Medium-Sized AGV for soft-fruit Production (MeSAPro) - Technical Annex

Leonardo Guevara Marc Hanheide Simon Parsons
Lincoln Centre for Autonomous Systems

Version 1.0

1 Executive Summary

The use of Robotics and Autonomous Systems (RAS) in agriculture is an emerging area which can significantly improve the food production sector in the UK. The aim of the MeSAPro project is to lay down the basic requirements for using RAS in soft fruit production from the perspective of ensuring system safety during Human-robot interaction (HRI). Thus, in this document, we provide some guidelines from our work so far, addressing aspects of occupational health (which deals with assuring both physical and mental welfare) in scenarios where agricultural robots are intended to share the workspace with people in the near future. The objectives met by the MeSAPro project are listed below. The work carried out and results achieved under each of them are described in the sections below.

- **Definition of scenarios**: A total of four representative agricultural scenarios were covered in this project including: plant treatment using UV-C light, data gathering using a mobile robotic platform, automatic fruit picking, and robot-assisted harvesting using robots for transportation purposes.
- **Hazard analysis and mitigation strategies**: A hazard analysis was performed for the four scenarios studied. Failure modes were identified and general safety policies were defined. A three-layered safety architecture was proposed to tackle the hazards/failures.
- **Formal verification**: The HRIs present within the four scenarios were translated into a mathematical representation in order to perform a formal verification based on probabilistic model checking. The model checking analysis focused on quantifying the probabilities of producing human injuries when the proposed hazard mitigation strategies are applied under failure modes.
- **Integration and evaluation**: The proposed mitigation strategies were integrated into a commercial agricultural robot. The robot-assisted harvesting scenario was chosen as case study to test the robot safety operation. An user experience assessment was carried out in order to determine qualitatively and quantitatively the safety and trust perception.

2 Definition of Scenarios

Figure 1 gives a schematic of the overall structure of the kind of soft-fruit (strawberry) farm in which the robots will operate. There are three main locations that concern us. The polytunnels are the main focus of the work of the robots. The Figure 1 shows one set of polytunnels, in reality a farm may contain multiple sets of polytunnels, and these may be located in non-contiguous areas. The other locations on the farm are the robot shed and the collection point. The shed is where the robots are located when not in use, where they recharge, and where they are serviced. The collection point[1] is, as the name suggests, where the fruit is collected. As we will see below, in different scenarios, the robots traverse footpaths to move between various of these different locations. The final location of interest is the control centre. This is a remote location from which we envisage that the robots will be monitored during operations.

[1] The fruit will often be moved to a packhouse for packaging and storage until the fruit leaves the farm. However, there is typically one packhouse which is near the main farm buildings. In an individual field the fruit will first be amassed at a collection point from which it will be driven to the packhouse. Since this journey will often be over public roads, we do not consider it to be part of the world in which the robots operate.

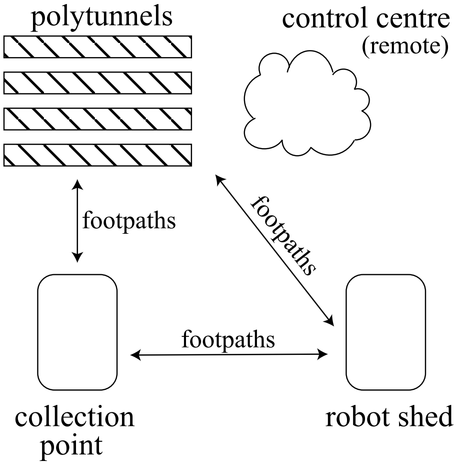


Figure 1: The major elements of a farm on which the scenarios are set.

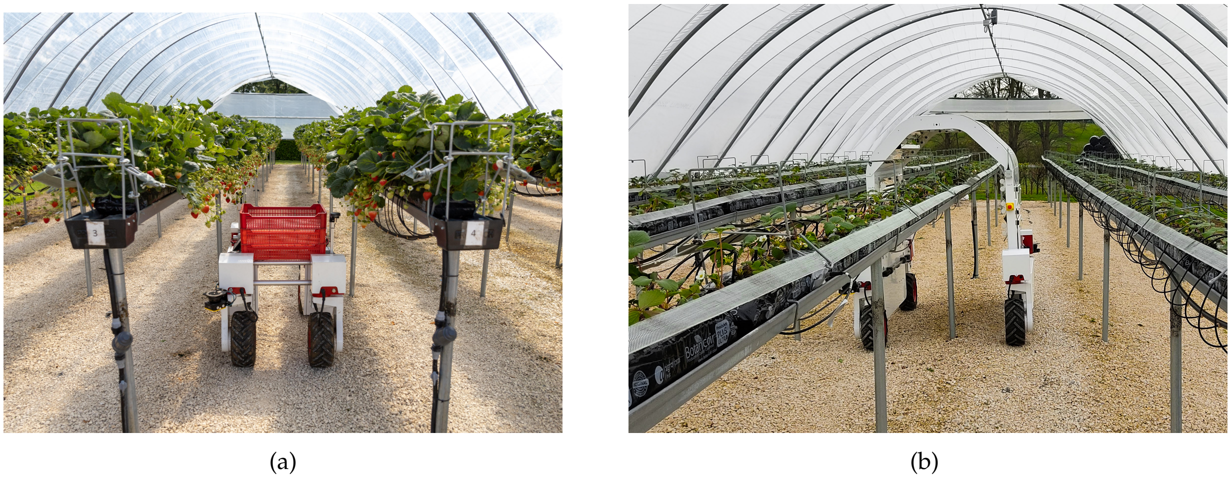
2.1 Scenario Description

Based on the farm structure mentioned above, we studied four scenarios that are typical of current and projected future use of robotic platforms such as Thorvald II [14] in soft fruit production. A description of each scenario is given as follows:

- **UV-C Treatment scenario**: In commercial growing operations, plants are typically sprayed with various pesticides in order to keep diseases, such as powdery mildew, at bay. However, with a growing interest in reducing the use of chemicals, there is interest in using robots that can treat strawberry crops with UV-C light which has been demonstrated to be effective against powdery mildew [15]. The Thorvald II robot configuration used during the UV-C treatment is presented in Fig.2(b), where the robot is moving along the rows and straddles the tables on which the strawberries grow so that the UV-C emissions are directed inwards. The UV-C dose is carefully calibrated to not damage the strawberry plants but it can harm any other living thing that come closer than 7m to the robot. Thus, it is mandatory to restrict the access to the polytunnels during UV-C treatment. However, it is always possible that untrained people decide to come close to the robot to have a look, or accidentally finds themselves too close to the robot. For these unplanned HRIs it is crucial that the robot safety system can get an early human detection in order to alert the human of the danger and stop UV-C operations immediately.
- **Picking scenario**: The picking refers to the scenario when a harvesting robot performs automatic picking operations. In this scenario, the robot is moving slowly along the rows while a manipulator on it is picking the fruits and placing them on the trays. During this operation, the Thorvald II robot uses the same configuration showed in Fig.2(a) (i.e it is moving between the crop rows), but the harvesting platform includes a robotic manipulator which perform the picking task. Since the current harvesting robots do not have good enough harvesting rates to be an efficient solution on their own, then the harvesting operations still require human pickers. Thus, similar than the logistics scenario (see below), in the picking scenario the robots share the workspace with humans but unlike the logistics scenario, they do not require human interaction during the picking process to accomplish their tasks. The only situation when robots require a human interaction is for loading/unloading of trays of harvested fruit at collection points. Thus, we can treat this scenario as a complementary operation to the logistics scenario.
- **Logistics scenario**: We use the term *logistics* to refer to scenario in which a service robot assists human pickers in transporting fruit that has been picked, allowing human pickers to concentrate on picking fruit. As in a traditional picking operation, human pickers remove the fruit from the plant and place them in small containers, the punnets that are familiar from the supermarket. These containers are then placed in larger trays. The picker works with a given tray until it is full, and then summons a robot. The full tray is placed on the robot, which also brings a new tray, and the robot then leaves the current picker location, moves to the end of the row of plants, and waits there until a new picker summons it. When the robot is fully loaded, it transports the full trays to the collection point. To do this the robot has to navigate along paths outside the

polytunnel. The Thorvald II robot configuration used during the logistics is presented in Fig. 2(a) where it can be seen that the robot moves between the crop rows transporting trays with fruits.

- **Scouting scenario**: The term *scouting* refers to the scenario when the robot operates in autonomous way, following a predefined pattern inside the polytunnel, in which the fruit is typically grown in the UK, in order to collect data about the crop. For example, the robot may count fruit in order to make yield forecasts. The robot can traverse the polytunnel using: i) the same configuration and pattern as for UV-C application (allowing each plant to be imaged from both sides simultaneously, or ii) using the same configuration and pattern as for harvesting operations (imaging the plants from one side at a time. Currently, scouting is performed using a robot in the UV-C treatment configuration. Typically, the data collection is an independent task, however, it is possible to perform the data collection in parallel with the picking or logistics operations. The latter means that from a safety perspective, the scouting scenario does not introduce hazards that aren't already covered by the other scenarios. For this reason, in the following sections, we will only deal with UV-C treatment, picking, and logistics operations.


(a)

(b)

Figure 2: The Thorvald II configuration used to: (a) perform logistics and picking during fruit harvesting operations; and (b) perform the UV-C treatment and crop monitoring.

Table 1 summarizes the four scenarios covered by the MeSAPro project, pointing the actor or actors who are assumed to perform the tasks and the level of interaction between them (if it is the case). According to this table, it is assumed that the UV-C treatment and the scouting scenarios are performed by the robots without human interaction, the picking scenario requires the robots sharing the work space with human pickers and the logistics requires a higher level of coordination between pickers and robots, i.e. planned HRIs. However, the assumption of having only planned HRIs cannot be guaranteed in practice since untrained people may approach the robots during their operations. This can happen because, unlike the case of industrial robots, robots used in agriculture operate in the open, and the farms on which they are based are often traversed by public rights of way — see [26] for examples on one of our partner farms. These kinds of unexpected situations also called *failure modes* make it necessary to perform a hazard analysis to determine the effects of potential failures and ways to mitigate them.

Table 1: List of agricultural scenarios studied in MeSAPro with the corresponding interaction level between the robots and the human workers.

Scenario	**Actors**		**Interaction level**		
	Human	**Robot**	**unshared workspace**	**shared workspace**	**cooperative co-work**
UV-C treatment		X	X		
Scouting		X	X		
Picking	X	X		X	
Logistics	X	X		X	X

2.2 Initial assumptions

To complement the descriptions given in Subsection 2.1, we defined various operational assumption for each of the four defined scenarios. These assumptions are based on current farm and operational settings. Some of these assumptions are valid for a single scenario but others of them cover more than one at a time.

2.2.1 General assumptions

A-G1 It is assumed the working staff are well trained and aware of how the robot behaves according to each kind of operation mode.

A-G2 The robot maximum speed is restricted according to the operation mode and the workspace (inside polytunnels or on footpaths). It is assumed that for all the scenarios which require the robot operates along footpaths, the maximum robot speed is 1 m/s.

A-G3 During close HRIs, the robot is allowed to move next to a human while staying within the so-called social zone (from 1.2m to 3.6m) or public zone (above 3.6m) defined in [28]. The personal or intimate zones (less than 1.2m) are not considered admissible for a safe and sociable acceptable navigation since it is likely to get injuries product of collisions within these zones.

A-G4 The robot is supposed to be equipped with the necessary sensors to detect the human presence (e.g. LiDARs, RGB-D cameras) and the necessary hardware to send audiovisual messages that can be interpreted by the human workers (e.g. speakers and colored beacons).

2.2.2 Assumptions for logistics and picking scenarios

A-LP1 In logistics operations, the pickers can summon a robot from any point in polytunnels to pick full trays. Thus, it is assumed that the robot is capable of locating the picker based on GNSS information, and when the robot is getting closer to the picker, the localization and tracking rely on LiDAR/camera readings.

A-LP2 The robot payload capacity is limited, meaning it requires at least one trip to the collection point to unload the fruit and replace the full trays with empty ones.

A-LP3 There is at least a collection point strategically located in a central point with respect to all polytunnels. This collection point is where the harvested fruit is collected before trolleys transport it to the food handling point where the fruit will be cleaned, packed, and refrigerated before it is sent off to the markets.

A-LP4 There may be robots doing logistics tasks (i.e. summoned by pickers to transport fruit) sharing the workspace with the robots which perform specific picking tasks.

A-LP5 During the *picking mode* operation, the robot is largely stationary (less than 0.1 m/s), only moving a few centimeters between two plants after it has picked all the reachable ripe fruit on the first. During *logistics mode*, the robot can move up to 0.5 m/s inside the polytunnel, while the maximum speed outside the polytunnels is restricted according to **A-G2**.

2.2.3 Assumptions for UV-C treatment scenario

A-U1 The UV-C treatment is carried out at night. This is done because mildew is more vulnerable to UV-C treatment at night. However, it also has the advantage that it minimises the chance that the robot comes into contact with any people.

A-U2 UV-C light is only used when the robot is in the polytunnels.

A-U3 It is assumed that there will be no workers (trained people) at polytunnels when UV-C treatment is being performed.

A-U4 The safety minimum distance to avoid human injuries during UV-C light emissions is 7m (within the public zone).

2.2.4 Assumptions for Scouting scenario

A-S1 For this analysis, data collection is considered to be performed as a primary task. In other words, we are not considering scouting being combined with picking or logistics operations. Thus, similar to the UV-C treatment scenario (**A-U3**), scouting is assumed to be performed without sharing the workspace with human workers in the polytunnel, but unlike the UV-C treatment, in this application the minimum distance required to ensure a safe interaction is not restricted by 7m as in the UV-C treatment (**A-U4**).

A-S2 During data collection, the robot moves relatively slowly having a maximum speed of 0.5 m/s inside the polytunnels.

2.3 Robot Operation Modes

Before start identifying the potential hazards, we have to define a list of the most relevant operations that the robot will perform based on the scenario descriptions in Subsection 2.1. Since some robot operations are the same in several scenarios, the potential failure modes derived from these situations will be repeated in several scenarios. Thus, we first describe the operations that overlap more than one scenario, and then only mention them in the scenarios where they are repeated.

2.3.1 General Operations

The robot operations which differentiate each scenario from another are performed inside polytunnels. However, all the scenarios requires the robot to perform operations outside the polytunnels while they are moving along footpaths to any component of the scheme in Fig. 1. The two main operations which summarize these overlapping situations outside the polytunnels are listed below.

O-G1 **Robot moving between shed and polytunnels:** In all scenarios, the robot will move between shed and polytunnel. For most scenarios a typical robot "shift" will involve the robot leaving the shed/charging station, moving to a polytunnel, carrying out its scenario-specific operations, and then returning to the shed to charge or at end of the day.

O-G2 **Robot moving between polytunnels:** In a large-scale agricultural environment, the robot usually completes the task given in one polytunnel and then moves to another. While we distinguish the case for completeness, the hazards are the same as in case **O-G1** , and so we only deal with **O-G1** from here on.

O-G3 **Robot moving between polytunnels and collection point:** In logistics support the robot will frequently move between the polytunnel and collection point to unload full trays and replace with empty ones. Similar to **O-G2**, while we distinguish the case for completeness, the hazards are the same as in case **O-G1**, and so we only deal with **O-G1** from here on.

Thus, despite the differences between movements in these three cases, we consider all of the movements outside the polytunnel to be equivalent from the perspective of safety considerations.

2.3.2 Logistics and Picking Operations

Since the logistics and picking scenarios correspond both to harvesting operations. The robot operations in these scenarios are expected to overlap. For instance, when a robot is performing picking tasks, we can think of it working in two modes: *logistics mode* and *picking mode*. The picking mode refers to the behavior when the robot is navigating along the rows at a very low speed in order to collect the fruit autonomously. The logistics mode refers to a similar behaviour that the robot has in a logistics scenario, specifically when the robot navigates through the polytunnels with the objective of transporting full containers to the collection point and replacing them with empty ones. A picking robot will therefore perform all the same operations as a logistics robot and have all the same failure modes, in addition to the operations and failure modes from the picking mode. In this context, the robots operations during the harvesting can be split into four main operations. The first operation is **O-G1** which overlaps with all the scenarios. The next two operations (**O-L1** and **O-L2**) are typical for logistics but also overlap with picking scenario, and the remaining operation **O-P1** is unique for robots performing picking tasks.

O-L1 Planned interactions with workers to load/unload trays at polytunnels and/or collection point: This robot operation refers to the cases when the robot is required to interact with a human workers at the collection point or with pickers inside the polytunnel. In both, picking and logistics scenarios, the interaction with workers at the collection point is expected. However, in case of interaction with pickers in the polytunnels, it is only mandatory for the logistics scenario, but is not required for the picking scenario. In these planned interactions is necessary for the robot to detect and track humans precisely and stop at a safe distance from workers (restricted by **A-G3**). Moreover it is important to communicate the robot intentions to the human and make the robot know about the human intentions to avoid inefficient stops each time a human is detected.

O-L2 Robot in *logistics mode* moving inside the polytunnel: This operation refers the robot moving inside the polytunnel while transporting empty trays from collection point to an specific point at the polytunnel and/or when transporting filled trays from polytunnels to collection point. This operation covers the cases when the robot moves along the rows and it is performing transitions between rows. Since the robot is sharing the workspace with multiple pickers, it is expected to have unplanned interactions during robot navigation. These unplanned interactions are especially critical at the end of the rows because of the lack of visibility of pickers to the robot. Note that the robots do not typically turn at the end of the rows — since the robot is holonomic, it is typically easier and quicker for the robot to move sideways at the end of the rows when it moves between them. During this maneuver, if a worker is approaching to the robot laterally (that is from the side), the crop may occlude them from the perspective of the robot, and that occlusion may mean that the sensors (which only point forward) do not detect the human presence in time.

O-P1 Robot in *picking mode* moving inside the polytunnel: This operation is performed by the robot when it is moving inside the polytunnel and there is still payload capacity to place fruit on it. During this operation the robot is moving slowly (according to **A-LP5**) while the robot manipulator is picking the fruit and placing it on the trays. The robot may find human pickers approaching its location. The robot policy when a HRI happens depends on what it infers the human's intentions to be. If the robot infers that the human is approaching to unload fruit on it, then the robot must stop until the picker is far enough away that it can continue moving. If the robot infers that the human is also picking, then the robot can continue working but needs to keep audiovisual alerts[2] activated to let the picker know about the robot's intentions.

2.3.3 UV-C Treatment Operations

During the UV-C treatment, the robot behaviours can be reduced into two main operations. The first operation is the general **O-G1** expected for all the scenarios, and the second is described below:

O-U1 Robot performing UV-C treatment inside the polytunnel: This operation refers to the robot following a predefined path while applying UV-C on the crop. During UV-C treatment the robot is continuously monitoring for human presence. If a human is detected more than 7m away, human injuries can be avoided by the robot stopping operations completely[3] and turning on audiovisual alerts to inform the human about the danger. However, the robot may fail to detect the human presence even if the human approaches within 7m. This risky situation may happen when a human located at the end of the rows is occluded by the crop even if the robot can easily detect a human 7m away when they are not occluded. Since UV-C treatment is performed at night time, the robot is very visible during treatment as UV-C light is very prominent and can be noticed from a long distance away.

2.3.4 Scouting operations

Similar to the UV-C treatment scenario, in the scouting scenario robot behaviour can be summarized in terms of two main operations. The general operation outside the polytunnel **O-G1** and the operation inside the polytunnels described as follows:

O-S1 Robot collecting data while moving inside the polytunnel: During this operation the robot is moving relatively slowly, following a pre-defined path while the visual sensors are taking data from the crops. It

[2] For the hazard analysis it was assumed that the robot is equipped with speakers and colored beacons which are used as a audiovisual feedback system to let the human know about the robot's intentions and potential hazards of approaching. The robot was subsequently equipped with these elements.

[3] The UV-C treatment is halted both to avoid hurting the human if they approach closer, but also to avoid damaging the crop.

is assumed that polytunnel access is restricted during this operation, meaning HRIs are not likely to occur. However in case that a human is detected by the robot, the robot should decide whether to continue working (but keeping active audiovisual alerts to let the human know about the robot intentions) or stop operations until the human moves back. Thus, this robot operation can be seen as a variation of the **O-P1** operation, where the differences are in the robot speed while performing each task and that during **O-P1** the unplanned interactions include trained pickers while in **O-S1** the unplanned interactions are expected to be only with untrained people.

3 Hazard Mitigation Strategies

This section provides a hazard identification for the four scenarios described in Section 2. The analysis identifies a list of potential failures which we then examine to provide strategies that can mitigate the negative effects of these failures.

3.1 Risk assessment metrics

To assess the risk of each hazard consequence identified during hazard analysis, we will follow the guidelines given by the authors of [32] who presented a qualitative risk classification matrix based on the example presented in the safety standard IEC 61508 'Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems' (International Electro-technical Commission, 2000). This classification matrix is presented in Table 2. The matrix is organized into a series of columns, which define the hazard consequence severity, and rows denoting the frequency that the hazard could occur. Where a row and column intersect determines the risk class. Four risk classes are possible, from the most severe (Class I) to the least severe (Class IV). The process of assigning risks classes involves examining each hazard identified during hazard analysis and determining both the frequency with which the hazard is likely to occur and the severity of the consequences associated with the hazard. If the risk class is found to be either Class I (intolerable) or Class II (undesirable), then a safety requirement must be produced, which details how the risk will be reduced to Class IV (negligible) or Class III (tolerable).

Table 2: Risk classification based on occurrence and severity criteria (see [32]).

Ocurrence	**Severity**			
	Catastrophic	Critical	Marginal	Negligible
Frequent	I	I	I	II
Probable	I	I	II	III
Occasional	I	II	III	III
Remote	II	III	III	IV
Improbable	III	III	IV	IV
Incredible	IV	IV	IV	IV

3.2 Hazard identification

The hazard identification is summarized in Table 3. As this Table shows, for each of the robot operations described in the previous subsection, the possible situations that are expected to occur in terms of HRI are listed. Then, for each situation, the potential failures were identified, concluding that there are a total of 10 failures to be analysed. Seven failures are generic failures which can happen in different operation modes, and the remaining three are unique for logistics and UV-C treatment operations. Given this base, the following analysis will discuss each of the these 10 failures pointing their potential severity and expected occurrence probability according to the specific situation.

3.2.1 Robot fails to detect physical contact with human (F-G1)

This failure mode can happen in every operation mode where the HRI drives to a collision risk (**O-U1** is not included). Thus, the robot must be equipped with sensors which enhance safety and intuitiveness of physical HRIs for both intentional and unintentional contacts. In the unfortunate case of imminent contact either by robot failures or human mistakes, the sensors can be used to trigger an safety stop which stops the robot immediately

Table 3: Hazard identification during the four scenarios studied.

