

A Unified Topological Representation for Robotic Fleets in Agricultural Applications

Gautham Das, Grzegorz Cielniak, James Heselden and Simon Pearson

*

Lincoln Institute for Agri-food Technology (LIAT)
University of Lincoln, Riseholme Campus
Lincoln LN2 2LG, United Kingdom
{gdas, gcielniak, jheselden, spearson}@lincoln.ac.uk

Francesco Del Duchetto, Zuyuan Zhu, Johann Dichtl, Marc Hanheide

Lincoln Centre for Autonomous Systems (L-CAS)
University of Lincoln, Brayford Pool Campus
Lincoln LN6 7TS, United Kingdom
{fdelduchetto, zzhu, jdichtl, mhanheide}@lincoln.ac.uk

Jaime Pulido Fentanes, Adam Binch, Michael Hutchinson and Pal From

Saga Robotics
Think Tank, Ruston Way
Lincoln, LN6 7FL, United Kingdom
{jpfentanes, abinch, mhutchinson, pfrom}@sagarobotics.com

Abstract

Agricultural robots offer a viable solution to the critical challenges of productivity and sustainability of modern agriculture. The widespread deployment of agricultural robotic fleets, however, is still hindered by the overall system's complexity, requiring the integration of several non-trivial components for the operation of each robot but also the orchestration of robots working with each other and human workers. This paper proposes a topological map as the unifying representation and computational model to facilitate the smooth deployment of robotic fleets in agriculture. This topological abstraction of the system state results in an efficient representation of large-scale environments, but also offers the scalable and efficient operation of the entire fleet and allows for ex-situ modelling and analysis of operations. The practical use of the proposed framework is demonstrated in a horticultural use case with a fleet of robots supporting the work of human fruit pickers. The critical components of the system are analysed and evaluated in deployment in both realistic digital twin and real-life soft fruit farms of different scales, demonstrating the scalability and effectiveness of the proposed framework. The presented framework is general and should be easy to adopt in other multi-robot/multi-human scenarios such as warehouse logistics, cleaning and maintenance of public spaces.

*Das, Cielniak and Heselden are also with Lincoln Centre for Autonomous Systems (L-CAS), University of Lincoln, Lincoln LN6 7TS, United Kingdom.

1 Introduction

Key challenges in modern global agriculture include the need to drive productivity and sustainability of intensified food production. Increasingly, farming systems are having to respond to labour supply constraints, brought on by socio-political factors such as ageing populations or restrictions on migration. Agricultural robotics offers a viable solution to these problems. Opportunities include the development of lightweight and intelligent farming robots (e.g. (Grimstad and From, 2017)) which might achieve the intensification of food production through autonomous 24/7 operation or deployment within self-coordinating robotic fleets. Due to intelligent sensing, autonomy and learning functionality, robots can work with and manipulate individual plants, support and interact with human workers or selectively apply non-chemical treatments. These deliver reduced production costs, more friendly working environments, reduced contamination risks, etc. (Pedersen et al., 2006). The opportunities for robotic automation are substantial for soft fruit production since it involves many manual operations such as plant care, harvesting and transportation of the crop to storage facilities. For example, an automated solution for in-field transportation on its own could save up to 20% on labour costs and 10% on land usage (Baxter et al., 2018). To secure these gains, our research aims to develop enabling technologies for the deployment of autonomous fleets of robots that support and complement human workers.

The complex nature of agricultural tasks poses serious technical challenges for the smooth deployment of robotic fleets. This includes the integration of many non-trivial components for navigation, modelling, coordination and planning each with a distinct set of constraints and requirements, but also the orchestration of operations of individual robots interacting with each other as well as human workers. In this paper, we propose a topological map as the unifying representation and computational model to facilitate both the orchestration of a fleet of otherwise largely decoupled individual robots and their respective coordination and interaction with human workers. While individual robots in heterogeneous fleets will have specific implementations owing to their kinematic design, sensor equipment, navigation constraints, and respective derived tasks and abilities, the abstraction of the current overall system operational state into a topological representation will be shown as being sufficient to facilitate both the coordination and management of such a fleet, as well as the modelling of overall system behaviour to allow for ex-situ modelling and analysis of operations. In addition, the topological representation enables efficient state representation of large-scale environments and offers the scalable and efficient operation of the entire fleet.

