/fields transition, it might encounter other living or obstacles. We are interested in row transition risk during UVC treatment.

A. Robot protection system

Fig. 3 demonstrates the robot protection layers, which are activated based on the distance of a human from the robot. Hazard zones are defined as follows. The green zone is the area more than 7 m away from the robot. It is outside the influence of the system but can be monitored to assess the potential for interaction. The yellow area is where a person may be approaching and can be monitored by the system; it is estimated to be between 3 and 7 m from the robot. In the red zone, harm may occur: 0-3 m from the robot [8].

The robot's safety system has three main components. The primary system uses visual and audio warnings. The secondary sensor-based system includes ODS, which can be one or a combination of distinct sensor types, most prominently 3D camera and LiDAR, as these are typically also utilised for navigation.The final safety system includes impact-recognition bumpers and emergency stop buttons.

B. Operational assumptions and safety property

Here, we present a way to quantify the probability of human injury, or more precisely, of humans being at high risk inside the red zone during UVC treatment. Afterwards, we discuss mitigation plans to reduce the risk factors. The following assumptions have been considered to define the constants and probabilities used in the RoboChart model presented in Section V. First, any human entering the robot's operational area can move in an arbitrary direction. The robot does not carry any physical barrier that would prevent humans from entering the red zone. Second, the increased safety implied by the existing primary protection system is incorporated and accounted for in the approaching decision probabilities into the different zones. Third, our analysis focuses on ODS based on sensor readings that allow the detection of human presence within a distance of over 7m. Finally, the damage is considered to have occurred once

TABLE I: Hazard identification during the studied scenario, from [8].

Possible situation	Code	Possible failures	Potential effect	Consequence	Severity	Occurrence
Robot at the end of the rows when a worker is approaching laterally	F-G5	Robot detects the human only when they are too close (less than 3.6 m)	Robot stop using UV-C light too late	Human is getting injured by the UV-C light	critical	probable

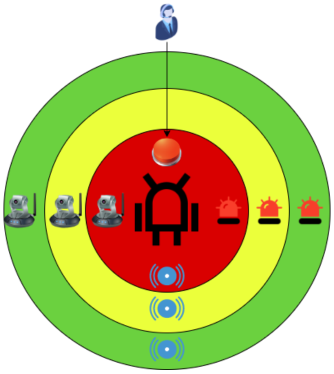


Fig. 3: Multi-layered protection system. To minimize human injuries, multi-layered protection system is deployed. Anomalies can be detected by sensors, camera and LiDar readings on different distances (green-yellow-red regions). Warning sub-system and emergency stop button are also included

the human enters the red zone. This situation causes human injury and is used to define the safety property in Section V.

According to the hazard identification in [8], we notice a major risk associated with operation O-U1, failure F-G5. We aim to reduce this risk by reducing its occurrence probability. Utilising the ODS as a Safety Instrumented System (SIS) and reducing the maximum tolerable occurrence probability, a higher Safety Integrity Level (SIL) can be obtained [5], [15]. We can reduce the risk associated with F-G5 by an order of magnitude so that it reaches tolerable levels.

IV. Probabilistic model checking using RoboTool and PRISM

Formal verification ensures that a system fulfils given specifications in all circumstances regardless of input possibilities [16], [17], [18]. In our use case, machine learning components are enabled in the robot. Due to the probabilistic nature of such components, probabilistic model checking is used to capture the behaviour of the perception system and its interaction with the environment [19], [20].

The UVC treatment and hazard scenario F-G5 from Table I is modelled as a Discrete-Time Markov Chain (DTMC). The transition between one state to another is assumed to be deterministic and depends on event probabilities.