Operation Code	Possible situations	Failure Code	Possible failures	Potential effect	Consequence	Potential severity	Expected occurrence	Risk level
O-G1 **O-L1** **O-L2** **O-P1** **O-S1**	Imminent collision is about to happen	**F-G1**	Robot fails to detect contact with human	Robot keep moving after collision	Human is getting injured	critical	remote	III
	A worker occluding the robot way	**F-G2**	Robot fails to detect the human above 3.6 m	Audiovisual alerts are not activated and then worker is not getting aware of the robot presence	Loss of efficiency due to excessive robot pauses and potential human injuries	marginal	remote	III
		F-G3	Robot is not visible to the human	Human is not aware of the robot approaching	Potential human injuries	critical	remote	III
O-G1 **O-L1** **O-U1** **O-S1**	Untrained human approaching to the robot with the audiovisual alerts activated	**F-G4**	Human can not interpret the audiovisual alerts	Robot requires to pause operations till the human get out of the robot way	Loss of efficiency due to prolonged robot pauses and potential human injuries	critical	remote	III
O-L2 **O-S1**	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 and worker are not aware each other presence on time	Potential human injuries due to collisions	critical	probable	I
		F-G2	Robot fails to detect the human above 3.6 m	Audiovisual alerts are not activated and then worker is not getting aware of the robot presence	Loss of efficiency due to excessive robot pauses and potential human injuries	marginal	occasional	III
O-G1	Robot tries to evade a human on its way	**F-G6**	Robot fails to predict human motion or intentions	Robot collides with the human or pause operations unnecessarily	Loss of efficiency due to excessive robot pauses and potential human injuries	critical	occasional	II
		F-G7	Robot fails to track accurately the human position					
O-L1	Robot detects more than one human in the same row	**F-L1**	Robot fails to distinguish which human detected was the one who summoned it	Robot try to approach to the incorrect worker	Loss of efficiency due to unnecessary pauses and potential human injuries	critical	occasional	II
	Robot starts reducing speed to finally stop next to the worker	**F-G6**	Robot fails to predict human motion or intentions	Robot fails to stop safely 1 m from the worker	Potential human injures	critical	occasional	II
		F-G7	Robot fails to track accurately the human position					
O-P1 **O-L1**	Picker approaching to the robot to place filled trays on it and replace with empty ones	**F-G6**	Robot fails to predict human motion or intentions	Human and robot collides with a low speed	Potential human injures	marginal	occasional	III
O-U1	Robot moving along a row while a human is approaching frontally	**F-U1**	Robot fails to detect the human above 7 m	Audiovisual alerts are not activated and then worker is not getting aware of danger	Potential human injuries due to UV-C light	critical	occasional	II
		F-U2	Robot fails to detect on time when a human is within 7 m	Safety stop is not activated and then robot keep using UV-C light	Human is getting injured by the UV-C light	critical	remote	III
	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	I

after an anomaly is detected. Human injury is not expected to be completely prevented by applying this safety stop, but at least the severity can be reduced. On the other hand, in the hypothetical case that the robot is not able to detect contact with a human during a collision, the robot directive will be keep moving, and then the severity of the human injuries is *critical*. The probability of occurrence of this kind of failure depends on the type of sensor implemented on the robot. A solution can be the use of sensorized flexible skin that could be mounted on strategic areas of the robot structure to detect pressure changes[4]. However since the impact can happen on an uncovered part of the robot, an alternative solution can be the use of force sensors attached to the robot actuators in order to detect any change in torque signal. Thus, by using a combination of these solutions, the occurrence of safety stop failure may be reduced to *remote*.

[4] This was one of the modifications that we subsequently made to the Thirvald robot that we were using for MeSAPro.

3.2.2 Robot fails to detect the human more than 3.6 m away (F-G2)

This failure can happen when the robot is operating inside and outside polytunnels. There are two possible situations, when the robot is approaching a human who was standing in the robot's path or when a human is approaching to the robot position. In both unplanned interactions, if the robot is not able to detect a human more than 3.6 m away, then no audiovisual alert is activated in time to ensure the human is aware of the robot's presence and intentions. This means, the human will not move away, or will keep approaching the robot, provoking the robot to trigger a safety stop to keep itself 3.6 m from the human. If this kind of unplanned HRI happens frequently, then the robot pauses may case a loss of efficiency in terms of robot usage and time wasted. Moreover if the human is still approaching to the robot and the robot fails to immediately cease operation, this could drive to a potential collision and consequently human injuries (specially outside polytunnels). To reduce the excessive waste of time during robot safety stops, the robot should be able to predict the human's intentions, and in case the human is mostly stationary, then the robot may be able to perform evasion maneuvers and continue its navigation. This alternative is only feasible on paths outside the polytunnels since inside polytunnels, the narrow space restricts any potential evasion maneuvers.

Since time spent and operational efficiency are not the main target of hazard analysis, the severity of this failure was assigned *marginal* considering that there is still an indirect collision risk. On the other hand, the occurrence probability depends on two factors: the reliability of the Human Detection System (HDS) and occlusion in the environment. For the moment, based on preliminary results of a LiDAR based HDS, the failure occurrence was assigned as *occasional* inside the polytunnel, and *remote* outside the polytunnels. This classification can be updated and improved after further tests.

3.2.3 Robot is not visible to the human (F-G3)

This failure was mainly derived from the scenario when the robot travels from the shed to a polytunnel at night to perform UV-C treatment. The robot visibility in polytunnels is ensured by the visible light emitted during UV-C treatment. However while the robot is moving outside, the robot is not performing any UV-C treatment, then another source of light is required to ensure the robot visibility to humans. Moreover, visibility is also an issue in the remaining scenarios, even though the robot operates during the day. This is because in a polytunnel, the robot may easily get into a position where the human workers are not able to see it — the robot may be ahead of them but out of their of line of sight (which tends to be focused on the plant to either side), or behind them. as a result, If the robot fails to continuously signal its presence, the human picker may easily collide with the robot. Because the kind of collision that may happen — when a human picker, for example, steps back into a stationary robot — the result of this failure may not result in injury but we still need to consider it. Therefore, considering the risky situations at night and during the day, this failure was placed in risk level III with a *critical* severity and a *remote* probability of occurrence.

3.2.4 Human can not interpret the audiovisual alerts (F-G4)

According to assumption **A-U3**, the farm, and especially the polytunnels, are considered closed areas where any unauthorized access is not allowed. However, in reality it is likely that there would be visitors who can be interested to take a closer look at robot operations, or people who have strayed off a footpath through the farm. These people will not have been trained, meaning they do not know about the potential hazards of approaching to the machinery and they will not be familiar with the robot behaviours according to the specific operations. For instance, for a robot in *logistics mode*, the directive when any unexpected human is detected closely is to pause operation until the human is far enough away for the robot to continue on its route. To avoid a long pause period, the robot uses audiovisual alerts to make people know that they are blocking its way. However since a non trained person is not familiar with these kind of alerts, they could stay longer occluding the robot provoking a loss of efficiency in terms of time spent. In case a robot is in *picking mode* the robot can continue working since the robot speed while picking is very slow. In this case the severity of the pauses is almost negligible. On the other hand, during UV-C treatment, since robot alerts are meaningless for this person, then the human may decide to keep approaching the robot and receive skin injuries.

Thus, assuming that the staff on the farm restrict the access to the polytunnels as much as they can (to satisfy the assumption **A-U3**), then, the chances of having unplanned interaction with untrained people can categorized as *remote*. Despite the failure being remote, the severity of its effects can be *critical* in the sense of loss of efficiency and human damage (specially during **O-U1**), that is the reason for classifying it as risk level III.

3.2.5 Robot detects the human only when they are too close (F-G5)

In certain situations, it can often happen that the robot becomes aware of the human presence only when it is very close (within the social zone according to **A-G3**), which does not give the robot time to take preventive action. In this context, this failure is similar to failure **F-G2**, where, according to the capability of the perception sensors and the occlusion from the environment, the maximum distance at which the perception system can detect a person will be affected. Occlusion by crops is a common problem during operations inside the polytunnel, and that's why this failure is categorized as *probable*. Moreover, the limited field of view of sensors at the end of a row when a human is approaching laterally means that this failure is considered as *critical* (because of the possibility of harm in this situation) for **O-L2** and **O-S1** operations making it a potential collision risk of level II. However, a higher risk level is expected during operation **O-U1** where the severity of the human injuries is categorized as *critical*.

3.2.6 Robot fails to predict human motion or intentions (F-G6)

The prediction of the future trajectory of humans is an essential component to plan safe and efficient movements of the robot. The accuracy of trajectory prediction is very important from a safety perspective to keep the robot away from the human within a safe distance. Moreover, in the case of planned interactions during load/unload of trays the human motion prediction is especially useful to determine when to start approaching the worker and when to stop. From a safety perspective, the robot's directive should be to not move if the human is tended to approach the robot. Thus, during unplanned interactions at footpaths, if the robot determines that the human tends to stay stationary, then the robot may perform evasion maneuvers to avoid unnecessary stops, which may affect the overall efficiency of the robot operations. However, if the robot is unable to correctly predict human trajectory (because of random human behavior or unreliable HDS readings), this may result in unnecessary stops or unwanted collisions. The severity of potential collisions depends on the speed of the robot during the close interactions. Since a robot in picking mode (**O-P1**) is allowed a maximum speed of less than 0.1 m, then the severity of the collision is assumed to be *marginal*. On the other hand, during robot operations **O-L1** and **O-G1**, the robot can move up to 0.5 m/s inside the polytunnels and 1 m/s (according to **A-LP5**), meaning the potential human injuries can be treated as *critical*. Due to the complexity of developing a trustworthy human prediction system, the occurrence of a failure in prediction was set as *occasional* which mostly depends on the randomness of human behavior and the input information given by the HDS.

3.2.7 Robot fails to accurately track human position (F-G7)

It is possible that the HDS fails in certain scenarios due to the complexity of distinguishing between environmental features and human features. In this context, false positive or false negative readings can provoke a mismatch between the actual human position with the position perceived by the robot. Thus, accurate human position estimation is a mandatory requirement during close interactions which require the robot to maintain a safety distance of 1.2m from the workers. Moreover, once a human is detected, the robot starts tracking this person in order to use the historical motion data as an input to the human prediction system mentioned in failure **F-G6**, to infer if the human is going to keep stationary or if they intend to move. In other words, this failure overlaps the operations **O-G1** and **O-L1** which were also affected by the failure **F-G6**. Thus, as was explained for **F-G6**, the severity of potential human injuries is categorized as *critical* with an occurrence described as *occasional*.

3.2.8 Robot fails to distinguish whether human detected was the one who summoned it (F-L1)

This failure is derived from the operation **O-L1** when the robot is in *logistics mode* transporting empty trays to the picker who summoned it. Let's assume the hypothetical scenario where more than one picker is working in the same row, or both are working in consecutive rows at the same time. In these scenarios, the HDS detects both pickers and somehow infer which of them was the one who summoned it. Once the correct picker is chosen, the robot tracks them in order to keep a safe distance of 1.2m. The problem in this situation is that the inference of which human is the one who summoned the robot is not an easy task, and can not be solved by using only LiDAR-based approaches. Complimentary hardware should be incorporated into the robot perception system in order to perceive messages from human workers. These messages help the robot to decide which human it should approach in order to avoid waste time waiting for actions from the picker who did not call the robot. The messages can be sent verbally if the robot has the capability of understanding specific words. Another alternative is to use non-verbal communication such as hand gestures where the robot must have the capability to interpret their meaning. According to the Table. 3 this failure was assigned a *critical* severity category since the loss of efficiency during pauses is not the only consequence. The incorrect selection of picker may drive to collision risk since the incorrect human picker was not aware of the robot's intentions to approach him, so, if the robot approach from behind the picker, any sudden movement may provoke a collision.

3.2.9 Robot fails to detect the human more than 7m away (F-U1)

This is a failure derived from operation **O-U1**. After the analysis of the previous failures modes, it is clear that the HDS is the most important component of the robot safety system for all of the four scenarios. In the case of close planned interaction during harvesting operations, the HDS should ensure an accurate human posture estimation to avoid getting closer than the minimum safety distance allowed. However, for UV-C treatment scenario, the accuracy in posture estimation is not as important as reliability in terms of false negatives and false positives for detection. In other words, it is preferred than the robot can detect a human at a long distance than to accurately track them at a shorter distance. The latter is because if the human is detected more 7m away, it is possible to avoid human injuries by stopping operation immediately, however if the robot is only able to detect people when they are already within 7m, the human will probably be injured. The severity of this failure was considered as *critical* and according to the current performance of our LiDAR based HDS, the occurrence of a false negative detection above 7m is *occasional*.

3.2.10 Robot fails to detect when a human is 7m away in a timely manner (F-U2)

This failure follows the line of the failure **F-U1**. In this case the robot was not able to detect the presence of a human that is close to the minimum distance away in time, and so is only aware of human presence when the human is already within the dangerous distance from the robot. The consequences of this failure are *critical* but unlike the failure **F-U1**, the occurrence is considered as *remote*, at least for the situations when the human is approaching frontally to the robot, based on the performance of the HDS . The situation when a human is approaching laterally at the end of the rows are not considered for this failure, since these situations were already covered in failure **F-G5**.

3.3 Safety Requirements

According to the descriptions of the agricultural scenarios given above, the safety system implemented in agricultural robots such as the Thorvald II should include at least the following components:

SR-1 Audiovisual Feedback Alert System (AVFAS): The robots must be equipped with colored beacons and speakers to periodically or explicitly inform people of the current robot's behavior/intentions [1].

SR-2 Human Detection System (HDS): The robots must be equipped with sensors such as LiDARs, RBG-D Cameras or IR Cameras that allow it to detect people in a range above 7m (necessary in the UV-C treatment) [17].

SR-3 Human Tracking and Motion Inference System (HTMIS): Based on the HDS, the robot must be able to keep in track the position of the person detected by the HDS and use this information to try to infer the current person movements [29].

SR-4 Human Action Recognition System (HARS): By using the information taken from the HDS, the robot must be able to recognize certain hand gestures (used to control the robot) as well as common actions that pickers perform during harvesting operations (see [31, 30].

SR-5 Collision Avoidance System (CAS): The robot must be able to perform human aware maneuvers while evading unexpected people (for example where the robot crosses footpaths) and while approaching a worker to unload/load trays [16, 11].

SR-6 Safety Contact System (SCS): In case of imminent contact (either by robot failures or human mistakes), the severity of the human injuries can be reduced by implementing a sensorized flexible skin (such as the one presented in [7]) that could be mounted on strategic areas on the robot structure to detect pressure changes and stop the robot immediately.

Based on these six components, the following safety policies are proposed with the aim of reducing the severity and/or occurrence of the failures identified in Table 3.

SP-1 The AVFAS must activate visual alerts with the aim of make the people who were planning to approach the robot be aware of the potential danger without the need to stop robot operations. Moreover, the AVFAS can also activate auditory alerts with explicit prerecorded voice messages to make the robot's intentions easier to understand for untrained people.

SP-2 The AVFAS must activate audiovisual alerts periodically to warn people of the robot's current intentions, especially useful when the robot is going to perform rows transitions inside the polytunnels. (When transitioning between rows the robot moves sideways, and, given the position of the sensors, moves blind.)

SP-3 The AVFAS must activate visual alerts when a human is detected within the public zone ($d > 3.6m$) in order to make them aware of the robot presence and potential danger. Moreover, the robot must activate audiovisual alerts when a human is within the social zone ($1.2m \leq d < 3.6m$) to ask if they need the robot's service (in case of operations inside polytunnels) or to inform them that the robot yields priority to the human by default (in case of operations outside polytunnels). According to the replying hand gesture, the robot has to pause the operation, re-plan another route, or approach the human.

SP-4 The CAS can be used to slowly approach a picker to load/unload trays only if the HTMIS infers that the picker is mostly stationary or if the HARS recognizes a specific hand gesture indicating it.

SP-5 The CAS can be used to perform safety evasion maneuvers outside the polytunnels only if the maneuverability space is wide enough and the HTMIS infers that the human detected is going to be stationary.

SP-6 A hand gesture with the directive of pause/continue robot operations can be recognized by the HARS at any moment, inside or outside the polytunnels.

SP-7 A safety stop can be automatically triggered by the HDS when a human is detected within an unsafe distance from the robot ($d \leq 7m$ for UV-C treatment, or $d \leq 1.2m$ for the remaining scenarios) or by SCS if a physical contact is detected.

SP-8 The farm workers and visitors must receive a basic training about robot operation modes, the meaning of visual and auditory alerts, and the potential hazards of approaching them (specially critical for UV-C treatment scenario).

3.4 Safety system architecture

Based on the safety requirements defined in Subsection 3.3, this subsection presents the overall architecture of the proposed safety system designed to address a safe HRI in the four scenarios studied in the MeSAPRO project. The proposed architecture is made up of three connected layers where, each layer is designed to address safety interaction at a different level, and higher layers aim to reduce the activation of lower layers. The higher layer aims to prepare a hassle free route and keep it up to date with changes in environment. In case of unexpected HRIs, the middle layer corresponds to intelligent HRI with the aim of keeping the distance between them within a safety range. Finally, the lower layer is utilised only in emergency scenarios where the upper layers do not respond properly to a specific situation.

3.4.1 Third layer: Efficiency aware planning

Layer 3 is the most upper layer in this architecture which introduces the efficiency aware planning. This layer is relevant only in scenarios in which the robot requires to share the workspace with the human workers, i.e. picking and logistics. In this scenarios, the efficiency aware planning is responsible to generate routes which involved only the necessary HRIs to minimize unnecessary stops or rerouting. This layer relies on global information of the environment as the expected location of every worker inside the polytunnel using GNNS based approaches [20], camera based approaches [27], model prediction based approaches or any other solution which uses human data extracted from external sources. By having global information of the workers locations it is possible to improve the coordination within the fleet of robots and thus optimize the scheduling of departure times (minimizing non-productive times), and creating routes which avoid stops due to unnecessary interactions with a worker who did not require the robot support. However, since only pickers are expected to have a GNSS tracking device, there is still a chance to have unnecessary interactions with other farm workers or untrained people.

3.4.2 Second layer: Navigation intent communication and planning

In case of both planned and unplanned HRI, the robot should be able to communicate with the human and decide what action to perform in order to ensure a safe and efficient interaction. With this aim, the second layer uses the information delivered by the on-board sensors to infer the human actions/intentions and decide what the actions the robot should take according to the safety requirements defined in Subsection 3.3. Unlike, the layer 3, this layer is expected to be used in the four agricultural scenarios, however not all the methods used in this layer are necessary or useful for all the scenarios. To explain how all the methods used in the second layer works, they will be divided into two groups:

Robot-to-human communication methods: To make the human aware of the robot current behaviour and future intentions, the following tree methods are proposed:

- **Robot back-off movement:** In operations which allow the presence of human at work space, if the robot is within a human social space, the robot should or not yield priority to humans (it depends on the robot operation). To communicate with human that the robot is yielding the priority, the robot performs a back-off slow movement before stop. In case the robot has priority, the robot motion will continue as planned. The backward movement directive is only applied in specific situations which include the planned interactions to load/unload trays at the collection point or when a picker is going to place fruit on the robot during picking operations.
- **Visual alerts:** In case the robot back-off movement is not clearly interpreted by the human, then the robot is equipped with beacons which uses different colors to indicate different robot behaviours and also indicate the level of risk based on the distance between the human and the robot.
- **Audio alerts:** To complement the robot movement and visual alerts, (especially useful in case of interaction with non-trained people), the robot is equipped with speakers that reproduce prerecorded voice messages that explicitly indicate robot intentions and warn human about danger. Specific voice messages can be applied in UV-C treatment operations to warn human about UV-C light danger, or in any operation inside the polytunnels to warn any occluded human at the end of rows that the robot is going to leave a row. Moreover, in operations which are time critical and it is require to minimize the waste of time during pauses, the voice messages may produce a more efficient and fluid close interaction. Note that the audiovisual alerts can be triggered by a human detection or can be activated periodically to make any human (probably distracted) aware of the robot presence, intentions and potential danger. The visual alerts can be seen from a larger distance than the voice messages can be hear (limited by the speaker output power). However the audio alerts explicitly communicate the message whiles the visual alerts may be misunderstood by non trained workers.

Human-to-robot communication methods: A Non-verbal communication method such as using hand gestures can provide valuable information that allow humans to interact with robots in an intuitive manner, especially in planned interactions for load/unload trays. Thus, to make the robot aware of the human intentions, and allow a bilateral communication between them, the following human gesture recognition methodology is proposed:

- **Human gestures recognition:** While a robot is navigating along the rows inside the polytunnels, it will probably interact closely with multiple human workers, thus the robot should be able to infer if the worker detected requires its assistance or not. The latter can be done following the proposal presented in [31], where pickers give directives to a robot based on hand or hand gestures. The directives may include: *move towards me*, *return to collection point*, *stop*, *continue operations*, and *re-routing*.

  During planned close interactions, the robot can reproduce pre-recorded voice messages which ask human worker for directives. The worker can reply this messages by performing for instance the gestures corresponding to *move towards me* or *return to collection pont*. In this way the robot first ask, and the human reply. However, the communication is not necessary always in this way, for instance the gesture for *stop* aims to be used in any moment without requiring a robot request first. Additionally, not only hand gestures can be recognized, but actions can also be incorporated into the recognition model to be able to better infer what the human is currently doing or intended to do.