To demonstrate the benefits of the proposed representations, we introduce a validation use-case in which a fleet of autonomous robots supports human pickers for in-field transportation tasks (see Sec. 4.1). The central management tool is the *fleet coordinator*, which uses the topological representation of the environment to manage task allocation and coordination for all robots operating in the area and ensures that there are no deadlocks caused by moving robots. The system can also use the topology to keep track of humans e.g. by using smart trolleys that provide position updates to the fleet coordinator as explained in Sec. 4.2. The system employs the proposed topology also for modelling, here realised through Discrete Event Simulations, for future state prediction of both robot and human agents which is used for planning and system optimisation in simulated scenarios.

The main contribution of the paper is the proposition and evaluation of a topological map as a unifying representation to

- use as an orchestration principle to build integrated agricultural multi-robotic systems;
- represent the *probabilistic state* of a multi-agent (humans and robots) integrated system, incorporating the respective constraints of the environment;
- facilitate coordination of robots in tasks supporting human workers;
- upscale the analysis and modelling of a multi-agent system utilising probabilistic models in a digital twin of the soft fruit production farm;

- demonstrate its practical use through real-life deployment in real soft fruit farm environments of different scales.

2 Related Work

Fully autonomous robots have been successfully deployed into agricultural settings, so far mainly in prototypical proof of concepts. With this paper’s main contributions focusing on the benefits of topological representations, this section offers a brief overview of previous works on robotic applications in horticulture and the wider state of the art of topological representations in robotics.

2.1 Robotic Platforms for Strawberry Production

The attempts to create robotic harvesters date back to the mid-’80s (Bachche, 2015), whilst early examples of strawberry-picking robots were documented in (Kondo et al., 1998), (Feng et al., 2008) and (Rajendra et al., 2009). Since then, there were more than 40 reports on strawberry harvesting robots (Gozde Defterli et al., 2016) developed in different parts of the world. Despite the tremendous technological progress achieved through the prior work, and a number of startups targeting robotic strawberry production (Saga Robotics, Dogtooth, Agrobot, Octinion), there is currently no commercial solution available for the soft fruit industry that would be economically viable. Many of the previous reviews of agricultural robotics attempted to define barriers which prevent wider adoption of this technology (e.g. (Gozde Defterli et al., 2016), (Bac et al., 2014)). Whilst improvements of individual components and their integration, economic and logistical aspects were identified, the general autonomy, so the ability for the robots to operate safely and efficiently in the presence and collaboration with people in agricultural environments, is still a subject of active study (Duckett et al., 2018).

A successful robotic strawberry production system should consider all stages of production including crop care, harvesting, quality assurance and logistics. The in-field logistics solution, studied in this work, is particularly suitable for robotic deployment as it can optimise the working time of human pickers, reduce requirements for additional transportation infrastructure and shorten the time between picking and storing of the crop in the cold store. Such a system was initially proposed in our previous work which considered the problems of area coverage and routing problems to optimise the fleet operation (Das et al., 2018) and the work presented in this paper, stems from these initial investigations. A similar in-field transportation system for strawberries was independently proposed and developed by the team from UC Davis with a technical focus on dynamic scheduling (Peng and Vougioukas, 2020; Seyyedhasani et al., 2020b), modelling and simulation (Seyyedhasani et al., 2020a). This harvesting-aid system was demonstrated in open fields and utilised unobstructed GNSS signals for the localisation of transportation carts and humans. Any potential localisation errors in such situations would have minimal negative effects on the operation of the fleet as human pickers could simply cross the strawberry rows to reach the cart. In polytunnels, the GNSS signal is unreliable due to the metallic structure, but also localisation errors can direct the robots into the wrong row which requires costly additional manoeuvring either by the robots or humans (see Fig. 3). To deal with such challenges, the system presented in this work utilises topological structure as an underlying representation which allows for integrating individual components for localisation, navigation, safe operation and fleet management but also, through the integration of spatial constraints and fusion of information from multiple sources, to increase the system’s robustness to localisation errors.

2.2 Topological Representations

Topological representations are often a key enabler for building real-world robotic systems, as an underpinning concept to robot navigation and fleet coordination. Transient topological representation is a commonly used approach to create simple paths for a robot to navigate. This is often facilitated through a simple user interface such as in (ClearpathRobotics, 2022), to improve accessibility to using the robots. In this method,

the navigation is represented topologically where the robot is tasked with navigating to each node/edge along a defined path, however, the environment is not topologically represented as a whole.