RoboChart is a domain-specific language for model-based robotics software engineering, with formal semantics encompassing functional, timed, and probabilistic aspects tailored for formal verification. Its tool is called RoboTool[1] [21], [22]. Capturing the UVC-treatment behaviour in an abstract and formal modelling notation like RoboChart can be challenging, especially for practitioners who are used to working with code-driven, dynamic simulations based on sketches of system decomposition and design. This can be mitigated by training, and capturing the behaviours in PRISM directly is even more challenging for practitioners without knowledge of formal methods and PRISM. In our use case, for example, it is difficult to model the naturally asynchronous behaviour of the human, robot and ODS in terms of a global clock in PRISM. Here, asynchronous behaviour means they progress at their own pace but only at a time tick. In addition, RoboChart, through its diagrammatic notation, helps a developer to focus on the high-level design of the application, to write a correct model, instead of on analysis details. This can reduce verification cycles and improve efficiency.

A model checker requires use of bounded data types; it goes through all possible states and transitions to verify or disprove a property. Specific values for variables or inputs can be used, but that restricts the range of values analysed. Thus, keeping the right level of abstraction during modelling is essential for capturing all relevant aspects of the system, getting a meaningful result from model checking, and keeping the computational complexity at bay.

This paper uses RoboTool to automatically generate a PRISM model from our RoboChart model, and a PRISM property file from properties described using RoboChart's controlled natural language [10]. RoboTool runs multiple PRISM instances (229 for this use case and one for each property) in the background to analyse properties simultaneously. This procedure is fully automated. The generated PRISM model is larger than initially expected, as usual for automatically generated models. It captures not only the architecture of the RoboChart model, but also its complete semantics, such as all interactions between controllers and state machines, composite states, high-level transitions and actions in a low-level command-based PRISM language.

The two properties to be verified are formalised using the probabilistic temporal logic PCTL [23]. With RoboTool, we do not need to use PCTL to describe the properties: a controlled natural language is available, from which PCTL formulas can be automatically generated. In this section, we define the properties in PCTL and present the RoboTool facility in the next section. We define the property that captures the probability of human injury during UVC treatment on row transition using the operator P. It is used in a quantitative

[1] robostar.cs.york.ac.uk/robotool/

approach to reason about the probability of event occurrence in $P_{=?}[path\ property]$. The formalisation is as follows:

Property P1 (Injury):

$$P_{=?}\left[F\left(\begin{array}{l} shuman = inRed \wedge \\ srobot = transitionRow \wedge \\ ticks = t \end{array}\right)\right] \quad (1)$$

This is a query of the probability ($P_{=?}$) of the system finally (F) reaching a situation where the human is in the red zone ($shuman = inRed$), the robot is in the transition row ($srobot = transitionRow$), and the number of $ticks$ is that of a parameter t of the query.

The following property checks deadlock freedom.

Property P2 (Deadlock):

$$\neg E\ [F \text{ “deadlock”}] \quad (2)$$

This property requires that does not ($\neg$) exist (E) a path such that finally (F) the system deadlocks. Here, “deadlock” is a predefined label that identifies the states without outgoing transitions.

Next, we present our RoboChart models and properties as modelled using controlled natural language.

V. State Machines and Property Modelling

In Fig. 4, we show a RoboChart component model (in a block modUVC) describing the high-level architecture of the whole system. The use case implementation is available on RoboStar technology GitHub. [2] In that model, a robotic platform block rpUVC describes an abstraction of the robot via three shared variables: shuman, sods, and srobot, declared in the interface block stateInf to record the current status of each entity. In another interface eventInf, we define an event tick to synchronise the behaviour of the entities. Inside modUVC, a controller block called ctrlUVC contains four state machines: ODSSTM, RobotSTM, and HumanSTM, to specify the behaviours of the entities, and EventRelaySTM, relaying the tick event from the platform to EventRelaySTM.

The connections between the platform and EventRelaySTM, and between EventRelaySTM and the other three state machines are asynchronous: a one-place buffer in RoboChart. Every tick from the platform adds to or overrides the buffer connected to EventRelaySTM. As shown in Figs. 5-8, the machines need to take a tick event from the buffer for a transition to be taken. After the event is taken, the buffer is empty, and the state machine has to wait till the next tick to progress. With the EventRelaySTM machine, each tick is passed on from the platform to the other machines. We believe this protocol is the most difficult to be implemented in PRISM directly without using RoboChart.