3.4.3 First layer: Safety Stops

The first layer of the architecture is the last line of defense in the safety system. It is triggered when an unexpected physical contact is detected or when an imminent collision is likely to happen due to late human detection. Thus,

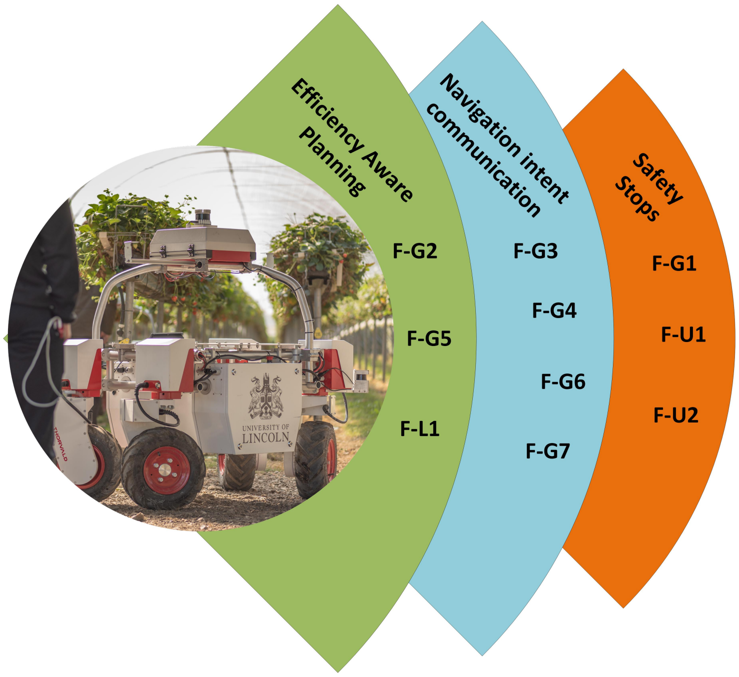


Figure 3: The layered safety system architecture corresponding to failures

the only solution to minimize human injuries is to stop the operations as soon as these anomalies are detected. The safety stop can be triggered by two subsystems:

- **Safety stop triggered by LiDAR/camera readings:** The first line of action to stop the robot motion in case of imminent collision is to use LiDAR/camera readings. Independent of the robot operation, if the robot detects a human within 1.2m, the directive is to immediately stop in order to reduce the probability of colliding with the human. Additionally a safety stop is triggered if the human is detected within 7m only for the UV-C treatment scenario.

- **Safety stop triggered by soft sensor readings:** As backup, the robot must be equipped with sensors to enhance safety and intuitiveness of physical HRIs for both intentional and unintentional contacts. In case of imminent contact (either by robot failures or human mistakes) the reading of this sensors can be used to stop the robot immediately. The human injuries are not expected to be completely removed by applying this safety stop, but at least the severity of these can be reduced. To implement this kind of safety systems, a possible approach is to use sensorised flexible skin (as the one presented in [7]) that could be mounted on strategic areas on the robot structure to detect pressure changes. However, since the impact can happen on an uncovered part of the robot, an alternative approach can be the use of force sensors attached to the robot actuators (i.e. the wheels motors) in order to detect any change in torque signal. A combination of these solutions can minimise the probability of having false positives or false negatives detection.

The Figure 3 represents the failures from Table 3 and the corresponding layer from safety architecture that will be handling the failure.

4 Formal verification

In order to verify the satisfaction of safety requirements of the agricultural scenarios studied in this work, each component of the proposed safety system were modeled as discrete-time Markov decision process (MDP) [24]. MDPs are a widely used formalism to model sequential decision-making problems where there is inherent uncertainty about the system's evolution. The use of MDPs makes it possible for us to use a probabilistic model-checking tool such as PRISM [21] which already has support for solving MDPs against properties in Linear Temporal Logic (LTL), in particular allowing for the maximisation of the probability of satisfying an LTL formula. This, in turn, means that the model of system behaviour, in the form of an MDP, can be checked against conditions, expressed in LTL, that correspond to states that we wish to achieve (such as "the robot stops when the human makes the appropriate signal") or wish to avoid (such as "the robot collides with the human"). This section first introduces the assumptions used to construct the proposed HRI model, then the discrete states that define the MDP representation of the proposed model are shown, and finally, a case study is presented to illustrate how to implement each component of the proposed model using the PRISM modeling language.

4.1 Modelling

4.1.1 Defining the robot agricultural tasks as sequence of steps

To determine whether the tasks within the agricultural scenarios described in Subsection 2.1 are completed or not, they can be divided into a sequence of steps. The robot finishes a specific agricultural task when it has successfully completed all the steps. For instance, the sequence of steps to complete the UV-C treatment scenario corresponds to a closed loop in which the robot starts operation from its home location at the storage shed, the robot moves from the shed to the polytunnel, the robot performs the UV-C treatment, the robot moves back to the shed, and finally, the robot is back at the shed. Thus, as it can be seen in Fig. 4, for every agricultural scenario studied here, the sequence of steps corresponds to a closed loop that starts and finish with the robot at the shed. The number of steps and complexity depends on the level of HRI expected on each scenario. For instance, in the case of performing the logistics tasks, the corresponding sequence of steps is more complex than for UV-C treatment (where it is not expected to have HRI) since now it is possible to interact with human workers on footpaths between the shed and the polytunnels or with pickers inside polytunnels. Additionally, as can be seen in Fig. 4 (highlighted in red), in order to determine when a robot has completed a specific step within the sequence, there are conditions to be satisfied such as traversing a specific distance or completing the harvesting of a certain amount of fruit placed on trays. Thus since a planned/unplanned HRI may happen each time the robot traverses a new polytunnel row or footpath segment, then the inclusion of these conditions exponentially increases the chances of having human injuries.

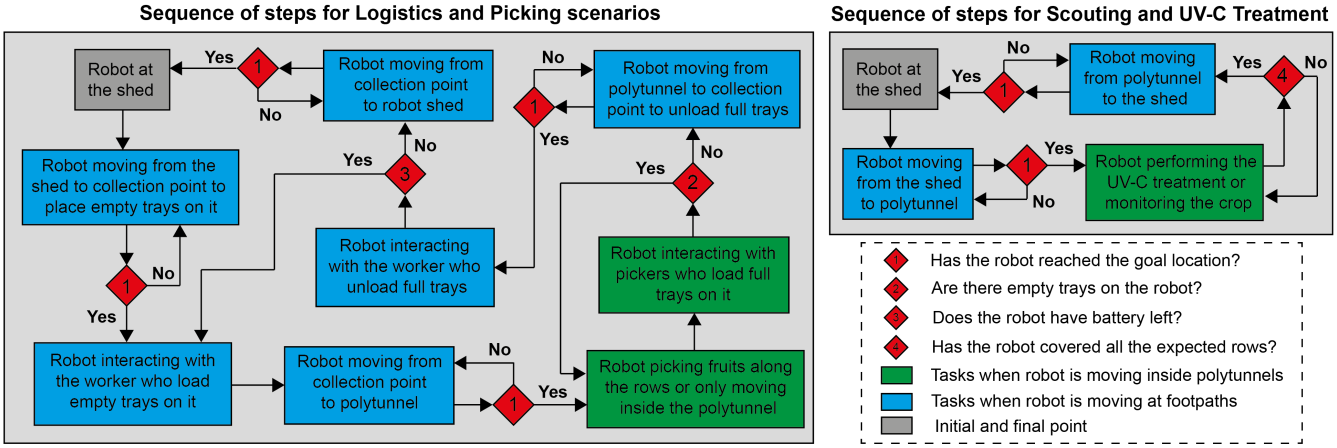

Sequence of steps for Logistics and Picking scenarios
Robot at the shed
Yes
1
No
Robot moving from collection point to robot shed
No
1
Yes
Robot moving from polytunnel to collection point to unload full trays
Robot moving from the shed to collection point to place empty trays on it
Yes
3
No
Robot interacting with the worker who unload full trays
Yes
2
No
Robot interacting with pickers who load full trays on it
1
No
Yes
Robot interacting with the worker who load empty trays on it
Robot moving from collection point to polytunnel
No
1
Yes
Robot picking fruits along the rows or only moving inside the polytunnel
Sequence of steps for Scouting and UV-C Treatment
Robot at the shed
Yes
No
1
Robot moving from polytunnel to the shed
Yes
4
No
Robot moving from the shed to polytunnel
1
Yes
No
Robot performing the UV-C treatment or monitoring the crop
1 Has the robot reached the goal location?
2 Are there empty trays on the robot?
3 Does the robot have battery left?
4 Has the robot covered all the expected rows?
🟩 Tasks when robot is moving inside polytunnels
🟦 Tasks when robot is moving at footpaths
⬜ Initial and final point

Figure 4: Diagrams with the sequence of steps which define all the agricultural scenarios.

4.1.2 Robot operation modes

In order to successfully complete each of the sequential steps defined above, the robot must have the ability to operate in different modes and with different levels of autonomy to adapt to possible unexpected situations. Therefore, we describe the robot motion during the above-mentioned scenarios by 11 different modes divided into 3 groups:

- **Standard operation modes:** The operation modes in common for all agricultural scenarios include: 1) Robot in pause mode, 2) Robot moving along footpaths, 3) Robot performing a transition between footpath segments, 4) Robot performing a transition between rows inside the polytunnel.
- **Custom operation modes:** In order to perform certain agricultural tasks, the robot must be configured in different ways (see Figure 2) and move at different speeds. Thus, while operating inside the polytunnel, the robot can be working in three custom modes: 5) Robot moving along a row transporting trays (normal speed), 6) Robot moving along a row while picking fruit (slow motion) 7) Robot moving along a row performing UV-C treatment (same configuration used for scouting).

- **Human-aware operation modes:** The implementation of the safety system components described in Subsection 3.3 aims to ensure a safe HRI during planned and unplanned interactions. Thus, the operation modes which make use of these safety components include: 8) Robot stopping the current action because of human detected within a dangerous distance or because a collision was detected, 9) Robot approaching the human worker's position (reducing the speed) to load/unload trays, 10) Robot moving away from the human's position after loading/unloading trays, 11) Robot performing evasive maneuvers on a footpath.

4.1.3 Human decision-making

In order to emulate human decision-making during planned and unplanned HRI, the following assumptions were considered according to the human's training level:

- If the HRI happens with the picker who summoned the robot, then the picker starts the interaction being stationary, but they can decide to stay stationary or move towards the robot if they are within the social zone.
- If the HRI happens with a picker who didn't call the robot, then they start the interaction being stationary, but they can decide to approach or move away when the robot is within the social zone.
- Untrained people inside polytunnels and workers at the end of the rows are always moving to the robot position until they become aware of the danger. If they become aware of danger, then they move away.
- Workers and untrained people on footpaths move towards the robot until they become aware of the danger. If they become aware of danger, then they can decide to stop or move away.
- If workers or untrained people decide to remain stationary before the robot enters their social zone, then the robot starts evading them and the human can decide to keep stationary or walk next to the robot.
- Only workers can perform hand gestures, and this can happen only when the robot is within their social zone and the workers are aware of the robot presence.
- If a person (independent of the training level) is injured, or if the robot performs a safety stop, then the person becomes aware of the danger, and decides to move away from the robot.
- Workers who are supposed to place trays on the robot (inside and outside polytunnel) are assumed to always be aware of the danger of HRI.
- An untrained person who is going to interact with a robot can become aware of the danger only if he/she is able to interpret the audiovisual alerts activated by the AVFAS. If he/she is not able to interpret the alerts, then he/she decides to continue approaching till reaches an unsafe distance.

4.1.4 Hazard situations

Considering all the situations and failure modes presented in Table 3, we can conclude that there are basically five ways to produce human injuries:

HI-1 Actual human injuries when $d \leq 7$m and the robot continues to carry out UV-C treatment in the polytunnels (UV-C treatment scenario).

HI-2 Actual human injuries when the human collides with the robot but the robot stops just after the contact (in any agricultural scenario).

HI-3 Actual human injuries when the human collides with the robot and it is still moving after contact (in any agricultural scenario).

HI-4 Potential human injuries when $1.2\text{m} \leq d < 3.6\text{m}$ (social zone) and the robot is not aware of the human's intentions during logistics and picking operations (logistics and picking scenarios).

HI-5 Potential human injuries when $1.2\text{m} \leq d < 3.6\text{m}$ (social zone) and the human is not aware of the presence of the robot (in any agricultural scenario).

HI-2 and **HI-3** are hazards that represent real physical human injuries where the severity level depends on the robot speed during and after the collision. The remaining hazards represents only potential or virtual injuries. In the case of **HI-1**, it is a virtual injury but may produce real injuries if the human is explosed to UV-C radiation at an unsafe distance for a long enough period of time [25]. In the case of **HI-5**, since the robot is getting close to the intimate zone, any failure in the HDS, or the HTMIS can produce delays in the activation of safety stops which subsequently may lead to **HI-2** or **HI-3**. In the same way with **HI-4**, if HARS fails to interpret a hand gesture performed by a trained worker located next to the robot, an incorrect action by the robot may lead to **HI-2** or **HI-3**.

As was initially mentioned in Subsection 2.1 and later was shown in Table 3, the hazards identified for the scouting scenario overlap with the hazards covered by the rest of the scenarios. Thus, to simplify the modelling and human injury assessment, the following sections will cover only the first three scenarios described in Subsection 2.1.

4.2 Implementation in PRISM language

4.2.1 The PRISM language

As presented in [22], instead of using a traditional flat representation of a MDP, we can use an equivalent factored representation, where the state is decomposed into relevant state features, and the transition function is encoded over such features. In this work, we decided to take advantage of such a factored representation, using a probabilistic STRIPS-like representation of factored MDPs, based on the PRISM modeling language.

The fundamental components of the PRISM language are modules, variables, and constants. A model is composed of a number of modules which can interact with each other. A module contains a number of local variables. The values of these variables at any given time constitute the state of the module. The global state of the whole model is determined by the local state of all modules. The behaviour of each module is described by a set of commands. A command takes the form:

$$[\mathtt{action}]\ (\mathtt{guard\_1} \mid \mathtt{guard\_2})\ \&\ ...\ \&\ \mathtt{guard\_m} - > \ \mathtt{prob\_1} : \ \mathtt{update\_1} + ... + \mathtt{prob\_n} : \mathtt{update\_n} \qquad (1)$$

The $\mathtt{m\text{-}th}$ guard is a predicate over all the variables in the model (including those belonging to other modules) which may be related with other guards by logical operators such as AND (&), OR (|), Not Equal (!=), among others. Each $\mathtt{n\text{-}th}$ update describes a transition which the module can make if all the guards are true. A transition is specified by giving the new values of the variables in the module, possibly as a function of other variables or constants. Each update is assigned a probability (or in some cases a rate) which will be assigned to the corresponding transition. The command also optionally includes an action, either just to annotate it, or for synchronisation.

With this base, the following subsections aim to introduce the variables and probabilities (constants) needed to construct commands as the one in (1) which represents the state transitions of the proposed HRI model.

4.2.2 Modelling the agricultural tasks

As was mentioned in Subsection 4.1.1, the agricultural tasks are divided into a sequence of steps. In order to define in which step of each scenario we are, the variables *x_uvc*, *x_logistics* and *x_picking* were introduced. Table 10 in the Appendix lists and describes the 4 discrete values that variables *x_uvc* can take as well as the possible transitions between them. On the other hand, Table 10 lists 9 values that the variable *x_logistic* can take where 8 of them overlap with the values that the variable *x_picking* can take. For these three variables related to the agricultural tasks, the transition between two different values is always deterministic, thus it is not necessary to introduce any probabilistic terms to model the transitions. However, it is necessary to introduce some auxiliary variables to determine when an agricultural step has been completed and a transition can be carried out. These auxiliary variables are also listed in Table 10 and include: the number of full trays that the robot is carrying represented by *x_trays*; the number of times the robot performed two way trips from polytunnel to collection point represented by *x_runs*; and the number of rows and footpaths that have been traversed by the robot represented by *x_rows* and *x_segments* respectively.

The range of values that the auxiliary variables can take is constrained by the constants described in Table 11, which basically define the scale of the field to be covered and the capacity of the robot to transport full trays.

4.2.3 Modelling the robot operations

Depending on the agricultural scenario, the robot is expected to perform specific operation modes that may or may not overlap with another scenario. Thus, based on the robot operation modes introduced in Subsection 4.1.2, the robot operation is defined by the variable *x_robot* which can take 11 values listed in Table 12. Similar to the variables related to the agricultural tasks presented in Table 10, the variable *x_robot* is purely deterministic, thus, it does not depend on any probabilistic term to make transitions between two values.

4.2.4 Modelling the safety system components

According to the safety requirements presented in Subsection 3.3, the proposed safety system is made up of 6 main components which include: AVFAS, HDS, HTMIS, HARS, CAS, and SCS. To implement these components in PRISM, their behavior has been characterized by 6 variables and 16 constants that are used to define the probability of success/failure (i.e. the reliability) of each component.

Table 13 summarizes the values that the 6 variables *x_hds*, *x_htmis*, *x_hars*, *x_scs*, *x_visual* and *x_voice* can take, highlighting with * the transitions that are non-deterministic. Table 14 shows a list of constants that are introduced to define the probabilities to make non-deterministic transitions. These constants can take values between 0 and 1 depending on the actual effectiveness of the safety system implemented. For instance, if the HDS is considered virtually perfect to detect a human above 7m, then, the constants *x_hds_fail_1* and *x_hds_fail_5* are set as 0 while they are set as 1 if 100% of times the HDS fails to detect a human on time. Notice that the performance of HDS in terms of failures may be different inside the polytunnel than at footpaths due to the different levels of occlusion on each environment. Thus, Table 14 shows independent constants for define the probability of failure in each environment.

It is important to notice that in Tables 13-14 no variables or constants were introduced to implicitly define the CAS, but its behavior was included when *x_robot*=3,8 and the success of the collision-free maneuvering depends indirectly on the success of the remaining safety system components.

Finally, the behavior of the AVFAS was defined by introducing two independent variables for audio and visual alerts. According to Table 13, the transitions of these variables are mostly deterministic, but it is because the non-deterministic behavior of their potential failures are modelled as part of the human side instead of the AVFAS itself.

4.2.5 Modelling human behaviour

For PRISM implementation purposes, human behavior is described by 5 variables for which the possible values are listed in Table 15. The first variable is denoted by *x_human* and determines if the human decides to interact with the robot or not and captures what the human's training level is (trained or untrained). The second variable denoted by *x_motion* represents the most relevant actions that trained farm workers typically perform during harvesting operations and the expected actions from untrained people whom approach the robot without being aware of the danger. The third variable is denoted by *x_aware* and represents the human knowledge about the robot intentions or potential danger. The fourth variable denoted by *x_gesture* determines if the human performs or not a hand gesture to make the robot knows about his/her intentions. Finally, the fifth variable denoted by *x_dist* represents the distance between the human and the robot in discrete steps according to the human personal spaces. In order to introduce non-deterministic human behavior, 5 constants are used to define the probability of making a decision or another (see Section 4.1.3). Table 16 summarizes the list of probabilities used that include the probabilities that unplanned HRI are going to happen, the probabilities that the human gets aware of the robot presence or potential danger, the probability that the human decided to perform a risky movement during close HRI, and the probability that a trained worker performs a hand gesture to communicate him/her intentions to the robot.

It is important to notice that the transition probabilities may vary according to the training level of the human interacting with the robot. This can be seen in Table 16 where it was used different constants to characterize the behaviour of untrained visitants and trained farm workers.

4.2.6 Modelling the hazard situations

Finally, since the values of the variables at a given time constitute the state of the whole HRI, then the PRISM model checker tool should be able to determine if a potential hazard situation is happening or not when a failure is introduced into the robot safety system. The failure modes to be evaluated in PRISM were identified previously in Table 3. Moreover, in Subsection 4.1.4, it was identified the hazard scenarios which may lead to actual human injuries, remarking **HI-1**, **HI-2** and **HI-3**. These three possible human injuries can be evaluated in PRISM by introducing the following conditions:

HI-1 Pmax=? [F task_finished=false & (x_dist>=2 & x_robot!=10 & x_uvc=2)].

HI-2 Pmax=? [F task_finished=false & (x_dist=5 & x_scs=1)].

HI-3 Pmax=? [F task_finished=false & (x_dist=5 & x_scs=2)].

where Pmax=? [F<t task_finished=false (<condition>)] is the notation of LTL properties to be evaluated in PRISM. This particular notation determines the maximum probability of satisfying a specific condition before the simulated agricultural process ended. The auxiliary variable *task_finished* becomes true when the variables *x_uvc*, *x_logistics*, and *x_picking* have taken all their possible values and return to the initial value 0.

4.3 PRISM implementation: UV-C treatment case study

In Subsection 4.2, all the variables, possible values, and transition probabilities were introduced in order to model the agricultural scenarios defined in Subsection 2.1. The aim of this Subsection is to illustrate with an example how to use these elements to construct a simplified UV-C treatment model in PRISM by using commands of the form (1).

Since this example is only for illustration purposes, a simplified version of the UV-C treatment scenario is presented here, where only the robot operations inside the polytunnels are modeled, i.e when *x_uvc*=2. The complete model with all the transitions presented in Tables 10, 12, 13, and 15 will be used in Section 4.4.

4.3.1 Commands to model the agricultural task

Since only operations inside the polytunnel are considered, then the variable *x_uvc* does not need to be updated. However, the auxiliary variable *x_rows* needs to be updated each time the robot covers a row. Thus, in order to cover the possible transitions for the variable *x_rows*, the following command is needed:

$$(guard\_1 \,\&\, x\_rows < N\_rows) - > \ (x\_rows' = x\_rows + 1); \tag{2}$$

where the guard expression is defined by:

$$guard\_1 = (x\_uvc = 2 \,\&\, x\_human = 0 \,\&\, x\_robot = 8) \tag{3}$$

The command in (2) updates *x_rows* starting from 0 and increasing in steps of 1 till reach *x_rows*=*N_rows* which represents the condition when the robot has covered all the rows that are expected to be treated. To better understand how the command (2) is executed in PRISM, Fig 5 illustrates the transition between states, where the number inside the circles are the possible values of variable *x_rows* from 0 to *N_rows*, and the numbers in red represent the probabilities of transition if the condition *guard_1* is satisfied.

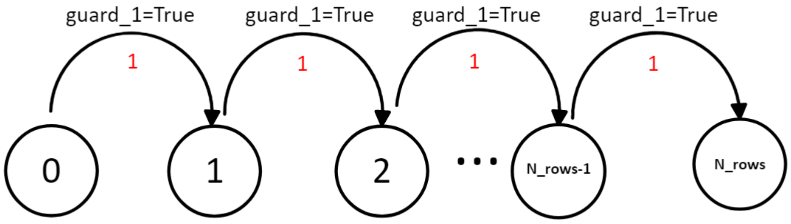


Figure 5: Diagram with the states transition for variable *x_rows*.