Fundamentally, the topological representation of an environment, which is the primary discussion for this paper, is an explicit collection of nodes and edges used to describe the regions an agent is able to traverse. However, utilising such systems is difficult as very few practical technologies exist. Four such practical technologies tailored for the use cases are OSM Cartography Map (OpenStreetMap-ROS, 2021), Open RMF (open rmf, 2023), ANYbotics topological map (ANYbotics, 2023) and NavGraph (FawkesRobotics, 2018).

In (OpenStreetMap-ROS, 2021), the system utilises information from Open Street Map, an open geographic database, to construct pseudo-topological descriptions of the environment saved in an XML format. In this approach, nodes are defined independently, and then collections of node references are used to construct more complex systems. These collections, called Ways, can include tags that hold additional information such as a road name, if a highway is one-way, if there are any junctions, etc. Way's can also be used to describe objects such as buildings or coastlines. The OSM system also includes a Relation tag, which can be used to group references of Ways to describe larger constructs such as larger highways, marinas, or islets. While effective at describing large-scale environments, it has a lot of complexity which is unnecessary for robotic deployments in smaller environments.

In (open rmf, 2023), topological maps are defined as part of an all-encompassing collective description of the environment, in which nodes and edges are described as vertices and lanes, alongside; human-lanes, doors, floors, walls, lifts and drawings in a single YAML file. This file also includes all information on crowd simulations and robots to spawn in simulated environments. The vertices themselves do not hold too much information, listing only their position alongside details of any specific attributes such as it holding a charger or being a pre-approved parking spot. Lanes are similarly simplified, containing required information on their position, orientation, width, bidirectionality, and most importantly their graph index, which is used for connecting lanes for route planning. The OpenRMF maps are effective at fully describing their environments, however, the utility of these maps also requires a dependency on the rest of the OpenRMF ecosystem.

In (ANYbotics, 2023), their topological map is constructed within a simulation generated from CAD files of the operation environment. Nodes and edges are kept simple, allowing the robot's own autonomy to make decisions on traversal methods. Their mapping system also fuses additional sensing points alongside the topological map for data collection such as locations of meters to gather readings from, locations to analyse temperature levels etc.

In (FawkesRobotics, 2018), a map representation named as NavGraph is described as "topological graphs used for path planning as well as storing location-specific semantic information related to the ground map". For each node, properties such as the orientation and goal tolerances can be defined to specify the maximum distance allowed for the robot to reach the location. Uniquely, NavGraph also includes a shortcut tolerance, in which the robot checks all upcoming nodes to look for potential shortcuts in its route, skipping nodes where possible and providing smoother navigation. NavGraph, however, does not offer any facility to describe information on the relation of how nodes are connected beyond the existence of a connection and a tag identifying it as an intersection.

Topological representations of the environment have been used in other works such as introduced in (Blanco et al., 2008), one of the earlier works on hybrid topo-metric maps in robotics, which can be seen as one of the foundations of the work we present. Probabilistic topological mapping is proposed in (Ranganathan and Dellaert, 2011), which somewhat underpins our concept of using topology as an underlying probabilistic model for state representation. In (Hanheide et al., 2017), topological maps are considered dynamic descriptions of the environment, expanding in a goal-directed manner through open-ended planning in object search to discretise planning space for general integrated route and mission planning. Topological maps are used as an abstraction for probabilistic mission planning in (Lacerda et al., 2019; Street et al., 2022), where edges are equipped with information on traversal time and traversal success probability to provide temporal guarantees for individual missions in uncertain environments. In (Street et al., 2022), this is also applied to

agricultural settings similar to ours for fleet coordination.

One key aspect also considered in this paper is the ability to use topological representations to include *constraints* into navigation frameworks, also focusing on planning methods that efficiently perform agricultural tasks taking into account both the robot’s motion constraints and those of the environment (Binch et al., 2020; Bakker et al., 2011). The work in (Jensen et al., 2012) demonstrated how the task constraints in scenarios with multiple and heterogeneous systems, like automated combine harvesters and infield transportation systems could be incorporated. In all these scenarios, however, the constraints were pre-defined by a priori information such as geo-located obstacles, crop rows, or field limits. In our representation, such a priori restrictions are set as static constraints and combinations of the states of other agents in the neighbourhood forming dynamic constraints. Essentially, these constraints enable our topological representation to be the abstraction to underpin the operation of a fleet of heterogeneous robots and agents.