The variable ticks records the number of steps of the system (akin to the passage of global time). With that, we can formalise our property of interest, namely, the probability of a human entering a dangerous zone at different steps. With the constant N_ticks, we can bound the value of ticks as required for model checking (in PRISM). The value of N_ticks is fixed during translation to PRISM so that our verification can explore different values.

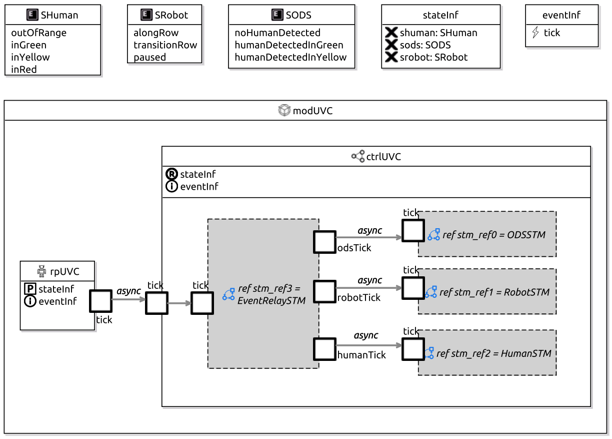


Fig. 4: UVC RoboChart component model. The controller receives clock-generated ticks. These ticks are distributed to three state machines: ODSTM, RobotSTM, and HumanSTM.

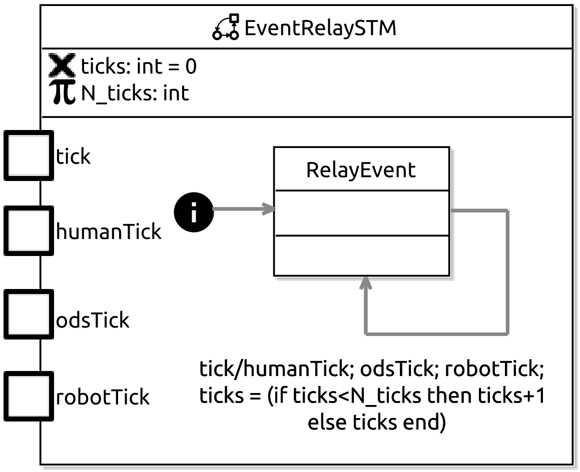


Fig. 5: UVC RoboChart model: event relay state machine. Keeps generating tick messages until N_ticks is reached.

The machines HumanSTM, RobotSTM, and ODSSTM start at the state targeted by the transition from their initial junctions (a black circle with an i). For example, HumanSTM in Fig. 6 enters the OutOfRange state, which captures behaviour when the human is out of the robot operation area. InGreenZone, InYellowZone, InRedZone are concerned with behaviour when the human's position relative to the robot is in each of the zones in Fig. 3. The transitions from one state to another have a trigger tick and go via a probabilistic junction where a choice is made based on a configurable probability defined by a constant whose value is left open. As explained later in Section V-A, each probability is based on awareness of the risk to get closer or leave the zone.

RobotSTM in Fig. 7 starts at the state MoveAlongRow, capturing UVC treatment along the row, and alternates

[2] https://github.com/UoY-RoboStar/uvc-case-study/

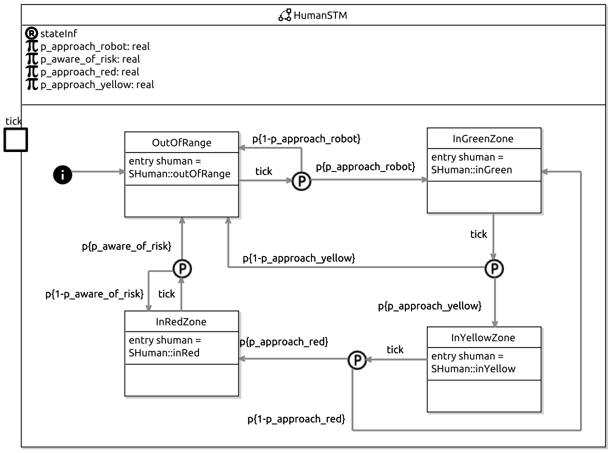


Fig. 6: UVC RoboChart model: human state machine.