4.3.2 Commands to model the robot operation

In order to make all possible transitions for the variable *x_robot* during the UV-C treatment operation, the following commands are needed:

$$(guard\_2 \,\&\, x\_robot = 7) - > \ (x\_robot' = 4); \tag{4}$$

$$(guard\_2 \,\&\, (x\_robot = 4 \mid x\_robot = 10)) - > \ (x\_robot' = 7); \tag{5}$$

$$(guard\_3 \,\&\, x\_robot! = 10) - > \ (x\_robot' = 10) \,\&\, (x\_aware' = 1); \tag{6}$$

where the guards are defined by:

$$guard\_2 = (x\_uvc = 2 \;\&\; x\_human = 0) \tag{7}$$
$$guard\_3 = (x\_uvc = 2 \;\&\; x\_human! = 0 \;\&\; x\_hds = 2) \tag{8}$$

Commands (4)-(5) aim to update the value of *x_robot* between moving along a row and making transition between rows when no human presence is detected. The command in (6) activates a safety stop (i.e. stop motion and turning off the UV-C light) in case a human is detected within the range 3.6m$\leq$ d $\leq$ 7m. This command also updates the variable *x_aware* to make the human aware of the danger. To reactivate the robot operation after a safety stop, the condition (*guard_2* & *x_robot*=10) has to be satisfied in command (5) which makes the robot resume the operation from a transition between rows. Figure 6(a) illustrates the execution of commands (4)-(6) where the color used on each arrow indicates which condition/guard has to be satisfied to make a transition.

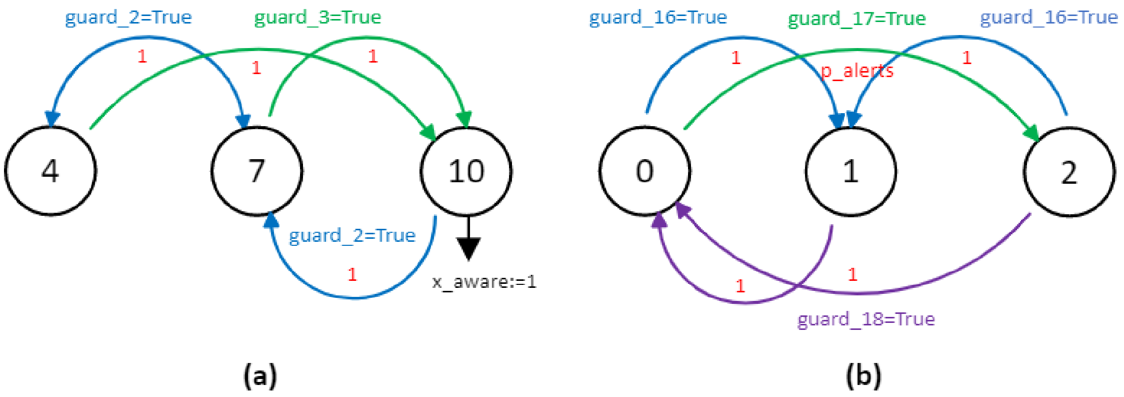


Figure 6: Diagrams with the states transition for variables: (a) *x_robot*, considering only operations inside the polytunnels during UV-C treatment (b) *x_voice* and *x_visual*.

4.3.3 Commands to model the safety system

To simplify this example, the safety system during the UV-C treatment only includes the AVFAS and HDS. Thus, in order to make transitions for the variable *x_hds* when the robot is inside the polytunnels, the following commands need to be implemented:

$$(guard\_4 \;\&\; x\_hds = 0) - > 1 - p\_hds\_1 : (x\_hds' = 1) + p\_hds\_1 : (x\_hds' = x\_hds); \tag{9}$$
$$(guard\_5 \;\&\; x\_hds = 0) - > 1 - p\_hds\_5 : (x\_hds' = 1) + p\_hds\_5 : (x\_hds' = x\_hds); \tag{10}$$
$$(guard\_6 \;\&\; x\_hds <= 1) - > 1 - p\_hds\_2 : (x\_hds' = 2) + p\_hds\_2 : (x\_hds' = x\_hds); \tag{11}$$
$$(guard\_7 \;\&\; x\_hds <= 1) - > 1 - p\_hds\_6 : (x\_hds' = 2) + p\_hds\_6 : (x\_hds' = x\_hds); \tag{12}$$
$$(guard\_8 \;\&\; x\_hds <= 2) - > 1 - p\_hds\_3 : (x\_hds' = 3) + p\_hds\_3 : (x\_hds' = x\_hds); \tag{13}$$
$$(guard\_9 \;\&\; x\_hds <= 2) - > 1 - p\_hds\_7 : (x\_hds' = 3) + p\_hds\_7 : (x\_hds' = x\_hds); \tag{14}$$
$$(guard\_10 \;\&\; x\_hds <= 3) - > 1 - p\_hds\_4 : (x\_hds' = 4) + p\_hds\_4 : (x\_hds' = x\_hds); \tag{15}$$
$$(guard\_11 \;\&\; x\_hds <= 3) - > 1 - p\_hds\_8 : (x\_hds' = 4) + p\_hds\_8 : (x\_hds' = x\_hds); \tag{16}$$
$$(guard\_12 \;\&\; x\_hds = 4) - > (x\_hds' = 3); \tag{17}$$
$$(guard\_13 \;\&\; x\_hds = 3) - > (x\_hds' = 2); \tag{18}$$
$$(guard\_14 \;\&\; x\_hds = 2) - > (x\_hds' = 1); \tag{19}$$
$$(guard\_15 \;\&\; x\_hds = 1) - > (x\_hds' = 0); \tag{20}$$

where the guards are defined by:

$$guard\_4 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot! = 7 \,\&\, x\_dist = 1) \tag{21}$$
$$guard\_5 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot = 7 \,\&\, x\_dist = 1) \tag{22}$$
$$guard\_6 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot! = 7 \,\&\, x\_dist = 2) \tag{23}$$
$$guard\_7 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot = 7 \,\&\, x\_dist = 2) \tag{24}$$
$$guard\_8 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot! = 7 \,\&\, x\_dist = 3) \tag{25}$$
$$guard\_9 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot = 7 \,\&\, x\_dist = 3) \tag{26}$$
$$guard\_10 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot! = 7 \,\&\, x\_dist = 4) \tag{27}$$
$$guard\_11 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_robot = 7 \,\&\, x\_dist = 4) \tag{28}$$
$$guard\_12 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_dist = 3) \tag{29}$$
$$guard\_13 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_dist = 2) \tag{30}$$
$$guard\_14 = (x\_uvc = 2 \,\&\, x\_human! = 0 \,\&\, x\_dist = 1) \tag{31}$$
$$guard\_15 = (x\_uvc = 2 \,\&\, x\_human = 0 \,\&\, x\_dist = 0) \tag{32}$$

Commands in (9)-(16) cover the updates of *x_hds* from 0 to 4 which depend of the probabilities of failure defined in Table 14. On the other hand, commands in (17)-(20) cover the updates of *x_hds* from 4 to 0 which in this case are purely deterministic. Figure 7 illustrates the transitions of variable *x_hds*. However, for readability, the transitions whose conditions depend on *x_robot* = 7 are not illustrated on this diagram (i.e when the robot is making transitions between rows).

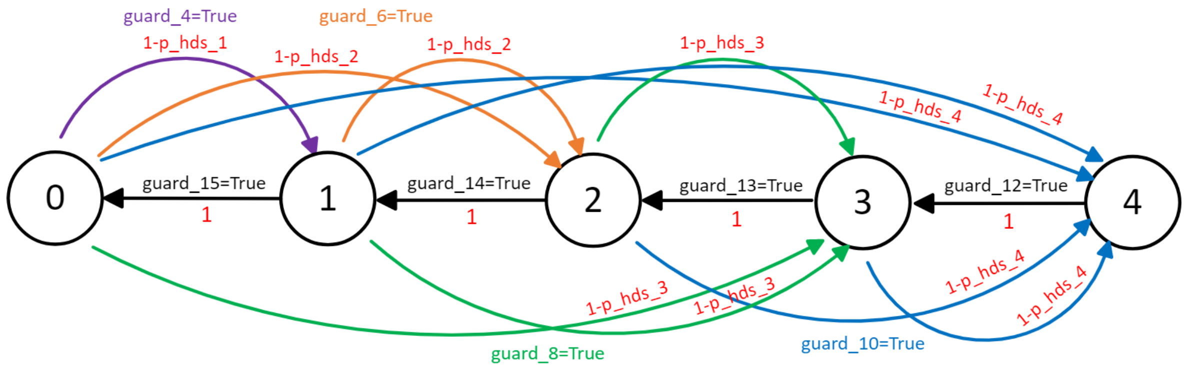


Figure 7: Diagram with the states transition for variable *x_hds*, without considering transitions when robot operation *x_robot*=7.

Then, in order to make the transitions for the two variables related with AVFAS, the following commands are needed:

$$(guard\_16 \,\&\, (x\_visual! = 1 \,|\, x\_voice! = 1)) -> (x\_visual' = 1)\&(x\_voice' = 1); \tag{33}$$
$$(guard\_17 \,\&\, x\_voice = 0) -> p\_alerts : (x\_voice' = 2) + 1 - p\_alerts : (x\_voice' = x\_voice); \tag{34}$$
$$(guard\_17 \,\&\, x\_visual = 0) -> p\_alerts : (x\_visual' = 2) + 1 - p\_alerts : (x\_visual' = x\_visual); \tag{35}$$
$$(guard\_18 \,\&\, (x\_visual! = 0 \,|\, x\_voice! = 0)) -> (x\_visual' = 0)\&(x\_voice' = 0); \tag{36}$$

where the guards are defined by:

$$guard\_16 = (x\_uvc = 2 \,\&\, x\_hds >= 1) \tag{37}$$
$$guard\_17 = (x\_uvc = 2\&(x\_robot = 7 \,|\, x\_robot = 1 \,|\, x\_robot = 4)) \tag{38}$$
$$guard\_18 = (x\_uvc = 2 \,\&\, x\_hds = 0) \tag{39}$$

Command (33) activates the audiovisual alerts (visual and voice alerts in parallel) when the HDS detects a human above 7m. Commands (34)-(35) can also activate the audiovisual alerts, but in this case without need to detect a human, this activation depends of the probability that at this specific moment a periodic alert was programmed to happen. In any case, after alerts were activated, they can be deactivated only if the condition in (36) is satisfied. The execution of commands (33)-(36) is illustrated in Fig. 6(b).

4.3.4 Commands to model the human behavior

To simplify this example, the human behavior considers only the actions performed by untrained people (since farm workers are aware of the UV-C danger). Thus, the transitions of the variable *x_gesture* were not included since only trained people is able to perform hand gestures.

In order to update the value of *x_human*, the following commands are required:

$$(\mathrm{guard\_19} \;\&\; \mathrm{x\_human} = 0) - > \mathrm{p\_int\_1} : (\mathrm{x\_human}' = 1) + 1 - \mathrm{p\_int\_1} : (\mathrm{x\_human}' = \mathrm{x\_human}); \tag{40}$$

$$(\mathrm{guard\_20} \;\&\; \mathrm{x\_human}! = 0) - > (\mathrm{x\_human}' = 0); \tag{41}$$

where the guards are defined by:

$$\mathrm{guard\_19} = (\mathrm{x\_uvc} = 2 \;\&\; \mathrm{x\_rows} < \mathrm{N\_rows}) \tag{42}$$

$$\mathrm{guard\_20} = (\mathrm{x\_uvc} = 2 \;\&\; \mathrm{x\_dist} = 0 \;\&\; \mathrm{x\_motion} = 2 \;\&\; \mathrm{x\_aware} = 1) \tag{43}$$

Command (40) is used to determine if an untrained person is going to interact with the robot inside the polytunnels or not. Once a HRI has happened, the variable *x_human* returns to the default value of 0 if the condition in command (41) is satisfied.

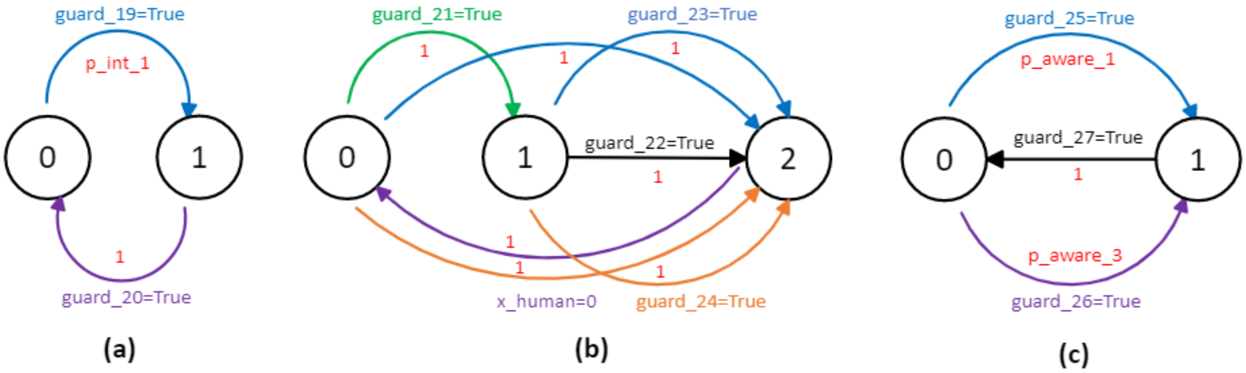


Figure 8: Diagrams with the states transition for variables: (a) x_human (b) x_motion and (c) x_aware.

Once we know that an untrained person is going to interact with a robot, the actions which govern the human motion are represented by the variable *x_motion*, which transitions are defined by the following commands:

$$(\mathrm{guard\_21} \;\&\; \mathrm{x\_motion} = 0) - > (\mathrm{x\_motion}' = 1); \tag{44}$$

$$(\mathrm{guard\_22}\&\mathrm{x\_motion} = 1) \,|\, (\mathrm{guard\_23}\&\mathrm{x\_motion}! = 2) \,|\, (\mathrm{guard\_24}\&\mathrm{x\_motion}! = 2) - > (\mathrm{x\_motion}' = 2); \tag{45}$$

$$(\mathrm{x\_human} = 0\& \;\mathrm{x\_motion} = 2) - > (\mathrm{x\_motion}' = 0); \tag{46}$$

where the guards are defined by:

$$\mathrm{guard\_21} = ((\mathrm{x\_robot} = 4 \,|\, \mathrm{x\_robot} = 7) \;\&\; (\mathrm{x\_human} = 1\&\mathrm{x\_dist}! = 5\&\mathrm{x\_aware} = 0)) \tag{47}$$

$$\mathrm{guard\_22} = ((\mathrm{x\_robot} = 4 \,|\, \mathrm{x\_robot} = 7) \;\&\; (\mathrm{x\_human} = 1 \;\&\; \mathrm{x\_aware} = 1)) \tag{48}$$

$$\mathrm{guard\_23} = (\mathrm{x\_dist} = 5 \;\&\; \mathrm{x\_human}! = 0) \tag{49}$$

$$\mathrm{guard\_24} = (\mathrm{x\_human}! = 0 \;\&\; \mathrm{x\_robot} = 10) \tag{50}$$

Then, to make the untrained people aware of danger of approaching the robot, the following commands are included to update *x_aware*:

$$(\mathrm{guard\_25} \;\&\; \mathrm{x\_aware} = 0) - > \mathrm{p\_aware\_1} : (\mathrm{x\_aware}' = 1) + 1 - \mathrm{p\_aware\_1} : (\mathrm{x\_aware}' = \mathrm{x\_aware}); \tag{51}$$

$$(\mathrm{guard\_26} \;\&\; \mathrm{x\_aware} = 0) - > \mathrm{p\_aware\_3} : (\mathrm{x\_aware}' = 1) + 1 - \mathrm{p\_aware\_3} : (\mathrm{x\_aware}' = \mathrm{x\_aware}); \tag{52}$$

$$(\mathrm{guard\_27} \;\&\; \mathrm{x\_aware} = 1) - > (\mathrm{x\_aware}' = 0); \tag{53}$$

where the guards are defined by:

$$guard\_25 = (x\_uvc = 2\&x\_robot! = 10\&x\_dist >= 1\&x\_visual >= 1\&x\_human = 1) \tag{54}$$

$$guard\_26 = (x\_uvc = 2\&x\_robot! = 10\&x\_dist >= 2\&x\_voice >= 1\&x\_human! = 0) \tag{55}$$

$$guard\_27 = (x\_uvc = 2\&x\_human = 0) \tag{56}$$

Commands (51)-(52) update the variable *x_aware* in case of any of the audiovisual alerts were activated at a specific *x_dist*. The variable *x_aware* returns to the default value of 0 by using the command (53).

Finally, to fully model human behavior in this case, the variable *x_dist* needs to be updated according to the relative distance between the human and the robot. The latter is done by introducing the following commands:

$$(guard\_28 \ \& \ x\_dist <= 3) - > (x\_dist' = x\_dist + 1); \tag{57}$$

$$(guard\_29 \ \& \ x\_dist <= 3) - > (x\_dist' = x\_dist + 1); \tag{58}$$

$$(guard\_30 \ \& \ x\_dist = 4) - > (x\_dist' = 5) \ \& \ (x\_aware' = 1); \tag{59}$$

$$(guard\_31 \ \& \ x\_dist >= 1) - > (x\_dist' = x\_dist - 1); \tag{60}$$

where the guards are defined by:

$$guard\_28 = (x\_human! = 0\&x\_motion = 1) \tag{61}$$

$$guard\_29 = (x\_human! = 0\&x\_motion = 0\&(x\_robot = 1|x\_robot = 2|x\_robot = 4)) \tag{62}$$

$$guard\_30 = (x\_robot! = 10\&x\_human! = 0) \tag{63}$$

$$guard\_31 = (x\_human! = 0\&x\_motion = 2) \tag{64}$$

Commands (57)-(58) update the variable *x_dist* from 0 to 4 in case the robot or the the human are approaching to each other. The command (59) covers the remaining transition from 4 to 5 in case a safety stop was not activated on time, and the command (60) makes the variable *x_dist* update from 5 to 0 by steps of 1.

The state transitions of variables *x_human*, *x_motion* and *x_aware* are illustrated in Fig. 8, and the transitions of variable x_dist are shown in Fig. 9.

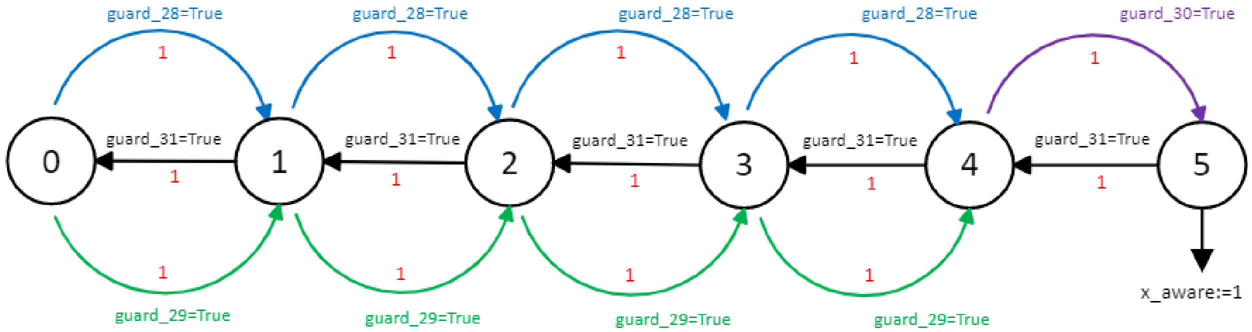


Figure 9: Diagram with the states transition for variable x_dist.

4.3.5 Synchronization

After defining the commands to update all the variables within a PRISM module, we need to ensure that the commands are executed in a specific order to emulate the behavior of the real system. One way to synchronize the execution of the commands presented above is to label each command with actions as was illustrated in expression (1). However, this kind of synchronization method depends on the product of the individual rates of transition of all commands with the same label. Thus, for synchronization purposes, we decided to use auxiliary Boolean variables called flags which restrict the execution of commands in a specific order each simulation cycle.

A flag can be assigned to one or more local variables within the model. A flag is always initiated as false and can be updated to true only after the corresponding variable was updated. Thus, a simulation cycle starts with all the flags set as false and ends when all of them are true (being reset as false after that). The order of updating these flags is shown in Figure 10 starting with the flag related with *x_uvc*, *x_logistics*, and *x_picking*, and ending with the flag related with variables *x_trays*, *x_runs*, *x_rows*, and *x_seg*.

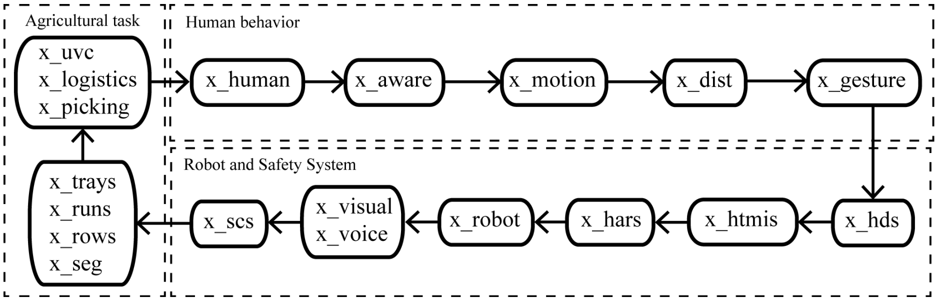


Figure 10: Diagram with the order in which the local variables are updated in PRISM.

For instance, in order to execute the command in (4) to make the variable *x_robot* change from 7 to 4, the command should be modified as follows:

$$(\mathtt{guard\_2}\ \&\ \mathtt{flag\_hars} = \mathtt{true}\ \&\ \mathtt{flag\_robot} = \mathtt{false}) - > (\mathtt{x\_robot}' = 4)\ \&\ (\mathtt{flag\_robot}' = \mathtt{true}); \tag{65}$$

Apart from this modification, it is necessary to include an additional command per each variable in order to avoid being stuck in the execution of the same flag forever when no update is required. By considering again the *x_robot* as example, the extra command have to make *flag_robot*=true when none of the conditions in (7)-(8) are satisfied. Thus, the extra command for the *x_robot* example takes the following form.

$$(\mathtt{flag\_hars} = \mathtt{true}\ \&\ \mathtt{flag\_robot} = \mathtt{false}\ \&\ \mathtt{guard\_sync} = \mathtt{false}) - > (\mathtt{flag\_robot}' = \mathtt{true}); \tag{66}$$

where $\mathtt{guard\_sync} = (\mathtt{guard\_2} \mid \mathtt{guard\_3})$.

4.4 Human injury assessment

Using the PRISM-based model[5] described in Section 4.1, this section presents a sensitivity analysis that evaluates the probability of human injuries in the presence of the most relevant failures identified in Table 3. The evaluation considers three cases where we assumed three different overall performances of the robot safety system and the human behavior/perception. These three cases are called i) the *ideal* case, when the safety system is working almost perfectly; ii) the *regular* case, when the safety system is as reliable as would expect in real-life; and iii) the *worst* case, when the safety system is not at all reliable. Tables 14, 16 show the values of the constants (used as probabilities of transition) that were chosen to characterize the three cases to be evaluated. Then, the sensitivity analysis for each case is performed as follows. We fixed most of the probabilities but varied from 0% to 100% (in steps of 10%) the values which correspond to the behavior of a specific failure in Table 3. For all the cases evaluated, the agricultural tasks were defined by using the following constant values: N_rows = N_runs = 5, N_segments_shed = 2, N_segments_collect = 1, and N_trays = 2.

The results in Tables 4-6 show the consequences of: i) having or not unplanned HRI, ii) having HRI with a trained or untrained people, and iii) having two different levels of failure being the *ideal* case 10% and the *worst* case 100%.

4.4.1 UV-C treatment

Table 4 shows the probability of getting injuries of types **HI-1**, **HI-2** and **HI-3** according to the specific failure evaluated. The failures **F-G2** and **F-G6** are not evaluated in the UV-C treatment since those are failures related with farm workers, and it was established in **SP-8** that trained workers knows that they can not interact with robots inside the polytunnel during UV-C treatment.

As was expected, the highest probabilities of getting human injuries happen under the condition of 100% of probability of failure with 100% of probability of unplanned HRI with untrained people. In the *worst* case, the highest probability of getting human injuries reaches 48.22%, 46.5% and 100% for injuries of type **HI-3**, **HI-2** and **HI-1** respectively. In contrast, when comparing these results with the *ideal* case, the human injuries related to physical contact are almost negligible but the injuries due to UV-C light exposure remains as dangerous as

[5] The PRISM-based model and properties used for the sensitivity analysis can be seen at `https://drive.google.com/drive/folders/1l2EIftoKPoewKHVeg2VtZq9lNpSRntHW?usp=sharing`. This includes the model described above for the UV-C treatment scneario and models for the logistics and picking scnearios as well.

Table 4: Resultant human injuries during UV-C treatment operations.

Scenario	Overall safety system performance	Human training level	Probability of unplanned HRI (%)	Probability of failure (%)	Probability of getting human injuries according to the failure evaluated (%)				
					HI-3	HI-2		HI-1	
					F-G1	F-G4	F-G7	F-G4	F-G5&F-U1,2
UV-C treatment	ideal case*	untrained	10	10	0.021	0.186	0.19	10.86	17.77
				100	0.21	6.581	0.566	61.25	20.75
			100	10	0.2	1.523	1.839	70.54	88.7
				100	2	37.37	5.385	100	92.98
		trained	10	10	0.019	-	0.177	-	-
				100	0.19	-	0.557	-	-
			100	10	0.19	-	1.715	-	-
				100	1.9	-	5.299	-	-
	regular case*	untrained	10	10	0.072	0.486	0.46	17.23	27.36
				100	0.727	5.721	0.9	61.25	34.18
			100	10	0.679	4.495	4.24	87.73	97.92
				100	6.645	32.72	8.17	100	99.59
		trained	10	10	0.068	-	0.427	-	-
				100	0.68	-	0.873	-	-
			100	10	0.638	-	3.962	-	-
				100	6.257	-	7.947	-	-
	worst case*	untrained	10	10	1.526	2.496	3.653	25.72	41
				100	7.44	8.879	4.22	61.25	47.68
			100	10	6.05	19.64	26.53	97.09	99.95
				100	48.22	46.5	30.14	100	99.99
		trained	10	10	0.6	-	2.89	-	-
				100	5.97	-	3.46	-	-
			100	10	4.88	-	21.83	-	-
				100	41.33	-	25.67	-	-

* see Tables 14, 16 for the transition probabilities used in each case.

in the *worst* case. This result demonstrate the importance of training everybody on the farm about dangers of approaching a robot during UV-C treatment but also the importance of the AVFAS to alert untrained people of the danger in the case the training fails.

4.4.2 Logistics and Picking

Unlike for UV-C treatment, for logistics and picking, the failures **F-U1** and **F-U2** are not evaluated but the failures **F-G2** and **F-G6** are included instead. Thus, Tables 5-6, show probabilities of injuries of types **HI-2** and **HI-3** only. According to these results, the probability of injury when SCS fails during both logistics or picking can reach above 90% in the *worst* case and around 20% in the *ideal* case. In general, the probabilities of getting human injuries during logistics operations have similar magnitudes but always higher than when robot is on picking operations. For instance, the probability of human injuries of type **HI-2** can reach 100% during logistics (due to **F-G4**), but for picking the resultant probability is up to 92.35%. They both are really critical results, and the slight difference is due to in logistics the robot performs more two way trips from polytunnel to collection point, so the chances of having unplanned HRI are higher.

As was mentioned for UV-C treatment, mandatory training and proper AVFAS design are crucial factor to attenuate the consequences of **F-G4**. Moreover, for logistics and picking, the HARS performance is another crucial factor to be considered since if **F-G6** happens with 100% of probability, then it may produce human injuries of around 80% even in *ideal* conditions.

5 Integration and evaluation

We have designed a Human-Aware Navigation (HAN) module based on the safety system presented in Subsection 3.4. This module can be integrated into the standard autonomous navigation system of commercial agricultural robots. In this section, we provide a much more detailed description of the components of the HAN module.

5.1 Robot Platform and Hardware

The robotic platform used to implement the HAN was a Thorvald II. Figure 11 shows the configuration of the Thorvald II used along with the list of hardware components mounted on it. In this list, components marked in

Table 5: Resultant human injuries during logistics operations.

Scenario	Overall safety system performance	Human training level	Probability of unplanned HRI (%)	Probability of failure (%)	Probability of getting human injuries according to the failure evaluated (%)					
					HI-3	HI-2				
					F-G1	F-G2	F-G6	F-G4	F-G5	F-G7
Logistics	ideal case*	untrained	10	10	1.681	14.15	-	14.16	14.19	14.2
				100	15.65	15.33	-	36.20	14.25	16.06
			100	10	2.255	18.21	-	18.3	18.59	18.61
				100	20.46	28.63	-	100	19.11	27.89
		trained	10	10	1.7	14.3	14.35	-	14.35	14.35
				100	15.82	15.54	78.93	-	14.38	15.4
			100	10	2.45	19.68	20.11	-	20.10	20.11
				100	22.09	30.57	87.92	-	20.44	22.3
	regular case*	untrained	10	10	3.549	22.32	-	22.4	22.44	22.13
				100	30.56	23.74	-	37.73	22.78	25.14
			100	10	5.376	30.96	-	31.7	31.97	31.08
				100	42.73	42	-	100	35.12	40.8
		trained	10	10	3.574	22.44	13.28	-	22.58	22.34
				100	30.74	24.05	69.71	-	22.88	24.7
			100	10	5.682	32.31	22.42	-	33.54	33.22
				100	44.58	44.73	81.93	-	36.18	37.92
	worst case*	untrained	10	10	27.27	54.67	-	52.89	55.08	53.77
				100	80.6	57.07	-	63.65	56.13	60.97
			100	10	31	82.54	-	73.97	83.7	83.82
				100	97.85	89.65	-	100	89.7	87.8
		trained	10	10	27.21	54.51	42	-	54.96	53.62
				100	80.4	57.17	82.29	-	56.01	60.22
			100	10	51.75	81.18	77.39	-	82.29	81.72
				100	96.3	88.08	93.84	-	87.95	88.5

* see Tables 14, 16 to know the transition probabilities used on each case.

Table 6: Resultant human injuries during picking operations.

Scenario	Overall safety system performance	Human training level	Probability of unplanned HRI (%)	Probability of failure (%)	Probability of getting human injuries according to the failure evaluated (%)					
					HI-3	HI-2				
					F-G1	F-G2	F-G6	F-G4	F-G5	F-G7
Picking	ideal case*	untrained	10	10	0.675	5.89	-	5.9	5.936	5.938
				100	6.578	7.176	-	29.13	5.966	7.365
			100	10	1.255	10.33	-	10.43	10.76	10.77
				100	11.9	21.75	-	88.11	11.04	20.41
		trained	10	10	0.747	6.497	6.548	-	6.547	6.548
				100	7.252	7.85	48.59	-	6.567	7.232
			100	10	1.976	16.04	16.5	-	16.49	16.5
				100	18.16	27.44	80.42	-	16.65	19.61
	regular case*	untrained	10	10	1.485	9.826	-	9.927	9.983	9.768
				100	14	11.47	-	26.95	10.18	11.83
			100	10	3.348	19.86	-	20.71	21.17	20.12
				100	29.07	32.61	-	81.47	22.87	30.26
		trained	10	10	1.629	10.71	6.466	-	10.9	10.73
				100	15.25	12.59	40.29	-	11.06	12.32
			100	10	4.798	27.72	20.33	-	29.19	28.55
				100	39.13	40.98	73.37	-	30.21	34.83
	worst case*	untrained	10	10	7.261	30.81	-	28.12	31.57	30.69
				100	53.93	34.41	-	43.73	32.31	35.19
			100	10	23.44	71.6	-	60.23	74.02	73.31
				100	93.97	80.97	-	92.35	76.24	78.16
		trained	10	10	7.564	31.94	25.27	-	32.67	31.61
				100	55.35	35.6	54.19	-	33.37	36.88
			100	10	24.83	73.63	70.5	-	75.26	73.17
				100	92.98	80.67	87	-	76.62	83.33

* see Tables 14, 16 to know the transition probabilities used on each case.

gray correspond to elements which are currently used for the autonomous navigation of the standard Thorvald II (as the one shown in Fig. 2a) , while the rest of the components were included specifically for our HAN module.

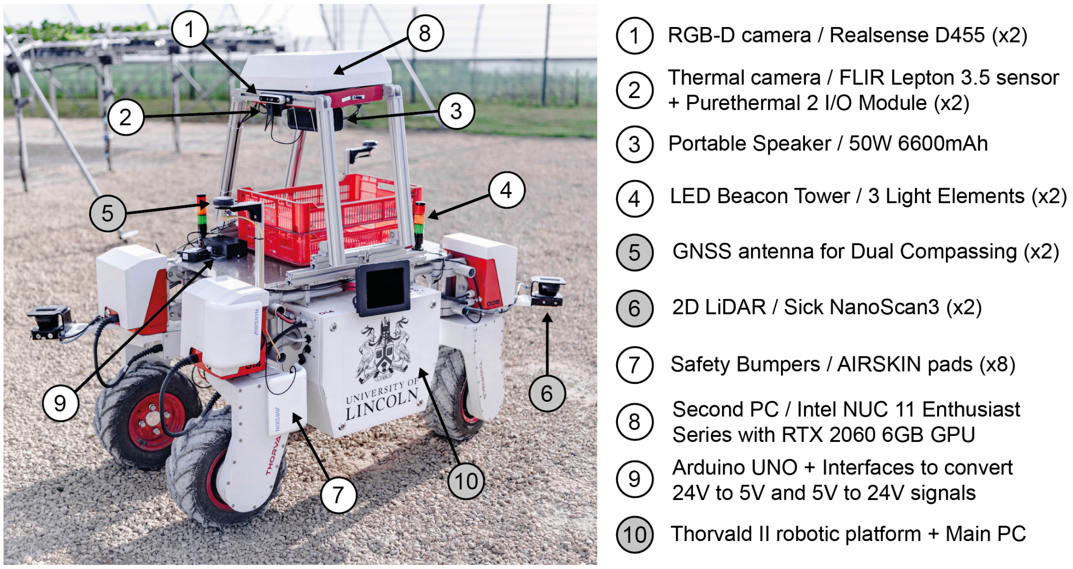

1
8
2
3
4
5
9
6
7
10
1 RGB-D camera / Realsense D455 (x2)
2 Thermal camera / FLIR Lepton 3.5 sensor + Purethermal 2 I/O Module (x2)
3 Portable Speaker / 50W 6600mAh
4 LED Beacon Tower / 3 Light Elements (x2)
5 GNSS antenna for Dual Compassing (x2)
6 2D LiDAR / Sick NanoScan3 (x2)
7 Safety Bumpers / AIRSKIN pads (x8)
8 Second PC / Intel NUC 11 Enthusiast Series with RTX 2060 6GB GPU
9 Arduino UNO + Interfaces to convert 24V to 5V and 5V to 24V signals
10 Thorvald II robotic platform + Main PC

Figure 11: Thorvald II robot with additional hardware components which are required to implement the proposed HAN module.

The new hardware components include a second computer with a dedicated GPU placed on the top of the robot with the aim of processing the data from RGB-D cameras, thermal cameras and 2D LiDARs. The aim of this processing is to extract information from any human interacting with the robot and as illustrated in the diagram in Fig. 12, this information is used as an input for a decision-making stage which based on HAN policies (defined by the users) decide if a safety action (e.g safety stop, reduced speed, gesture control) is required or not during HRI. Moreover, two LED beacon towers and a speaker are included to give audiovisual feedback to the humans about what the robot decides to do. A detailed description of each component included in the proposal is presented in the following subsections. [6]

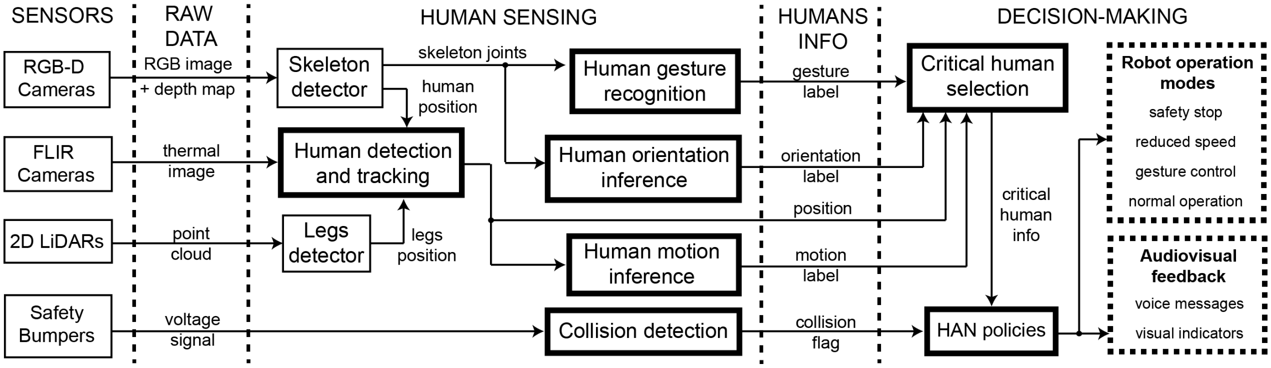


Figure 12: Overall architecture of the proposed HAN module.

5.2 Autonomous navigation standard components

To be able to navigate autonomously on footpaths or inside polytunnels, robust localization, precise motion control, and optimal route planning are required. These components are already part of the standard Thorvald II navigation system and are briefly described below.

[6] The proposed HAN module is contained into a ROS package that can be found in https://github.com/LeonardoGuevara/mesapro.

5.2.1 Robot localization and motion control

The localization of the Thorvald II depends on the information provided by the robot odometry, the use of RTK antennas for dual compassing [12], and the use of two 2D LiDARs which cover 360 degrees around the robot. The localization based on GNSS information is sufficient to localize the robot in open spaces such as footpaths, but when a higher precision is required (i.e., when moving along narrow rows within polytunnels), the LiDAR information is essential to avoid the robot colliding with the external infrastructure [23]. In terms of motion control, the robot movements inside polytunnels are constrained by the maneuverability space, making the robot able to move only in backwards or forwards mode. On the other hand, when the robot is moving in open spaces outside of polytunnels, the Thorvald II kinematic properties allow the robot to perform zero radius turns and move in sideways mode. The piece of code used for the localization and motion control of the robot along the rows inside the polytunnels was developed by the company SAGA robotics [7], which commercializes the Thorvald II platform and which was a partner in the MeSAPro project. More information about the Thorvald II hardware and software design can be found in [13, 14].

5.2.2 Topological navigation

To make the Thorvald II move autonomously from one point to another of the field, the route generation is based on topological navigation planning. In this kind of navigation, the robot follows paths formed by sequences of nodes in a topological map of the field. The nodes within this map are interconnected by edges that determine the possible actions or movements that the robot is allowed to do to move towards a nearby node, and planning consists of finding a path (in the graph-theorectic sense) through the graph composed of the nodes and edges. For example, if the robot is located on an edge that connects two nodes inside the polytunnels, this means that the robot is only allowed to move between nodes in the same row but never between different rows. This isn't a limitation outside polytunnels, where the robot can move to any node that make the route optimize a certain criterion (typically the shortest distance to the goal). The topological navigation framework is contained in a public ROS package that has been successfully applied in long-term autonomy application [4, 8]. In this work we modified the original package to make it compatible with the HAN requirements as will be explained later [8].

5.3 Human sensing components

5.3.1 Human detection and tracking

The proposed human detection relies mainly on using color/depth images (taken from two RGB-D cameras), and thermal images (taken from two thermal cameras aligned to the RGB-D cameras). However, the point cloud generated by the 2D LiDARs (originally installed for robot localization purposes) can be used as an additional data source. The Camera-based detection uses the OpenPose framework [6] to detect human centroids (by extracting skeleton joints) in the RGB images and determine their positions with respect to the camera frame. Then, these positions are mapped to the robot frame by using the depth image, and finally, the thermal image is used to reduce the number of false-positives delivered by a pre-trainned skeleton extractor model. Basically, a human detection is valid only if the pixels of the thermal image where the human centroid was detected satisfy a threshold of intensity. Once a human has been continuously detected for a certain number of frames, the position of this human starts to be tracked.

To complement the Camera-based detection, the LiDARs can also be used for human detection purposes. This LiDAR-based detection is based on the work presented in [18] which uses the so-called Distance Robust Spatial-Attention and Auto-regressive Model (DRSPAAM) to detect human legs in a 2D point cloud. This model was pre-trained with data from indoor environments, thus in order to reduce the false positive detection rate due to harsh lighting conditions outdoors and similarities between human legs and polytunnel infrastructure, a *no go map* of the polytunnel infrastructure is required (see Fig. 13(b)). Once the legs detection has been filtered by the *no go map*, the final step is to track the human position on time by using a particle filter as presented in [19, 3].

Figure 13(b) shows examples of the inputs, and outputs of the Camera-based skeleton detection (blue) and LiDAR-based leg detection (red). The ability of LiDAR-based detection to cover 360 degrees around the robot, while cameras have blind spots (as shown in Fig. 13(a)), is an argument for LiDAR-based detection being more suitable to ensure human-aware safe navigation than Camera-based detection. On the other hand, Camera-based

[7] SAGA robotics website `https://sagarobotics.com/`.

[8] The modified topological navigation package used in this work can be found in `https://github.com/LeonardoGuevara/topological_navigation.git`.

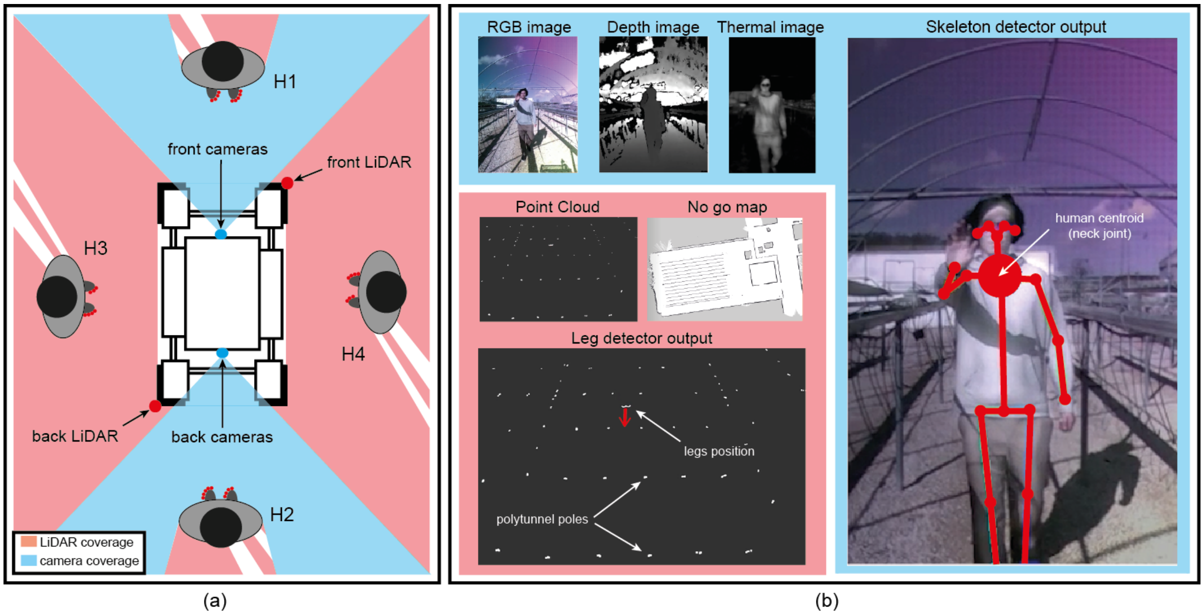


Figure 13: (a) Detection coverage areas according to the sensor used. (b) Inputs and outputs of the Camera-based skeleton detector (blue) and the LiDAR-based leg detector (red).

detection is able to extract additional features such as hand gestures and determine the orientation of the person detected (see Subsection 5.3.3). These additional features make the Camera-based detection suitable for enabling a more efficient and smooth HRI than what is possible when only using position information.

As illustrated in Fig. 13(a), if the robot is moving inside the polytunnels, the cameras field of view allows for detecting people in the same row (H1 and H2), while the LiDAR detection allows the robot to be aware of people even in other rows (H3 and H4). When working inside the polytunnels, the human detection on the robot sides is not mandatory since people in other rows are not considered in risk of collision, but it can be useful when the robot is outside the polytunnels since in that case H3 and H4 may be at risk. Thus, in order to improve the overall performance of human detection and keep continuous tracking, it is recommended to combine LiDAR-based detection with Camera-based detection. In this way, if a person is no longer detected by the cameras because of the blind spots on the side of the robot, the LiDAR can still track the person's position and when he/she enters the camera vision range again, the position information will be complemented by the gesture and orientation information. The main problem of combining both detection systems is the computational cost since it requires the second robot computer (the computer on the top of the robot) to run two algorithms based on convolutional neural networks at the same time which is not suitable for long time periods in the kinds of high temperatures that occur in polytunnels during harvesting. For this reason, to have a better idea of which kind of detection system is more suitable for outdoors implementation, Table 7, shows a qualitative and quantitative comparison of the human detection performance when using different combinations of sensor data. We compared 6 combinations, starting from the ones which use a single kind of sensor data, and finishing with the fusion of all the available sensor data. Overall, every combination shows a good detection performance in terms of accuracy and recall metrics. However, the combinations which include LiDAR data, obtained a lower precision metric which means a higher number of fake detections. This isn't a problem if safety is the main goal since the robot will stop as soon as any human is detected within unsafe distances. However, if efficiency and smoothness of the HRI is considered, the excessive and unnecessary stops due to fake detections will cause stress and loss of trust on the human co-workers. A good alternative to avoid this problem can be the use of RGBD + thermal data which although having limited detention range, obtained the highest F-score, includes valuable human information necessary for HRI, and is not computationally expensive (i.e. can be operated for long periods of time).