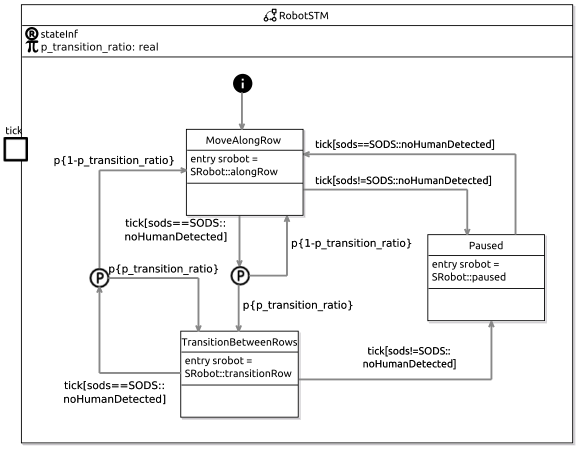

RobotSTM
stateInf
p_transition_ratio: real
tick
MoveAlongRow
entry srobot = SRobot::alongRow
p{1-p_transition_ratio}
tick[sods==SODS::noHumanDetected]
tick[sods!=SODS::noHumanDetected]
tick[sods==SODS:: noHumanDetected]
p{1-p_transition_ratio}
Paused
entry srobot = SRobot::paused
P
p{p_transition_ratio}
P
p{p_transition_ratio}
TransitionBetweenRows
entry srobot = SRobot::transitionRow
tick[sods!=SODS:: noHumanDetected]
tick[sods==SODS:: noHumanDetected]

Fig. 7: UVC RoboChart model: robot state machine.

between MoveAlongRow and TransitionBetweenRows with a ratio p_transition_ratio such as 10:1. This effectively implies that the robot spends ten-fold time in MoveAlongRow as compared to TransitionBetweenRows. If a human is detected in green or yellow zones, the robot will move to Paused via the transition with guard [sods!= noHumanDetected].

In Fig. 8, SODS enters NoHumanDetected upon initialisation. Afterwards, human position and system detection accuracy dictate the next state. The system then can move to HumanDetectedInGreen with accuracy p_ods_green or HumanDetectedInYellow with accuracy p_ods_yellow.

In the next section, we study the values of the probabilities, and in Section V-B, we formalise safety.

A. Probabilities and constants definitions

A human decides to get close to the robot with three different probabilities: p_approach_robot, p_approach_yellow, and p_approach_red. The decision is based on awareness and previous training. We differentiate between three levels of awareness as described in Table II. Finally, we set p_aware_of_risk to 0.01, since, as already said, the damage already occurs when a person is in the red zone.