5.3.2 Human gesture recognition

As was mentioned above, the Camera-based detection uses OpenPose to detect human in the RGB images. However, the purpose of using OpenPose is not only human detection but also the extraction of human skeleton joints, that later can be used for gesture recognition. In this context, we followed the ideas proposed in [30, 31], to con-

Table 7: Comparison of human detection based on different sensor data.

Sensor used	**Sensor detection range**	**Sensor data frequency**	**Data processing requirements**	**Human information available**	**Human detection performance**			
					Accuracy	**Precision**	**Recall**	**F-score**
LiDAR	360 degrees	30fps	Moderate processing	Position	0.9610	0.6738	0.9895	0.7922
RGBD	blind spots	30fps	Moderate processing	Position, gesture, and orientation	0.9436	0.9355	0.9800	0.9395
RGBD Thermal	blind spots	9fps	Moderate processing	Position, gesture, and orientation	0.9431	0.9451	0.9779	0.9441
LiDAR RGBD	360 degrees	30fps	Computationally expensive	Position, gesture, and orientation	0.9389	0.6301	1	0.7542
LiDAR RGBD Thermal	360 degrees	30fps & 9fps	Computationally expensive	Position, gesture, and orientation	0.9363	0.6701	1	0.7812

struct a feature vector made up of the relative distances from each skeleton joint to the neck joint and the angles between consecutive joints. Since the size of the human in the frame may vary according to the distance from the person to the camera, the features extracted on each frame must be normalized (see [30] for more details).

In situations when a human is too close to the camera, OpenPose is capable of extracting the human skeleton even if some of the lower body joints are missing (not visible). However, we need to ensure that no important joints are missing for gesture recognition, thus we decided to consider only upper body joints (red color in Fig. 14(a)) which are more likely to remain visible than the lower body joints (blue color in Fig. 14(a)). In this context, we use the 15 upper body joints from the *BODY_25* skeleton model which are transformed into a total of 28 features including 15 distances from each joint to the neck joint and 13 angles between consecutive joints. Those are the features we used as inputs to a Random Forest classifier [5] which was trained to classify 11 different hand gestures. These hand gestures will later be interpreted by the robot as commands (see Subsection 5.4.4). Figure 14(b) shows a sample image of each gesture with the name of the labels used for their training. To collect the data training set, we recorded videos of a person performing every gesture (the same during the whole video) while moving in front of the robot keeping a distance between 1.2m and 3.6m (i.e. within the *social* space).

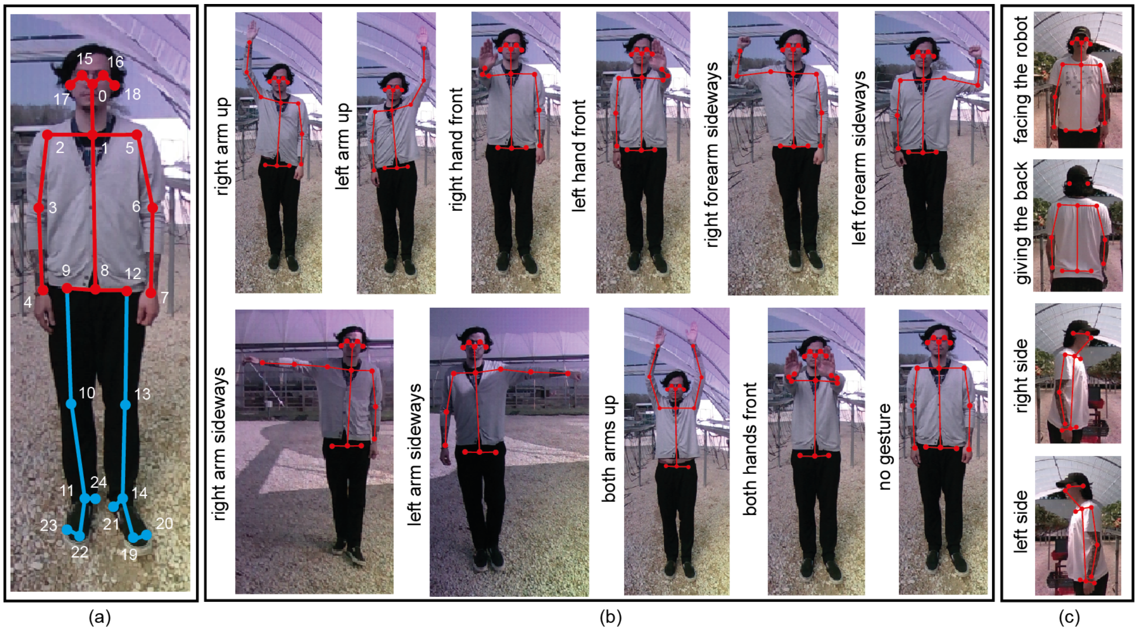


Figure 14: (a) *BODY_25* human skeleton estimation model showing upper body joints in red and the lower body joints in blue. (b) Examples of every hand gesture that can be interpreted by the robot using only upper body joints. (c) Examples of orientation inference based on missing head joints.

In total, the number of samples of each gesture label in the dataset was: *no gesture* (3384), *left arm up* (2997), *left* *hand front* (5602), *left arm sideways* (4666), *left forearm sideways* (3822), *right arm up* (4057), *right hand front* (2938), *right arm sideways* (4610), *right forearm sideways* (3813), *both arms up* (4073), *both hands front* (3825). The gesture dataset was split into training (70%) and testing (30%). Figure 15 summarizes the proposed gesture recognition performance using a normalized confusion matrix. According to this matrix, the prediction results obtained from the Random Forest classifier are shown to be very accurate for every gesture. These results may vary if the distance between the robot and the person is higher than 3.6m, thus, we restricted the gesture recognition to the *social* zone since the gesture control feature (see Subsection 5.4.4) is intended to be used in close distance interactions only.

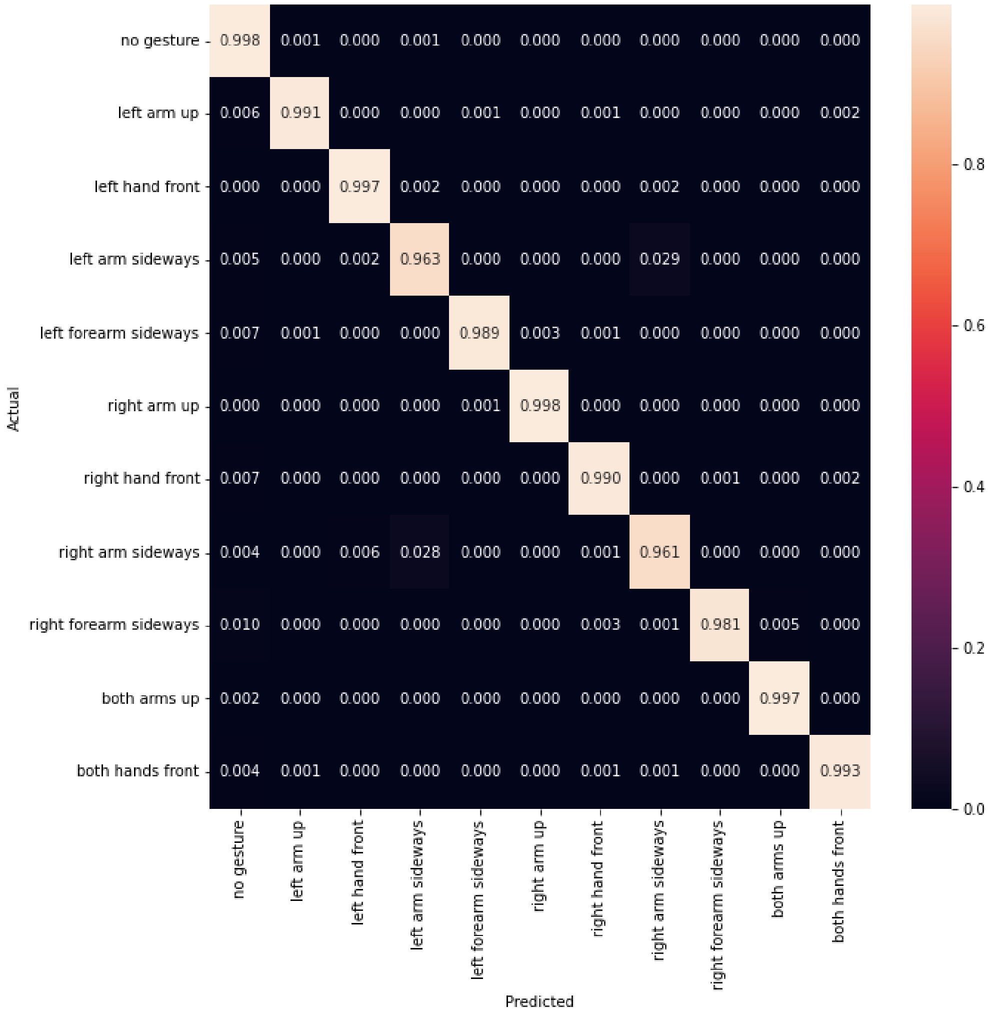


Figure 15: Normalized confusion matrix of the gesture recognition using a Random Forest classifier.

5.3.3 Human orientation and motion inference

As an addition to the information of the position of the human and label for the hand gesture, the human detection code also extracts information about the orientation of the detected human with respect to the robot and infers whether the human intends to stay stationary. In order to infer the human orientation and deliver a resultant label, the joints extracted by OpenPose are used once again. According to the head orientation, the OpenPose output may deliver a different number of joints. This behavior allows us to easily infer the human orientation by analyzing the joints that are not being detected. For instance, as is shown in Figure 14(c), the lack of the nose joint determines if the human is *facing the robot* or *giving the back* to the robot, and the lack of a specific ear joint determines if the human is detected from one side or another. The resultant orientation label is important to address the consideration in *SA4* which limits the robot to interact with a human only by facing them and never moving when behind them. On the other hand, a label obtained from human motion inference is going to address the consideration *SA2*. This inference is not aimed to predict future movements or infer the human intentions as a form of *implicit* communication. The purpose of this inference is simply make the robot stop if the human is currently moving, .i.e. always giving priority to the human partner as was stated in *SA2*. In this context, the way to infer the motion is extremely simple, the robot is continuously recording the displacement of humans tracked (by using cameras and/or LiDARs) and computing the average speed within a time window. If the speed is higher than a given threshold, then the motion label turns from *mostly stationary* to *moving*.

5.3.4 Collision detection

In the worst scenario, when a failure makes the robot not aware of an obstacle or human presence on time, the robot also requires *post-collision* actions such as equipping it with soft coverings around the chassis to absorb the energy of the impacts, and making the robot stop immediately to limit the injuries. With this aim, the Thorvald II was equipped with 8 soft pads [9] which act as safety bumpers making the robot to activate an e-stop when a contact is detected. As shown in Fig. 11, the pads where placed on the corners of the robot which are the most likely places to produce injuries when a human worker is too close to the robot, especially outside polytunnels where the robot can rotate or move sideways. The 8 AIRSKIN modules are connected in series ending in a connection box which delivers a 24V signal when no collision is detected or 0V when a collision is detected. This signal is converted to a digital signal using an optocoupler module in order to interfaces with an Arduino UNO. The Arduino UNO monitors the digital signal and communicates with the second PC (on the top of the robot) by using the library Rosserial Arduino.

5.4 Human-robot communication components

This Subsection describes the components of the proposed HAN module which allow the human workers to receive audiovisual feedback from the robot operation, but also allows the robot to understand the intentions and commands of the human workers.

5.4.1 Visual indicators

The Thorvald II was equipped with two LED beacon towers with three light colors to address the consideration *SA5*. The activation of a specific visual indicator depends on three different factors: i) the robot current operation (teleoperation or autonomous mode) ii) failures of any component of the HAN module iii) human presence and HRI risk level (see Subsection 5.5.1). The Table 8 summarizes the visual indicators and a description of their meaning. The colors and type of indicator (blinking or continuous) follow the standards described in [9] which aim to make them as intuitive as possible for human workers without previous experience interacting with the robot. For instance, since the color red is widely associated with emergency or danger situations, then, in our case, it is used to warn the human that the robot operation is dangerous and they shouldn't get closer to it while the alert is still on. The second color is yellow, which is a popular beacon color used mainly to let the people know they must proceed with care. Finally, since the color green is often associated with safety or go ahead, then, in our case it is used to show that there is no risk of being harmed while interacting with the robot. The visual indicators are activated by the same Arduino UNO that was used for collision detection, i.e., it receives the commands to change the visual alerts via ROS by using the library Rosserial.

It is worth mentioning that the tower light that we used follows the standard design of vertical traffic lights where each color is activated by a separate LED, with the red light located at the top and green light at the bottom. Thus, people who suffer from colour blindness can distinguish the alerts by seeing which light is on rather than by recognising the colors.

Table 8: A description of the indicators used as visual feedback for the human interacting with the robots.

Type	Color	Description
Continuous	Red	It means that a human is detected in a danger zone, or a collision was detected
	Yellow	It means that a human is detected in a warning zone
	Green	It means that a human is detected in a safe zone
Blinking	Red	It means that there is a failure in a component of the HAN module
	Yellow	It means no human is detected when robot operates autonomously
	Green	It means that teleoperation mode is activated

[9] The pressure-sensitive sensors used for collision detection are commercialized by AIRSKIN https://www.airskin.io/airskin.

5.4.2 Voice messages

The visual indicators mentioned above were implemented to increase the level of understanding of robot operations. Although visual feedback is a simple and intuitive way of communication in most of cases, its effectiveness depends on the level of training people interacting with the robot. Thus, in order to increase the effectiveness of Human-Robot Communication (HRC), we also incorporated a portable speaker to reproduce *explicit* voice messages that make people aware of robot current actions. In this way, independently of training level, the human workers can be aware of what the robot intentions are. The library of voice messages consist on 11 pre-recorded messages created using Wideo [10] in two languages: English (default) and Polish (optional). The second language was incorporated since a large part of human labor in UK farms comes from eastern European countries such as Poland and some of them do not speak English [11]. The voice messages generated by the Wideo software aim to be as natural and realistic as possible since according to [10], robots with human-sounding voices cause less stress to humans than robots with synthetic-sounding voices.

5.4.3 Joystick teleoperation

The audiovisual alerts mentioned above create an *unidirectional* communication channel between the human and the robot. However, a *bidirectional* communication is necessary for robot to be able to interpret what the human's actions or intentions are. The simplest way to accomplish this is by introducing a *teleoperation* mode in which the robot can be controlled by a human operator. As in every commercial robotic platform, the standard version of Thorvald II includes a *teleoperation* mode by using a joystick. When the teleoperation is activated, the human operator is completely in charge of the robot actions, and it is mainly used for moving the robot from the storage shed to the polytunnels. Once the robot is at the polytunnels, it can start operating in autonomous mode without requiring the human operator.

5.4.4 Gesture control

The main problem with a typical *teleoperation* mode is that it is limited to a single person interaction (the one who has the joystick). Thus, if multiple people are interacting with the robot at once, most of them are not able to command the robot. To address this limitation, another teleoperation mechanism was implemented by using hand gestures. This operation mode is called *gesture control* and uses the gesture labels mentioned in Subsection 5.3.2 to let the robot being controlled by any person who knows the specific hand gesture. Table 9 shows the list of commands based on those gestures, including a description of their usage and their influence in terms of topological navigation and velocity control. It is important to notice that depending on if a gesture is detected inside or outside polytunnels, the command can make: i) the robot change the topological goal and continue working autonomously or ii) continue with the current goal while the human commands the robot velocities as in *teleoperation* mode.

In order to avoid unnecessary robot movements due to false positives in the gesture recognition, the gestures are considered as valid commands only if: i) they have been continuously detected for a certain number of frames, ii) the human performing the gesture is not giving the back to the robot, and iii) the robot is within the *social* space of the person performing the gesture. The third condition is applied due to the skeleton joints extraction using OpenPose is not reliable enough when the robot is outside the *social* space of a human that it has detected.

5.5 Decision-Making components

This Subsection describes the HAN policies, HRI risk level criteria, and robot operation modes used by the decision-making stage shown in Fig.12.

5.5.1 HRI Risk levels

Figure 16 shows illustrative examples of the HRIs that may happen when the Thorvald II navigates inside the polytunnels (left) and outside the polytunnels (right). In these examples, the robot shares the workspace with three workers located at different distances. The risk of producing collisions in these examples is classified into three levels or zones (represented by different colors): *danger* (red), *warning* (yellow) and *safe* (green). The risk level

[10] Wideo includes a text to speech software available for free at `https://wideo.co/text-to-speech/`.

[11] Examples of the audiovisual alerts implemented in the Thorvald II are shown at `https://www.youtube.com/watch?v=WLEGSuPtYJU`.

Table 9: Description of human commands that can be interpreted by the robot based on hand gesture recognition.

Command	Corresponding hand gesture	Location	Command description	Topological navigation	Velocity control
Move towards me	Right forearm sideways	Polytunnels	Makes the robot start reduced speed mode after the gesture was detected	Updates the goal	automatic
		Footpaths	Makes the robot operate in reduced speed mode while the gesture is being detected	Keeps the current goal	teleperated
Move away from me	Left forearm sideways	Polytunnels	Makes the robot start normal mode after the gesture was detected	Updates the goal	automatic
		Footpaths	Makes the robot move away while the gesture is being detected	Keeps the current goal	teleperated
Stop operation	Both hands front	Polytunnels and Footpaths	Makes the robot activate stop mode after the gesture was detected	Keeps the current goal	none
Move to your left	Right arm sideways	Footpaths	Makes the robot move sideways while the gesture is being detected	Keeps the current goal	teleperated
Move to your right	Left arm sideways	Footpaths	Makes the robot move sideways while the gesture is being detected	Keeps the current goal	teleperated
Rotate clockwise	Right arm up	Footpaths	Makes the robot rotate while the gesture is being detected	Keeps the current goal	teleperated
Rotate counterclockwise	Left arm up	Footpaths	Makes the robot rotate while the gesture is being detected	Keeps the current goal	teleperated

in a certain situation is determined according to i) the robot location with respect to the polytunnels, ii) the human location with respect to the current robot goal, and iii) the invasion of *personal* or *social* spaces.

According to the risk level, the robot operation changes in order to minimize the chances of having unwanted physical contact. The robot operation modes will be described in detail later in Subsection 5.5.4. For now it is important to understand why some of the HRI in Fig. 16 are considered within a *danger* zone but other not. For instance, the human H2 and H3 they both are located 1m away from the robot (i.e. *personal* space), however only H2 is considered to be in danger while H3 is within a *safe* zone. This is because the robot is constrained to navigate inside the polytunnels only forward or backwards, thus there is not any chance to move sideways to another row and collide with H3. On the other hand, when the robot is moving outside the polytunnels towards the collection point, there is a chance to collide with people on its side as the example with H5. In the example outside the polytunnels is also important to notice that although H6 is located at 2.5m from the robot (i.e. *social* space), it is considered to be within a *safe* zone since the current robot path has location H4 as its goal. Thus, any person detected in the opposite direction than the robot goal is considered to be at a *safe* zone since the robot is not going to move in that direction.

5.5.2 Critical human selection

Since there are several human workers distributed within the field during the cooperative harvesting operations, it is expected that at some point the robot will interact with more than one human worker at a time (as was illustrated in Fig. 16), thus in case of detecting multiple humans, it is necessary to determine which human is the most critical to be monitored by the robot. This task is called the *critical human selection*, and in this work we considered the following considerations as priority criteria for selection:

CH1 If more than one person is detected nearby the robot and they are located in different risk zones, then: the people within *danger* zone are considered as the highest priority, followed by people in *warning* zone and finally people in *safe* zone as the last in priority order.

CH2 If more than one person is detected in the same risk zone, then: the closer the detected person is, the higher the priority they have with respect with other people in the same zone.

CH3 If more than one person is detected in the *warning* zone, then: apart from the condition in *CH2*, if one person is performing hand gestures then they have priority over another one who can be closer but not performing hand gestures.

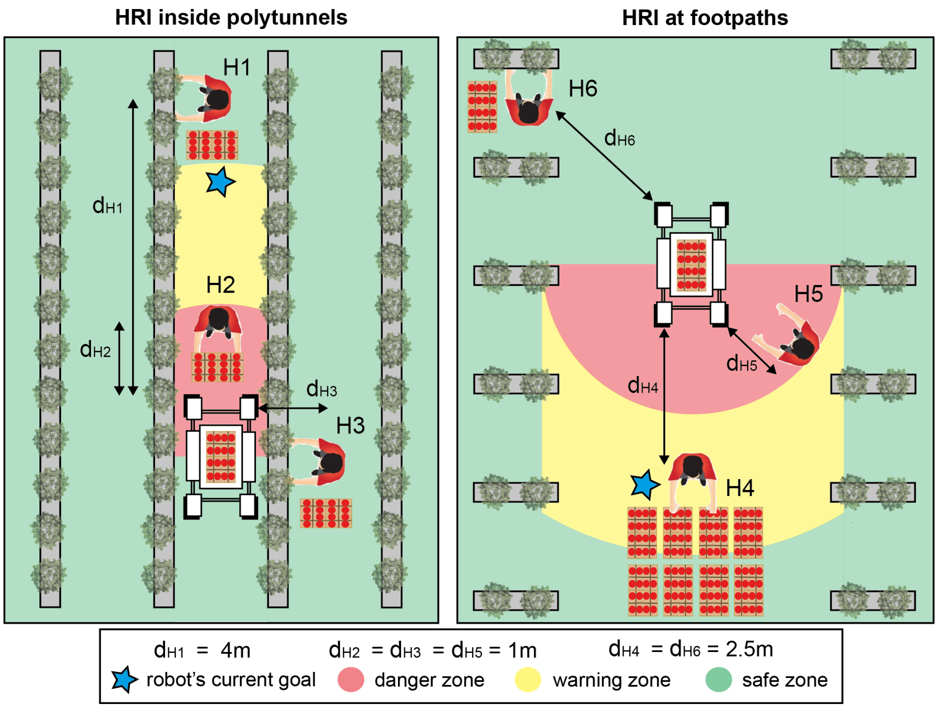


Figure 16: Illustrative examples of possible HRIs that may happen when a Thorvald II is moving inside polytunnels (left), at footpaths next to the collection point (right).