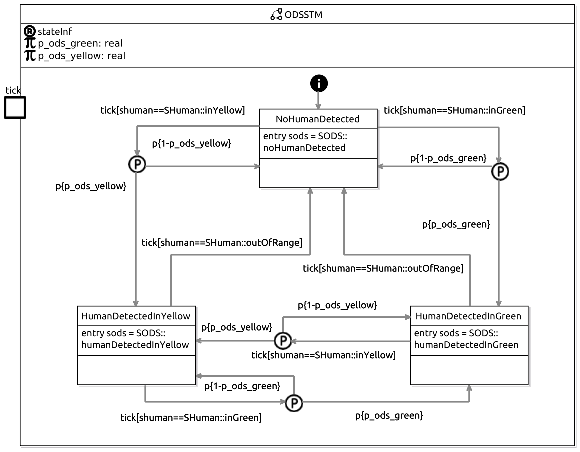

ODSSTM
stateInf
p_ods_green: real
p_ods_yellow: real
tick
tick[shuman==SHuman::inYellow]
NoHumanDetected
entry sods = SODS:: noHumanDetected
tick[shuman==SHuman::inGreen]
p{1-p_ods_yellow}
P
p{1-p_ods_green}
P
p{p_ods_yellow}
p{p_ods_green}
tick[shuman==SHuman::outOfRange]
tick[shuman==SHuman::outOfRange]
p{1-p_ods_yellow}
HumanDetectedInYellow
entry sods = SODS:: humanDetectedInYellow
p{p_ods_yellow}
P
tick[shuman==SHuman::inYellow]
HumanDetectedInGreen
entry sods = SODS:: humanDetectedInGreen
p{1-p_ods_green}
P
tick[shuman==SHuman::inGreen]
p{p_ods_green}

Fig. 8: UVC RoboChart model: ODS state machine.

TABLE II: Awareness levels in the experiment

Level	Description	Probabilities

Deliberate	A person is determined to reach the robot. For instance, if curious and unaware of the risk.	p_approach_robot=1
		p_approach_yellow=1
		p_approach_red=1

Aware	Even if the person has a higher chance of entering the green and yellow zones, entering the red zone is less probable.	p_approach_robot=0.5
		p_approach_yellow=0.5
		p_approach_red=0.3

Less Aware	The person is unaware of the risk but cautious about entering the red zone.	p_approach_robot=0.7
		p_approach_yellow=0.7
		p_approach_red=0.5

The *ODS* accuracy is characterised by two probabilities: p_ods_green is the accuracy in detecting an object within the green zone, and p_ods_yellow in the yellow zone. In the study, we set a higher value for p_ods_yellow since detecting an object closer to the sensor is easier.

The scenarios considered are as follows. (1) High performance ODS system: both p_ods_yellow and p_ods_green are 0.99. (2) Normal ODS: p_ods_yellow and p_ods_green are set to 0.7 and 0.4. (3) Non-functioning or lacking ODS system: p_ods_green and p_ods_yellow are set to 0.

B. Formulation of the safety property

As already mentioned, to specify properties and assertions, such as Properties **P1** and **P2** in section IV, RoboTool provides a simple and more readable textual domain-specific language. Listing 1 presents the safety property **P1**. Values of constants and probabilities are picked up from definitions in a configuration named `C1`. A configuration is just a list of constant names and their associated values, that we need to define in RoboTool to support proof my model checking. Listing 2 defines the deadlock property **P2**.

VI. Model-Checking Results

During the experiments based on Property **P1** using PRISM, the level of awareness and type of ODS system are changed. As expected and evident in all graphs depicted

```
import uvc_config::*

prob property P1:
  Prob=? of [Finally
    modUVC::rpUVC::shuman==inRed /\
    modUVC::rpUVC::srobot==transitionRow /\
    modUVC::ctrlUVC::stm_ref3::uvs==t]
  with constants C1
```

Listing 1: Property definitions in RoboTool of human injury. C1 denotes probability values in the configuration file.

```
prob property P2:
   not Exists [Finally deadlock]
   with constant C1
```

Listing 2: Deadlock property definition

in Fig. 9, the probability of injury during row transition is reduced when the awareness level of the human is increased.