5.5.3 HAN policies

The HAN policies can be divided into three different types: safety-aware policies, ergonomic-aware policies, and efficiency-aware policies. Each kind of policy requires the robot to operate in different modes (see Subsection 5.5.4). For instance, the safety-aware policy aims to address the safety considerations mentioned in Subsection 3.3 by making the robot activate *pre-collision* and *post-collision* actions as follows:

SAP1 The robot must activate *stop* mode if at least one of the following conditions is met: i) the critical human is in *danger* zone, ii) a collision was detected, iii) the critical human asked the robot to stop by using *gesture control* mode iv) a failure is detected in a component of the HAN module.

SAP2 If the robot activated *stop* mode because a collision was detected, then the robot motors are locked, and can be unlocked only manually by pressing a button on the robot.

SAP3 Once the robot is in *stop* mode it can be reactivated (i.e. start moving) only if no failure is detected in the HAN module and the robot motors are unlocked. Moreover, one of the following conditions has to be met: i) the critical human gives the robot a new goal or command it to move by using *gesture control* mode, ii) a human picker asks the robot to move to his/her location (i.e. topological goal is updated).

The ergonomic-aware policies aims to make the robot interact with the critical human in a socially acceptable way as follows:

EAP1 The robot can activate *gesture control* mode only if all the following conditions are met i) the critical human is in the *warning* zone, ii) the critical human is *facing the robot*, iii) the critical human stays *mostly stationary*.

EAP2 When *gesture control* mode is activated inside polytunnels, only three commands are allowed: *move towards me*, *move away from me*, and *stop operation*.

EAP3 When *gesture control* mode is activated inside polytunnels, the critical human only needs to perform a gesture momentarily to make the robot move or stop.

EAP4 When *gesture control* mode is activated outside polytunnels, the critical human needs to perform a gesture continuously to make the robot react accordingly.

EAP5 Robots can move towards a critical human in *reduced speed* mode only if all the following conditions are met i) the critical human isn't in *danger* zone, ii) the critical human is *facing the robot*, iii) the critical human stays *mostly stationary*, iv) the critical human asked the robot to move towards him/her by using *gesture control* mode.

EAP6 When a robot is moving towards the critical human in *reduced speed* mode outside the polytunnels, the robot must rotate until it is aligned to the critical human before starting moving towards them.

The efficiency-aware policies aim to reduce the non-productive times generated during prolonged safety stops by reactivating the robot operation as follows:

EAP1 The robot activates *pause* mode when the critical human is in *warning* zone.

EAP2 Once the robot is in *pause* mode it can be automatically reactivated and resume the previous operation if the critical human moves to a *safe* zone or no human is detected.

EAP3 If robot has been in *stop* mode for a long time (without receiving a new human command), then the robot can decide if summarize the previous topological goal (if it was not reached) or move to the collection point.

5.5.4 Robot operation modes

The robot operation modes that are activated by the HAN decision-making can be split into the following modes:

- *Normal mode:* This is the default autonomous operation mode of the Thorvald II, which makes the robot navigate between two topological nodes inside or outside the polytunnels without applying any specific safety action apart from the audiovisual alerts that are always activated (as stated in *SA5*). This mode is activated when no human is detected or when the critical human is in a *safe* zone.
- *Teleoperation mode:* This is the default *teleoperation* mode of the Thorvald II, which allows a human operator to control it by using a joystick. Similar to *normal* mode, this mode does not apply any specific safety action apart from the audiovisual alerts that make the operator aware of the current robot operation.
- *Stop mode:* This is the main safety action which aims to minimize the chances of cause injuries to human co-workers. This mode is activated according to the policy *SAP1*. The robot can change to *normal* mode when a human worker asks the robot to move to a new goal as stated in the policy *SAP3*. Moreover, if no goal is given to the robot for a long time, the robot can decide to complete the previous route plan or decide to move to the collection point as stated in *EAP3*. The decision of where to move depends on whether the original route plan wasn't completed before being stopping operation and if there is any worker occluding the current robot path.
- *Pause mode:* This is an alternative version of the *stop* mode that makes the robot stop the current operation. The main difference between *stop* mode and this mode is that the robot can automatically reactivate *normal* mode following the policy *EAP2*, i.e. without receiving a human command or being waiting a long time for a command as was required for *stop* mode.
- *Gesture control mode:* This is similar to the *teleoperation* mode in the sense that the robot is commanded by a human operator. However, in the typical *teleoperation* mode, the robot velocity is manually controlled by the operator commands, on the other hand, by using *gesture control*, the human operator can change/update the robot topological goal while keeping the robot operating autonomously. For instance, when the robot is moving inside polytunnels, the *gesture control* mode is activated, and according to the human command, the robot can activate *stop* mode (keeping the same goal), or update the current topological goal to make the robot move towards the human (i.e. activating the *reduced speed* mode), or to make it move away from the human (i.e. activating the *normal* mode).
- *Reduced speed mode:* This mode makes the robot to move towards the critical human by reducing its speed till it stops completely when it reaches the *personal* space (addressing consideration *SA3*). This mode can be activated only after a human worker performs a hand gesture as stated in policy *EAP5*, i.e. the robot is not allowed to move towards any human without his/her consent.

5.6 Experiments under real conditions: logistics case study

The Thorvald II platform with the proposed HAN module (see Fig. 11) was tested in a strawberry polytunnel located at the Riseholme campus of the University of Lincoln, UK. Several experiments were performed in order to cover both ideal situations (normal cases) and non-ideal situations (special cases) in the context of cooperative harvesting [12]. This section shows the results of some these experiments with the intention of illustrating the satisfaction of the HAN policies defined in Subsection 5.5.3 and evaluating the user's experience after interacting with the robot. The experiments were performed during the UK harvesting season in 2022. Due to the high temperature levels during this period (with peaks of almost 40 degrees Celsius in the polytunnels), it wasn't possible to use Camera-based + LiDAR-based human detection systems for long periods without having overheating problems. For this reason, the experiments in this section show results of human sensing based only on RGB-D and thermal cameras which had the best detection performance according to Table 7. The main downside of this choice is the restricted detection range. Therefore, to mitigate the risk of colliding with people when they are in a blind spot, every person who interacted with the robot was informed in advance that the robot had blind spots in this configuration, and they were restricted to interact with the robot only from the front or the back.

5.6.1 Expected HRI inside polytunnels

The first experiment illustrates how the robot behaves when it has to interact with a picker (the one who summoned the robot) to load/unload trays inside the polytunnel. Figure 17 shows the evolution in time of the most important parameters related to the HRI, as well as snapshots of important moments during the whole experiment. Snapshot A shows the robot approaching the picker, till it activates *stop* mode for safety reasons. In this example, the safety stop was activated when the picker turned around, placing their back towards the robot as illustrated in snapshot B (i.e. the policy *EAP5* was not satisfied). Immediately after stopping, the robot activates a voice message making the picker aware of the *stop* mode and asking for a new command from the picker. This is when the picker answers the robot's request by making the robot move toward them by performing a hand gesture as shown in snapshot C. While the robot is approaching the picker, the *reduced speed* mode is activated, making the robot's linear velocity modulated according to the distance between them. The robot activates *stop* mode once again when it reaches the personal space of the picker as shown in snapshot D. At that point, the picker can approach safely the robot and load/unload trays on it. Then, the robot can move away from the picker (having a new topological goal) once the picker perform an specific hand gesture, however, as illustrated in snapshot F, any hand gesture performed while not facing the robot is not considered since it doesn't satisfy the conditions stated in policy *EAP1*. Once the picker turns and faces the robot again, they can make the robot move away and also make the robot stop by performing human gestures as illustrated in snapshots G and H respectively.

5.6.2 Expected HRI outside polytunnels

The second experiment illustrates how the robot behaves when it has to interact with a human worker to load/unload trays at the collection points outside the polytunnels. Figure 18 shows snapshots of important moments during the whole experiment. The experiment starts with the robot moving in *normal* mode towards the human location. Then, the robot activates *pause* mode when reaches the social space of the human as illustrated in snapshot A. Immediately after pausing the current operation, the robot activates a voice message making the human aware that they are occluding the current robot's path. At this point, the human worker can control the robot's movements by performing different hand gesture as illustrated in snapshots B, C, and D. As shown in Fig. 18(g), when the human is no longer performing a hand gesture, the robot operation changes from *gesture control* mode to *stop* mode. As shown in snapshot E, when the human asks the robot to move towards them, the *reduced speed* mode is activated, making the robot reoriented in order to be facing the human (following the policy *EAP6*), and then modulating its linear velocity according to the distance between them (see Fig. 18(e)). The robot activates *stop* mode once it enters the human *personal* space (*danger* zone) as stated in policy *SAP1*. At that point, the picker can approach safely the robot and load/unload trays on it as shown in snapshots F and G. Then, the robot can move away from the picker once the picker performs a specific hand gesture as illustrated in snapshot H. It is important to notice that unlike experiment 1, the *gesture control* mode outside the polytunnels does not modify the topological goal of the robot but it directly controls the robot's velocity commands.

[12] Videos of the experiments are available in `https://youtu.be/x9O0CB1MV5k`.

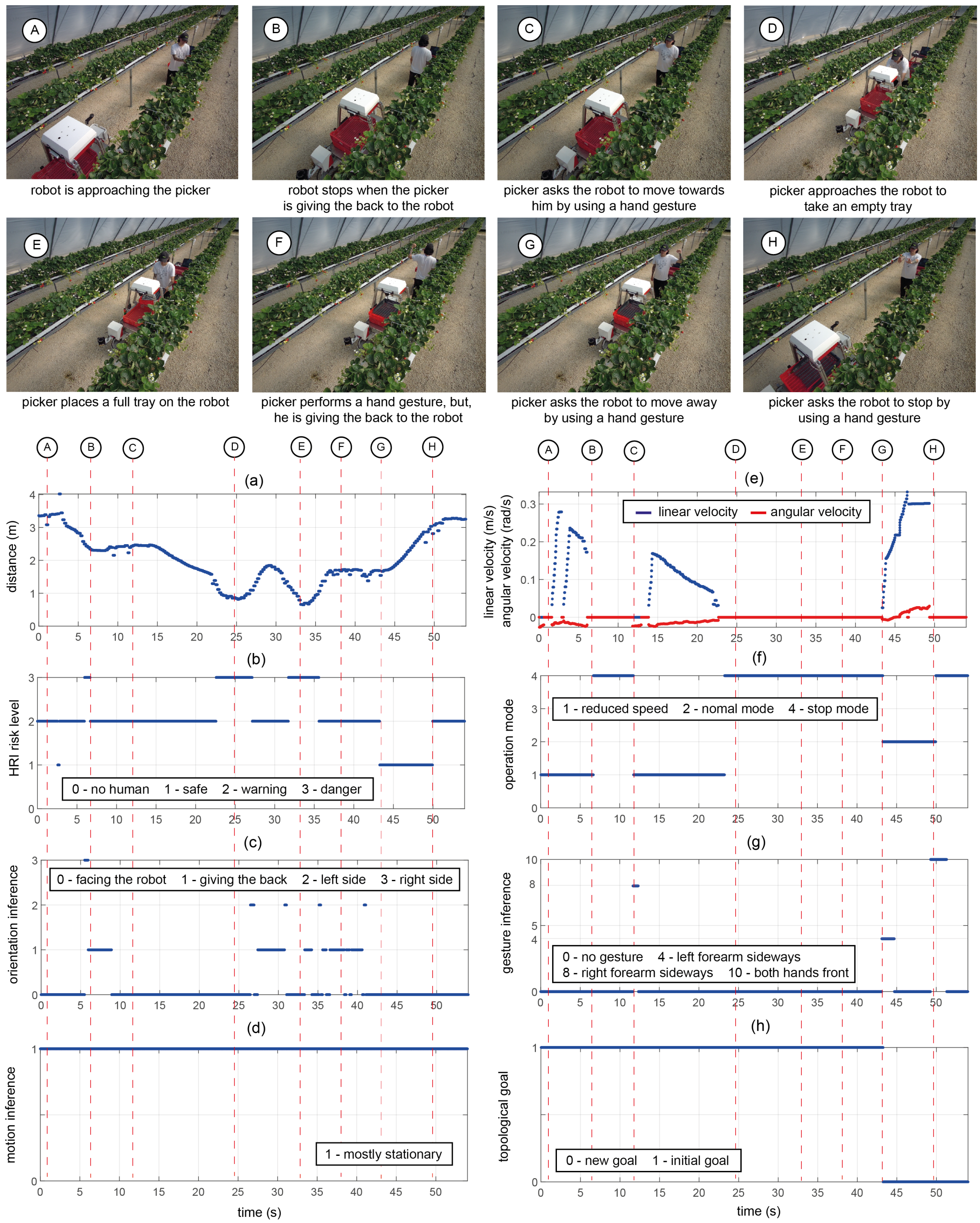


Figure 17: Snapshots from experiment 1 showing the evolution in time of parameters including: (a) distance between the robot and the picker (b) HRI risk level label (c) orientation label (d) motion label (e) robot linear and angular velocities (f) robot operation mode (g) gesture label (h) robot topological goal.

5.7 User experience evaluation

The aim of the experiments in Subsection 5.6 was to illustrate the satisfaction of the HAN policies defined in Subsection 5.5.3 in different situations that are expected to happen during cooperative harvesting operations. These experiments were performed with people who knew very well how the proposed HAN module works, i.e. they were familiar with the audiovisual alerts and the *gesture control* mode integrated into the HAN. However, if we consider a real farm implementation, we will face the situation in which the robot has to share the workspace

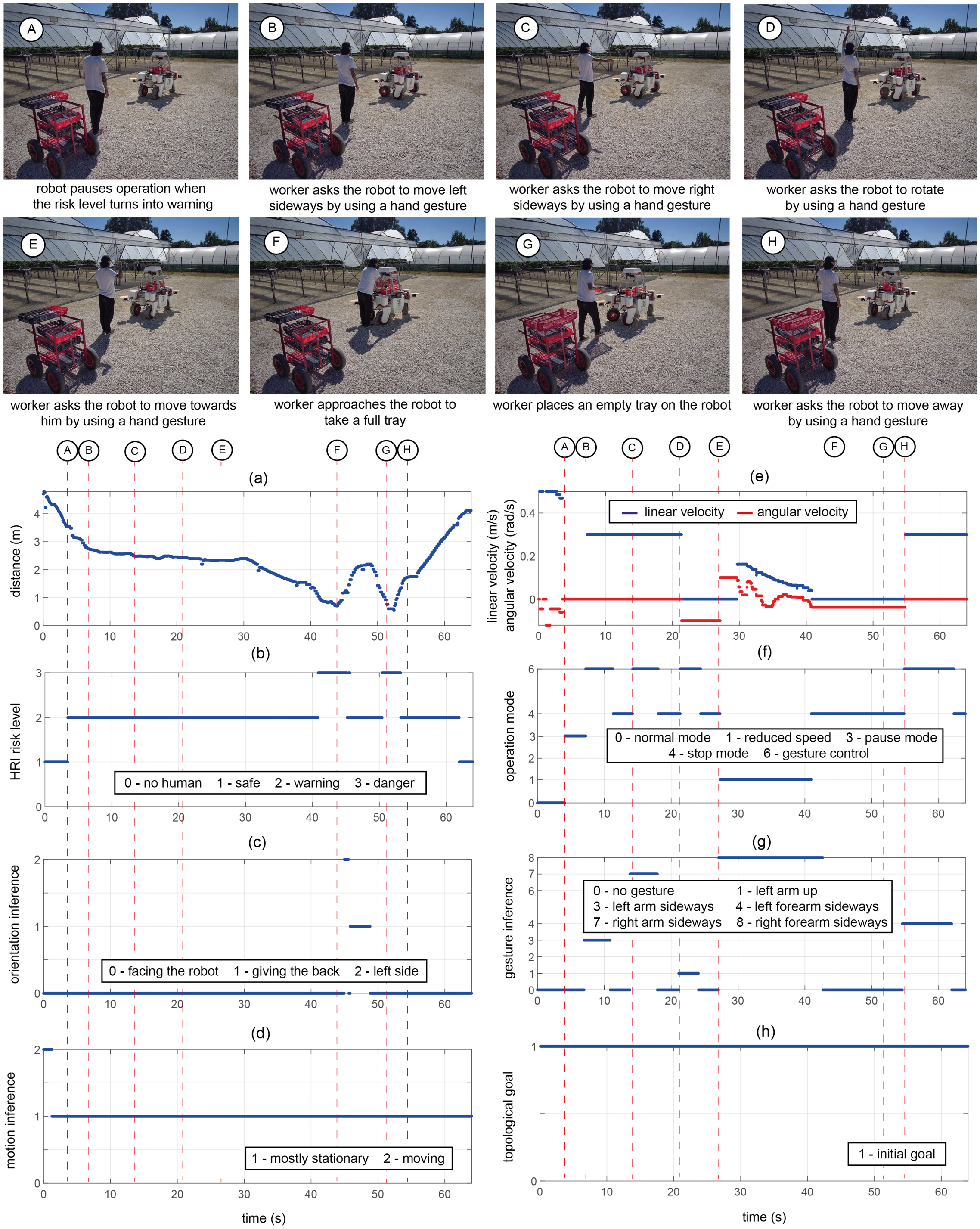


Figure 18: Snapshots from experiment 2 showing the evolution in time of parameters including: (a) distance between the robot and the picker (b) HRI risk level label (c) orientation label (d) motion label (e) robot linear and angular velocities (f) robot operation mode (g) gesture label (h) robot topological goal.

with human workers who don't have previous experience interacting with robots. Thus, in order to evaluate how safe the robot operation is perceived and how intuitive it is for new users, we performed similar experiments to the ones presented before but now with a group of 15 individuals who didn't have previous experience using the HAN module [13]. The experiments consisted of making each person interact freely with the robot (for a total of

[13] The experiments with human participants were approved by the Human Ethics Committee of the University of Lincoln (Ethics code UoL2022_9510).

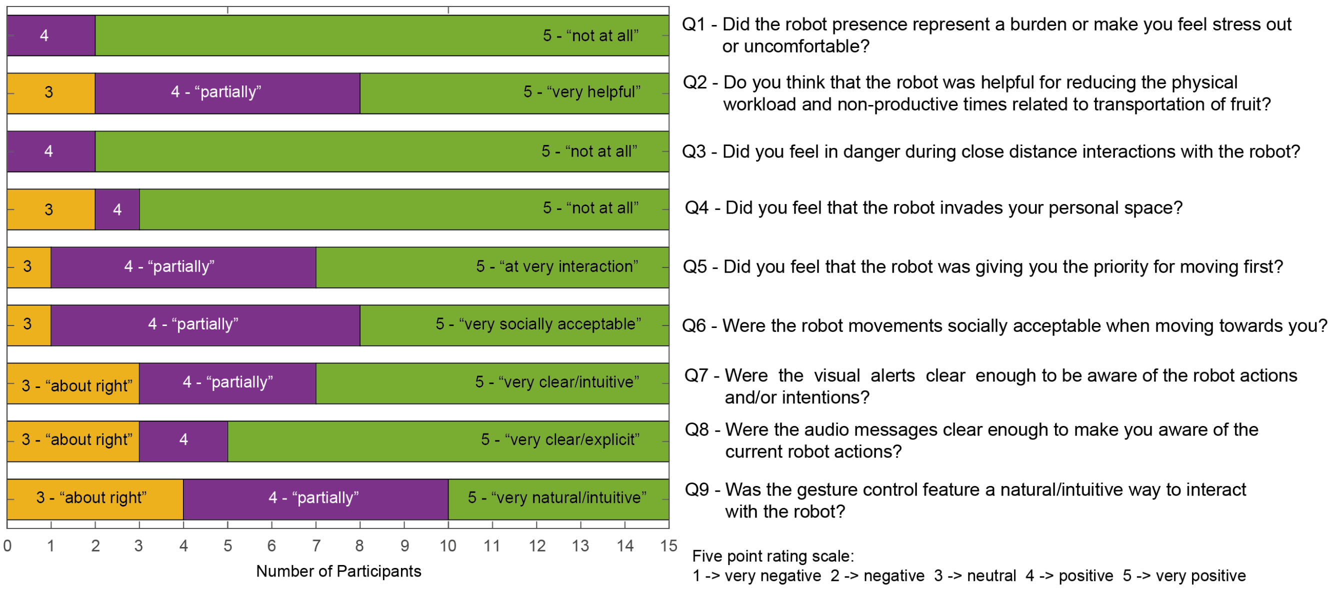


Figure 19: Outcome of questions that participants were asked (all were answered on a five-point scale).

20-30 minutes) inside and outside the polytunnels in a similar way as was illustrated in experiments 1 to 3. Before the experiments started, participants received a brief induction on how the robot operates and how to interact safely with the robot. At the end of the experiment, each person received a questionnaire to fill out. The aim of this questionnaire was to obtain metrics to evaluate the user experience in terms of: i) acceptance and suitability of co-robots in harvesting operations, ii) safety and trust perception in HRI, iii) naturalness in HRC. Thus, according to the answers given by the 15 participants involved in the experiments, we quantified the metrics mentioned previously by using a five-point rating scale as can be seen in Fig. 19.

The aim of questions *Q1*-*Q2* was to evaluate the overall level of suitability of the propose cooperative harvesting strategy. According to the majority of participants' responses, they didn't feel uncomfortable or stressed out by sharing the workspace with the robot, in fact, they thought the use of a robot for fruit transportation is very helpful for reducing their physical workload and reducing non-productive times.

Questions *Q3*-*Q6* focused on evaluating how safe and trustworthy robot operation was perceived to be by the participants. According to their responses to *Q3* and *Q4*, the HAN module successfully accomplished the goal of ensuring safe operation, not only physically (by avoiding any collision) but psychologically by stopping its operations before invading the participants' *personal* space (as was stated in *SA1*). Moreover, according to the responses to *Q5* and *Q6*, for most of the participants, the robot behavior was considered socially acceptable, by following the criteria given above such as giving priority to human during HRI (*SA2*), reducing velocity when moving towards a human (*SA3*), and only approach him/her when the human is facing the robot (*SA4*).

Finally, questions *Q7*-*Q9* aimed to evaluate how natural and intuitive the proposed audiovisual feedback alerts and *gesture control* mode are as HRC interfaces. Similar to the previous questions, the participant's responses were mostly positive (giving answers over 3) which means that they considered the audiovisual alerts clear enough to be able to understand what the robot was doing or intended to do (as was stated in *SA5*). In the same way, the proposed *gesture control* mode (which is used for getting a smooth HRI when human pickers load/unload trays), was considered by the participants intuitive enough to let them able to learn how to use it after only a quick initial training.