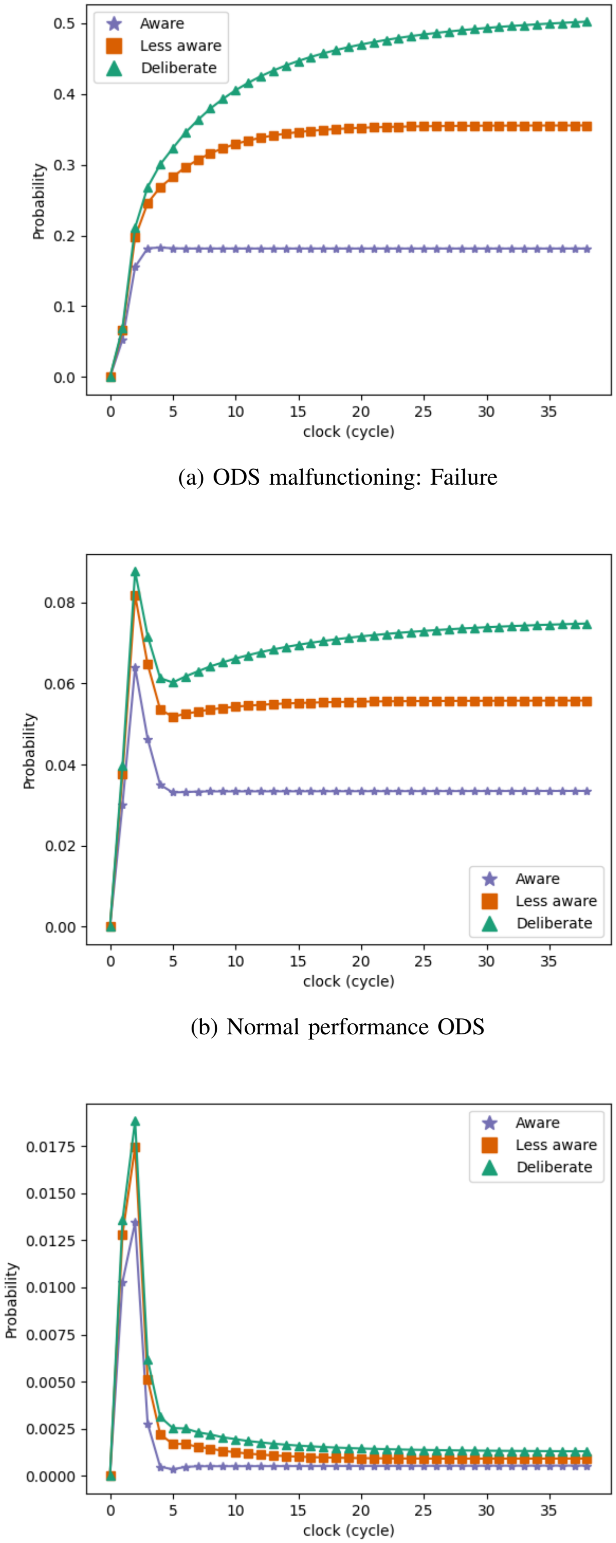
Next, attention is turned towards the effect of the ODS on the occurrence probability of hazard F-G5 as elaborated upon in Section I. Utilising the ODS for safety purposes helps reduce the occurrence probability and enables us to push the associated risk below an acceptable level. As F-G5 leads to human injury with high likelihood and thus the risk of this hazard is high [8], the maximum tolerable occurrence probability must be sufficiently low.

From Fig. 9a and Fig 9b, it is concluded that the probabilities of injury are above 0.1 and 0.01 when the ODS is not functioning or using an average-performing ODS. Fig. 10 compares the effect of hazard occurrence probability as a function of ODS type. This figure presents results considering both deliberate and aware human behaviour and depicts how accurate and high-performance sensors reduce hazard occurrence probability. Compared to not using any ODS or having a malfunctioning one, the normal ODS with average quality provide a risk reduction factor 10. This represents a Safety Integrity Level (SIL) of 1. Opting for a high-performance ODS yields a risk reduction factor of 100, representing SIL 2. These results emphasise the importance of improving the ODS system when designing a safety architecture for agriculture RAS [11].

To increase the confidence in the correctness of the model, it was verified that the UVC state machine could not deadlock; that is, Property **P2** is also fulfilled.

VII. Concluding Remarks and Future Work

Our main contributions are a risk analysis and safety assurance approach for agricultural robots. We use probabilistic model checking to quantify the risk of human injury due to UVC-light exposure. The results give insight and guidelines during the early development phase on improving the safety system and implementing a risk mitigation plan. These include implications of improving the detection algorithm, adopting a higher performance and more expensive sensors,



(a) ODS malfunctioning: Failure

(b) Normal performance ODS

(c) High performance ODS

Fig. 9: Probability of human injury when encountering the robot during row transition for various ODS safety systems and human awareness levels.

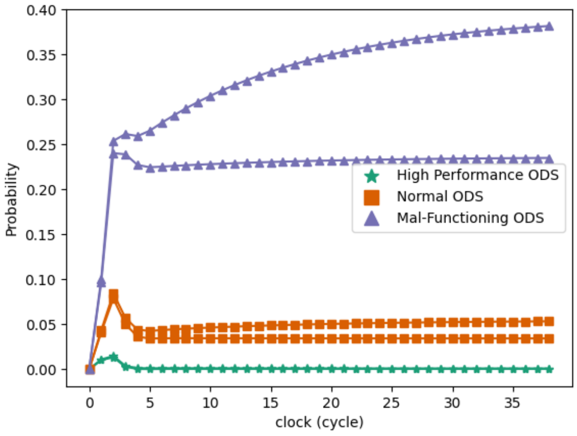


Fig. 10: Comparison between different ODS systems when a person is aware and deliberately approaching the robot. The risk is reduced by an order of a magnitude when high-performance ODS is applied while awareness does not reduce risk level to an acceptable level

or improving the safety policies through more elaborate warning systems to increase human awareness.

As Functional Safety Assessment is needed in all phases of development, a natural progression of the current work includes: (1) engineering and realisation of (high-performance) ODS including machine-learning components [11]; (2) development and formal verification of navigation and control laws for the agricultural robot platform, taking both hardware and software components into account explicitly (co-verification) [24]; (3) formal verification and validation of ODS performance during the operation phase; and (4) run-time monitoring and verification of Property **P1**.

Acknowledgements

The authors would like to gratefully acknowledge all guidance and fruitful discussions provided by Robert Morris on the hazard and safety aspects of agricultural robots. The research presented in this paper has received partial funding from the Norwegian Research Council (RCN) Robo-Farmer, project number 336712, the UK EPSRC Grants EP/M025756/1, EP/R025479/1, and EP/V026801/2, and the Royal Academy of Engineering Grant No CiET1718/45.

References

[1] A. Suthaparan, K. A. Solhaug, N. Bjugstad, H. R. Gislerød, D. M. Gadoury, and A. Stensvand, “Suppression of powdery mildews by UV-B: Application frequency and timing, dose, reflectance, and automation,” *Plant Disease*, vol. 100, no. 8, pp. 1643–1650, 2016, pMID: 30686239.

[2] M. Forges, M. Bardin, L. Urban, J. Aarrouf, and F. Charles, “Impact of UV-C radiation applied during plant growth on pre- and postharvest disease sensitivity and fruit quality of strawberry,” *Plant Disease*, vol. 104, 06 2020.

[3] M. Fargnoli and M. Lombardi, “Safety vision of agricultural tractors: An engineering perspective based on recent studies (2009–2019),” *Safety*, vol. 6, no. 1, 2020.

[4] ISO Central Secretary, “Agricultural machinery and tractors — safety of highly automated agricultural machines — principles for design,” International Organization for Standardization, Geneva, CH, Standard ISO 18497:2018, 2018.

[5] D. J. Smith and K. G. Simpson, *The Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety: IEC 61508 (2010 Edition), IEC 61511 (2015 Edition) and Related Guidance*. Butterworth-Heinemann, 2020.

[6] F. A. Batarseh, L. Freeman, and C.-H. Huang, “A survey on artificial intelligence assurance,” *Journal of Big Data*, vol. 8, no. 1, p. 60, 2021.

[7] O. Zaki, K. Brown, J. Fletcher, and D. Lane, “Detecting faults in heterogeneous and dynamic systems using dsp and an agent-based architecture,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1112–1124, 2007.