Figure 20 summarizes the information of the participants who were involved on these experiments. The participants' age covered the range of 18-42 years (see Fig. 20(a),) where 60% of them were male and the rest female (see Fig. 20(b)). It is important to notice that not all the participants spoke English as their primary language (see Fig. 20(c)), and that is something we expect to happen in real harvesting operations along UK were most of human labor comes from different countries. Nevertheless, according to their responses to questions *Q7*-*Q8*, the language barrier wasn't an issue since the visual indicators and the default English voice messages were designed to be as simple and clear as possible to make any person aware of the robot actions.

Then, Figures 20(d)(e) summarize the participants level of experience in terms of robotics and fruit harvesting. Overall, most of the participants had certain previous experience with robots, while only few of them were familiar with the fruit harvesting process. This lack of agricultural expertise from the participants means that it not

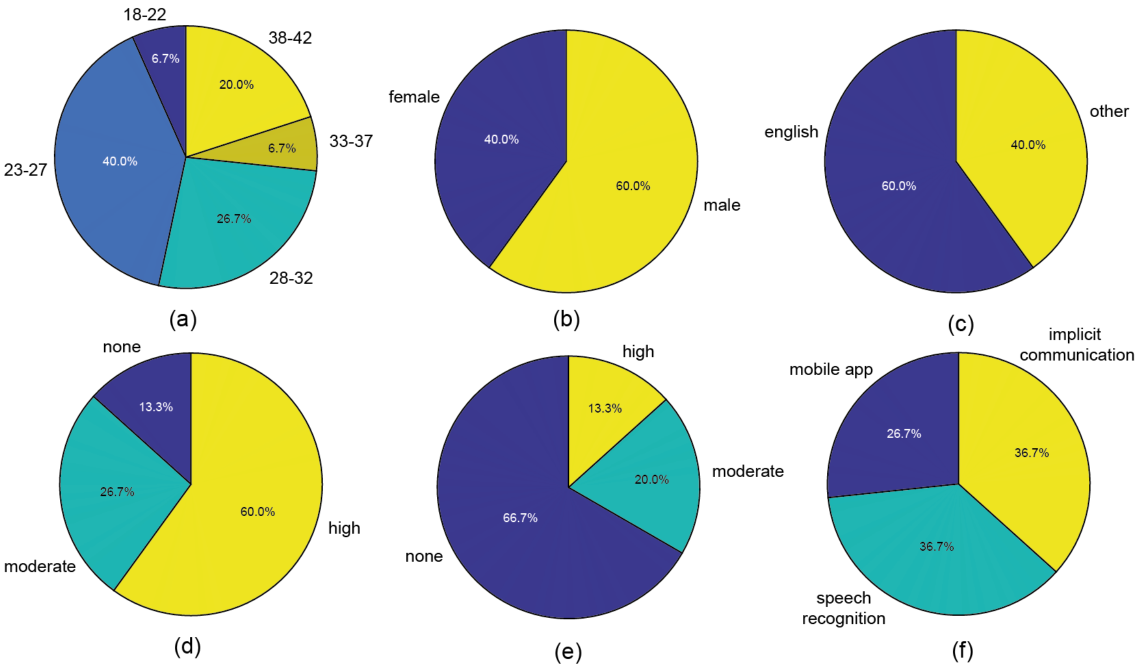


Figure 20: Summary of participants information including (a) age range (b) gender (c) primary language (d) experience level in robotics (e) experience level in fruit picking (f) HRC interface preferences.

possible to generalize the responses in *Q2*, however, it still represents a valid metric that evaluates the ergonomics of the proposed HAN module which as far as our knowledge, it is the first time that HAN is implemented and evaluated in real conditions. Some preliminary findings in terms of safety and ergonomics in cooperative harvesting operations were presented in [2], but the experiments in that study only used teleoperated robots and no HAN module or HRC interface was used to interact with the human pickers.

In fact, an intuitive HRC interface is crucial to increase the levels of safety, trust, and naturalness of any HRI. For instance, according to the participants feedback, initially, some of them were not completely comfortable with the use of hand gestures to control the robot, but as they were interacting with the robot, they commented that this feature was becoming more natural and intuitive, thus, after a couple of minutes they were completely familiar with the gesture control procedure. Additionally, as part of the user's experience evaluation, the participants were asked to choose their preference in terms of HRC interfaces among three suggested options. These options aim to complement the existent hand gesture recognition feature to make the robot capable to better interpret the human workers intentions when loading/unloading trays. The options included:

- *Mobile app*: Use of a mobile phone app to command the robot and to receive audio/vibratory feedback according to the HRI risk level.
- *Speech recognition*: Give the robot the capability to understand voice commands.
- *Implicit communication*: Give the robot the capability of interpret the human intentions (implicitly) based on my normal behavior/movements, i.e. without performing any special hand gesture or speech or using any phone app.

According to the responses of the participants (see Fig. 20(f)), implicit communication and speech recognition were chosen as the most attractive options for them. This choice was expected since they both are very natural and simple HRC interfaces that do not require any additional device (as the third option) or special training (as the hand gesture control).

References

[1] Sarah Al-Hussaini, Jason M Gregory, Yuxiang Guan, and Satyandra K Gupta. Generating alerts to assist with task assignments in human-supervised multi-robot teams operating in challenging environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11245–11252. IEEE, 2020.

[2] Paul Baxter, Grzegorz Cielniak, Marc Hanheide, and Pål From. Safe human-robot interaction in agriculture. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, page 59–60, New York, NY, USA, 2018. Association for Computing Machinery.

[3] Nicola Bellotto and Huosheng Hu. Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters. *Autonomous Robots*, 28(4):425–438, 2010.

[4] Adam Binch, Gautham P. Das, Jaime Pulido Fentanes, and Marc Hanheide. Context dependant iterative parameter optimisation for robust robot navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3937–3943, 2020.

[5] Necati Cihan Camgöz, Ahmet Alp Kindiroglu, and Lale Akarun. Gesture recognition using template based random forest classifiers. In Lourdes Agapito, Michael M. Bronstein, and Carsten Rother, editors, *Computer Vision - ECCV 2014 Workshops*, pages 579–594, Cham, 2015. Springer International Publishing.

[6] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[7] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani. A conformable force/tactile skin for physical human–robot interaction. *IEEE Robotics and Automation Letters*, 1(1):41–48, 2016.

[8] Francesco Del Duchetto and Marc Hanheide. Learning on the job: Long-term behavioural adaptation in human-robot interactions. *IEEE Robotics and Automation Letters*, 7(3):6934–6941, 2022.

[9] DS/EN 60073. Ds/en 60073 : Basic and safety principles for man-machine interface, marking and identication - coding principles for indicators and actuators, 2003.

[10] Chad Edwards, Autumn Edwards, Brett Stoll, Xialing Lin, and Noelle Massey. Evaluations of an artificial intelligence instructor's voice: Social identity theory in human-robot interactions. *Computers in Human Behavior*, 90:357–362, 2019.

[11] Tingxiang Fan, Pinxin Long, Wenxi Liu, and Jia Pan. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *The International Journal of Robotics Research*, 39(7):856–892, 2020.

[12] Andrzej Felski, Krzysztof Jaskólski, Karolina Zwolak, and Paweł Piskur. Analysis of satellite compass error's spectrum. *Sensors*, 20(15):4067, 2020.

[13] Lars Grimstad and Pål J. From. Software Components of the Thorvald II Modular Robot. *Modeling, Identification and Control*, 39(3):157–165, 2018.

[14] Lars Grimstad and Pål Johan From. The thorvald ii agricultural robotic system. *Robotics*, 6(4), 2017.

[15] A. M. Hall and X. Jin. Integrated control of strawberry powdery mildew. *Acta Horticulturae*, 1156:771–776, 2017.

[16] Yew Cheong Hou, Khairul Salleh Mohamed Sahari, Leong Yeng Weng, Hong Kah Foo, Nur Aira Abd Rahman, Nurul Anis Atikah, and Raad Z Homod. Development of collision avoidance system for multiple autonomous mobile robots. *International Journal of Advanced Robotic Systems*, 17(4):1729881420923967, 2020.

[17] Md Jahidul Islam, Jungseok Hong, and Junaed Sattar. Person-following by autonomous robots: A categorical overview. *The International Journal of Robotics Research*, 38(14):1581–1618, 2019.

[18] Dan Jia, Alexander Hermans, and Bastian Leibe. DR-SPAAM: A Spatial-Attention and Auto-regressive Model for Person Detection in 2D Range Data. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[19] Muhammad W. Khan, Gautham P. Das, Marc Hanheide, and Grzegorz Cielniak. Incorporating spatial constraints into a bayesian tracking framework for improved localisation in agricultural environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2440–2445, 2020.

[20] Muhammad W Khan, Gautham P Das, Marc Hanheide, Grzegorz Cielniak, et al. Incorporating spatial constraints into a Bayesian tracking framework for improved localisation in agricultural environments. 2020.

[21] Marta Kwiatkowska, Gethin Norman, David Parker, and Gabriel Santos. Prism-games 3.0: Stochastic game verification with concurrency, equilibria and time. In *International Conference on Computer Aided Verification*, pages 475–487. Springer, 2020.

[22] Bruno Lacerda, Fatma Faruq, David Parker, and Nick Hawes. Probabilistic planning with formal performance guarantees for mobile service robots. *The International Journal of Robotics Research*, 38(9):1098–1123, 2019.

[23] Tuan D. Le, Vignesh R. Ponnambalam, Jon G. O. Gjevestad, and Pål J. From. A low-cost and efficient autonomous row-following robot for food production in polytunnels. *Journal of Field Robotics*, 37(2):309–321, 2020.

[24] Qingzhu Liang, Yinghao Yang, Hang Zhang, Changhong Peng, and Jianchao Lu. Analysis of simplification in Markov state-based models for reliability assessment of complex safety systems. *Reliability Engineering & System Safety*, 221:108373, 2022.

[25] RM Lucas, S Yazar, AR Young, M Norval, FR De Gruijl, Y Takizawa, LE Rhodes, CA Sinclair, and RE Neale. Human health in relation to exposure to solar ultraviolet radiation under changing stratospheric ozone and climate. *Photochemical & Photobiological Sciences*, 18(3):641–680, 2019.

[26] Simon Parsons. The strawberry growing environment at clock house farm. Technical report, University of Lincoln, January 2021.

[27] Riccardo Polvara, Francesco Del Duchetto, Gerhard Neumann, and Marc Hanheide. Navigate-and-seek: a robotics framework for people localization in agricultural environments. *IEEE Robotics and Automation Letters*, 6(4):6577–6584, 2021.

[28] Xuan-Tung Truong, Voo Nyuk Yoong, and Trung-Dung Ngo. Dynamic social zone for human safety in human-robot shared workspaces. In *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 391–396, 2014.

[29] Vaibhav V Unhelkar, Przemyslaw A Lasota, Quirin Tyroller, Rares-Darius Buhai, Laurie Marceau, Barbara Deml, and Julie A Shah. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and Automation Letters*, 3(3):2394–2401, 2018.

[30] J.P. Vasconez, H. Admoni, and F. Auat Cheein. A methodology for semantic action recognition based on pose and human-object interaction in avocado harvesting processes. *Computers and Electronics in Agriculture*, 184:106057, 2021.

[31] Juan Pablo Vasconez, Leonardo Guevara, and Fernando Auat Cheein. Social robot navigation based on hri non-verbal communication: A case study on avocado harvesting. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 957–960, New York, NY, USA, 2019. Association for Computing Machinery.

[32] Roger Woodman, Alan F.T. Winfield, Chris Harper, and Mike Fraser. Building safer robots: Safety driven control. *The International Journal of Robotics Research*, 31(13):1603–1626, 2012.

Appendices

Table 10: List of variables used to model the agricultural tasks.

Variable Name	Initial Value	Possible Values	Description	Transitions					
				Previous values			Next values		
				UV-C	Logistics	Picking	UV-C	Logistics	Picking
x_uvc	0	0	Robot at the shed	3	-	-	1	-	-
		1	Robot moving from the shed to polytunnel	0	-	-	2	-	-
		2	Robot performing the UV-C treatment	1	-	-	3	-	-
		3	Robot moving from polytunnel to the shed	2	-	-	0	-	-
x_logistics x_picking	0	0	Robot at the shed	-	8		-	1	
		1	Robot moving from the shed to collection point to place empty trays on it	-	0		-	2	
		2	Robot interacting with the worker who load empty trays on it	-	1		-	3	
		3	Robot moving from collection point to polytunnel	-	2,7		-	4	
		4	Robot picking fruits along the rows or only moving inside the polytunnel	-	3,5	3	-	5	6
		5	Robot interacting with the worker who summoned it inside the polytunnel	-	4	-	-	4,6	-
		6	Robot moving from polytunnel to collection point to unload full trays	-	5	4	-	7	
		7	Robot interacting with the worker who unload full trays	-	6		-	8	
		8	Robot moving from collection point to robot shed	-	7		-	0	
x_runs	0	0,1,2,...	Number of times that the robot performed a two-way trip from polytunnel to the collection point	-	x_runs-1		-	x_runs+1	
		N_runs	The robot completed the maximum number of runs before the robot battery requires to be recharged	-	N_runs-1		-	0	
x_seg	0	0,1,2,...	Number of footpath segments traversed by the robot before reaching the goal point	N_seg-1			N_seg+1		
		N_seg_shed- N_seg_collect	The robot traversed all the footpaths segments between the shed and the collection point, assuming that N_seg_shed >N_seg_collect	N_seg_shed- N_seg_collect-1			N_seg_shed- N_seg_collect+1	0	
		N_seg_shed	Robot traversed all the footpaths segments between the shed and the polytunnel	N_seg_shed-1			0		
		N_seg_collect	Robot traversed all the footpaths segments between the shed and the polytunnel	N_seg_collect-1			N_seg_collect+1	0	
x_rows	0	0,1,2,...	Number of rows traversed by the robot	N_rows-1	-		N_rows+1	-	
		N_rows	Robot covered all the rows in the polytunnel	N_rows-1			0		
x_trays	0	0,1,2,...	Number of times the robot replaced the trays	-	N_trays-1		-	N_trays+1	
		N_trays	All the trays on the robot are full of fruit		N_trays-1			0	

Table 11: List of constants used to define the scale of the agricultural tasks.

Constant Name	Possible Values	Description	Related to Variable
N_rows	1,2,3,...	Number of rows that the robot is able to cover before it has to come back to the robot shed to charge the battery	x_rows x_uvc
N_seg_shed	1,2,3,...	Number of footpath segments required to be traversed when moving from robot shed to the polytunnel	x_seg x_uvc x_logistics x_picking
N_seg_collect	1,2,3,...	Number of footpath segments required to be traversed when moving from collection point to the polytunnel	x_seg x_logistics x_picking
N_runs	1,2,3,...	Number of times a robot can perform two-way trips from polytunnel to collection point before it has to come back to robot shed to charge the battery	x_runs x_logistics x_picking
N_trays	1,2,3,...	Number of times a robot can place full trays on it before it has to come back to collection point to replace with empty trays	x_trays x_logistics x_picking

Table 12: List of values that the variable x_robot can take according to the agricultural scenario.

Variable name	Initial Value	Possible Values	Description	Transitions					
				Previous values			Next values		
				UV-C	Logistics	Picking	UV-C	Logistics	Picking
x_robot	0	0	Robot operation is paused (starting mode at the shed)	1	1	1	1	1	1
		1	Robot moving along footpaths	2	2	2	2,3,10	2,3,8,10	
		2	Robot performing a transition between footpath segments	1,9,10			1,3,10		
		3	Robot evading a human at footpaths	1,2			9,10		
		4	Robot moving along the row performing UV-C treatment	7	-	-	7,10	-	-
		5	Robot moving along a row transporting trays	-	7	-	-	7,8,10	-
		6	Robot moving along a row while picking fruits	-	-	7	-	-	7,10
		7	Robot performing a transition between rows	4,10	5,9,10	6,10	4,10	5,10	6,10
		8	Robot approaching to the worker position (reducing speed)	-	1,5		-	9,10	
		9	Robot moving away from the worker position	3	3,8		2,10	2,7,10	2,10
		10	Robot stops because of safety purposes	1,2,...,8,9			2,7		

Table 13: List of variables used to model the safety system.

Variable Name	Initial Value	Possible Values	Description	Transitions	
				Previous values	Next values
x_hds	0	0	No human detected	1	1,2,3,4
		1	Human detected when $d > 7m$	0,2	0,2,3,4
		2	Human detected when $3.6m \leq d \leq 7m$	0,1,3	1,3
		3	Human detected when $1.2m < d < 3.6m$	0,1,2,4	2,4
		4	Human detected when $0m < d \leq 1.2m$	0,1,2,3	3
x_htmis	0	0	No human tracked	1,2	1*,2*
		1	Accurate human motion inference	0	0
		2	Not reliable human motion inference	0	0
x_hars	0	0	No human gesture detected	1,2	1*,2*
		1	Correct human gesture/action recognition	0	0
		2	Wrong human gesture/action recognition	0	0
x_scs	0	0	No contact	1,2	1*,2*
		1	Collision is detected	0	0
		2	Collision is not detected on time	0	0
x_visual x_voice	0	0	Audiovisual alerts are not activated	1,2	1,2*
		1	Audiovisual indicators are activated when a human is detected	0,2	0
		2	Periodic audiovisual indicators are activated (even when there is not a human detected)	0	0,1

* : non-deterministic transition

Table 14: List of constants used to define the probability of failure of each safety system component.

Constant Name	Possible Values	Cases evaluated			Description	Related to Variable
		ideal	regular	worst		
p_alerts	[0,1]	0.9	0.7	0.5	Probability that at this specific moment a periodic audiovisual alert is activated to warn nearby human about danger	x_visual x_voice
p_scs	[0,1]	0.1	0.3	0.5	Probability that the robot contact sensors fails to detect a collision	x_hazard
p_hds_1	[0,1]	0.3	0.4	0.5	Probability that HDS fails in detect a human in the same row farther than 7m	x_hds
p_hds_2	[0,1]	0.2	0.3	0.4	Probability that HDS fails in detect on time a human at 7m in the same row	
p_hds_3	[0,1]	0.1	0.2	0.3	Probability that HDS fails in detect on time a human at 3.6m in the same row	
p_hds_4	[0,1]	0.1	0.1	0.2	Probability that HDS fails in detect on time a human at 1.2m in the same row	
p_hds_5	[0,1]	0.3	0.4	0.5	Probability that HDS fails in detect a human at the end of the rows farther 7m when the robot is going to perform row transitions	
p_hds_6	[0,1]	0.2	0.3	0.4	Probability that HDS fails in detect on time a human at the end of the row at 7m when the robot is going to perform row transitions	
p_hds_7	[0,1]	0.1	0.2	0.3	Probability that HDS fails in detect on time a human at the end of the row at 3.6m when the robot is going to perform row transitions	
p_hds_8	[0,1]	0.1	0.1	0.2	Probability that HDS fails in detect on time a human at the end of the row at 1.2m when the robot is going to perform row transitions	
p_hds_9	[0,1]	0.2	0.3	0.4	Probability that HDS fails in detect a human on time in the same footpath above 3.6m	
p_hds_10	[0,1]	0.1	0.2	0.3	Probability that HDS fails in detect a human on time in the same footpath at 3.6m	
p_hds_11	[0,1]	0.1	0.1	0.2	Probability that HDS fails in detect a human on time in the same footpath at 1.2m	
p_htmis_1	[0,1]	0.1	0.2	0.3	Probability that HTMIS fails to track accurately a human in the same row	x_htmis
p_htmis_2	[0,1]	0.1	0.2	0.3	Probability that HTMIS fails to track accurately a human in the same footpath	
p_hars	[0,1]	0.1	0.2	0.3	Probability that the HARS fails to detect the correct hand gesture	x_hars

Table 15: List of variables used to model the human behavior.

Variable Name	Initial Value	Possible Values	Description	Transitions	
		Possible Values		Previous values	Next values
x_human	0	0	No human presence	1,2	1*,2*
		1	Untrained human is interacting with the robot	0	0
		2	Trained worker is interacting with the robot	0	0
x_motion	0	0	Human stays stationary	0,1,2	1,2*,3*,4*
		1	Human is moving to the robot position	0	0*,2*
		2	Human is moving away from the robot position	0,1,3,4	0
		3	Human approaching to the robot to place trays on it	0	2
		4	Human walking next to the robot along the footpath	0	2
x_aware	0	0	Human is not aware of robot presence or potential danger of approaching to the robot	1	1*
		1	Human is aware of robot intentions or danger of approaching to the robot	0	0
x_gesture	0	0	Human is not performing any specific hand gesture	1	1*
		1	Human performing a hand gesture to make the robot knows about his/her intentions	0	0
x_dist	0	0	$d >> 7m$	1	1
		1	$d > 7m$	0,2	0,2
		2	$3.6m \leq d \leq 7m$	1,3	1,3
		3	$1.2m < d < 3.6m$	2,4	2,4
		4	$0m < d \leq 1.2m$	3	3

* : non-deterministic transition

Table 16: List of constants used to model the non-deterministic human behavior.

Constant Name	Possible Values	Cases evaluated			Description	Related to Variable
		ideal	regular	worst		
p_int_1	[0,1]	0,0.1,0.2,...,0.9,1			Probability that an unplanned interaction is going to happen inside the polytunnels with an untrained person	x_human
p_int_2	[0,1]	0,0.1,0.2,...,0.9,1			Probability that an unplanned interaction is going to happen inside the polytunnels with an trained worker	
p_int_3	[0,1]	0,0.1,0.2,...,0.9,1			Probability that an unplanned interaction is going to happen outside the polytunnels with an untrained person	
p_int_4	[0,1]	0,0.1,0.2,...,0.9,1			Probability that an unplanned interaction is going to happen outside the polytunnels with an trained worker	
p_aware_1	[0,1]	0.8	0.7	0.5	Probability that visual alerts are correctly interpreted by untrained people	x_aware
p_aware_2	[0,1]	0.9	0.8	0.5	Probability that visual alerts are correctly interpreted by trained people	
p_aware_3	[0,1]	0.9	0.9	0.5	Probability that voice alerts are correctly interpreted by trained and untrained people	
p_decision	[0,1]	0.9	0.8	0.5	Probability that the human decides to perform a risky movement or not	x_motion
p_reply	[0,1]	0.9	0.8	0.5	Probability that the human performs or not a hand gesture to make the robot knows about his/her intentions	x_gesture