[8] S. P. Leonardo Guevara, Marc Hanheide, “Medium-sized AGV for soft-fruit production (MeSAPro) - technical annex,” Lincoln Centre for Autonomous Systems, Tech. Rep., 2021. [Online]. Available: https://www.york.ac.uk/assuring-autonomy/demonstrators/robots-to-support-farming/mesaprofinaltechnicalreport/

[9] A. Miyazawa, P. Ribeiro, L. Wei, A. L. C. Cavalcanti, J. Timmis, and J. C. P. Woodcock, “RoboChart: modelling and verification of the functional behaviour of robotic applications,” *Software & Systems Modeling*, vol. 18, pp. 3097–314, Jan. 2019.

[10] K. Ye, A. Cavalcanti, S. Foster, A. Miyazawa, and J. Woodcock, “Probabilistic modelling and verification using RoboChart and PRISM,” *Softw. Syst. Model.*, vol. 21, no. 2, pp. 667–716, 2022.

[11] J. C. Mayoral, L. Grimstad, P. J. From, and G. Cielniak, “Towards safety in open-field agricultural robotic applications: A method for human risk assessment using classifiers,” in *2022 15th International Conference on Human System Interaction (HSI)*, 2022, pp. 1–6.

[12] J. C. Mayoral Baños, P. J. From, and G. Cielniak, “Towards safe robotic agricultural applications: Safe navigation system design for a robotic grass-mowing application through the risk management method,” *Robotics*, vol. 12, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2218-6581/12/3/63

[13] Z. Xie, J. Fan, M. T. Charles, D. Charlebois, S. Khanizadeh, D. Rolland, D. Roussel, and Z. Zhang, “Preharvest ultraviolet-c irradiation: Influence on physicochemical parameters associated with strawberry fruit quality,” *Plant Physiology and Biochemistry*, vol. 108, pp. 337–343, 2016.

[14] J. C. S. Yam and A. K. H. Kwok, “Ultraviolet light and ocular diseases,” *International ophthalmology*, vol. 34, no. 2, p. 383—400, 4 2014.

[15] W. G. Gulland, “Methods of determining Safety Integrity Level (SIL) requirements - pros and cons,” in *Practical Elements of Safety*, F. Redmill and T. Anderson, Eds. London: Springer London, 2004, pp. 105–122.

[16] M. Luckcuck, “Another tool in the box: Why use formal methods for autonomous systems?” *CoRR*, vol. abs/2012.00856, 2020.

[17] D. A. Peled, *Formal Methods*. Cham: Springer International Publishing, 2019, pp. 193–222.

[18] F. Ingrand, “Recent trends in formal validation and verification of autonomous robots software,” in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 321–328.

[19] M. Kwiatkowska, G. Norman, and D. Parker, “PRISM: Probabilistic model checking for performance and reliability analysis,” *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, p. 40–45, 03 2009.

[20] ——, “Probabilistic model checking and autonomy,” 2021.

[21] J. Baxter, A. Miyazawa, P. Ribeiro, and K. Ye, *RoboTool* RoboChart *Tool Manual*, University of York, 12 2021.

[22] A. L. C. Cavalcanti, W. Barnett, J. Baxter, G. Carvalho, M. C. Filho, A. Miyazawa, P. Ribeiro, and A. C. A. Sampaio, *Software Engineering for Robotics*. Springer, 2021, ch. RoboStar Technology: A Roboticist’s Toolbox for Combined Proof, Simulation, and Testing, pp. 249–293.

[23] W. Wan, J. Bentahar, and A. Ben Hamza, “Model checking epistemic–probabilistic logic using probabilistic interpreted systems,” *Knowledge-Based Systems*, vol. 50, pp. 279–295, 2013.

[24] Y. Murray, M. Sirevåg, P. Ribeiro, D. A. Anisi, and M. Mossige, “Safety assurance of an industrial robotic control system using hardware/software co-verification,” *Science of Computer Programming*, vol. 216, p. 102766, 2022.