2.3 Fleet Coordination

Agricultural and food logistics have an important role in ensuring food quality and the overall health of the personnel involved (Gebresenbet and Bosona, 2012). Fleets of autonomous transportation robots have been predominantly used in warehouse applications (Wurman et al., 2008) with many recent attempts in precision agriculture (Grimstad et al., 2015; Ball et al., 2017). Many of the existing works in in-field agricultural robotics focus on multi-robot path planning (Bochtis et al., 2010b; Conesa-Muñoz et al., 2015), and very few have looked at collaborative interactions between humans and robots. Planning, scheduling and coordination are fundamental to the control of multi-robot heterogeneous systems on the farm (Bochtis et al., 2010a). Example applications include in-field logistics, where vehicles need to be scheduled for area coverage and routing problems (Das et al., 2018). Aspects of swarm robotics could potentially be applied to fleet management systems in agriculture, as in the EU-funded ECHORD++ projects SAGA and MAR.

Fleet coordination could be achieved through centralised, decentralised and distributed decision-making systems, widely discussed in the literature on multi-robot task allocation (MRTA). Centralised approaches, although suffer from a lack of robustness, achieve global visibility of agent states, and optimal management of fleet operations. Decentralised and distributed approaches are often proposed to improve the robustness. However, in our validation use case, we deploy a centralised fleet coordinator so as to keep the focus on the topological abstractions and how they can underpin complex fleet operational management while making optimal decisions.

The optimisation of robotic fleet size for an operating environment (Das et al., 2018), or the optimisation of the environment for efficient deployment of a robotic fleet (Zhu et al., 2023) are important decisions in assessing the economical viability of agricultural robots. These factors influence the design and development of a fleet coordination system. A direct approach to achieve these would involve simulating continuous-time models of the processes. This, however, could result in long simulation runtimes when the state of the process does not change for a considerable amount of time. An alternative approach is to discretise the processes and run a discrete event simulation (DES) looking at the changes between the important states that are of interest. The discrete representations in our topological representations show a direct appropriateness to DES enabling quick system optimisations (Das et al., 2018). In this paper, we showcase the use of DES to investigate modifications to the topological representations for efficient robotic fleet traffic management.

2.4 Localisation of Human Workers

Detection, identification, localisation and tracking of objects are important subjects studied in the robotics community (Glas et al., 2009; Glas et al., 2015). Some systems have achieved sub-meter level accuracy but at the expense of high-cost sensors like LiDARs, high-definition cameras, RTK-supported GNSS or different combinations of the sensors.

One of the popular sensing approaches towards localisation relies on colour video streams. A colour camera can provide several local and temporal features, however, it fails to provide 3D location in the real world (Minyoung Kim et al., 2008; Wenmiao Lu and Yap-Peng Tan, 2001; Park et al., 2013) and thus has a limited range of applications. Thus, depth cameras have received considerable attention since 2010 with the release of Microsoft Kinect. In (Gritti et al., 2014), the authors proposed an algorithm for the Kinect-equipped small-footprint robot which employs a legs classifier to detect and track the legs of multiple people in a highly cluttered indoor environment. Sensors like lasers, cameras and RFIDs fail to provide the identity of localised objects, although some studies such as (Bellotto and Hu, 2007; Bellotto et al., 2018) developed algorithms that could identify and track human subjects simultaneously. This type of identification is system-generated and can only associate a track with a system-generated identifier. In (Li et al., 2017), the authors used a 3D camera (Kinect V2) and a passive RFID (Impinj R420) to simultaneously track and identify humans. In their work, the R420’s antenna behaved as a reference to match the RFID with the skeletons detected by the depth camera. Though a centimetre-level accuracy was achieved, the inability of Kinect V2 to detect more than six skeletons at once, confined the system’s capacity to six tracks only in a relatively short range.

In (Yan et al., 2017), an online learning framework taking advantage of multi-target tracking using 3D-LiDAR was presented. The framework detected, identified and tracked human subjects in real-time. However, as is the case with any homogeneous laser-based system, the tracker was liable for degraded accuracy due to false positives and negatives. In (Imperoli et al., 2018), a robot self-localisation scheme was presented which fused several heterogeneous sensors and leveraged their strengths by adding constraints on the shortcomings of each sensor. Using different sensor setups, the authors were able to achieve from 37% to 76% accuracy gains. In (Bellotto and Hu, 2010), multi-sensor human tracking with a mobile robot using different variants of the Kalman and particle filters was presented. The authors demonstrated that in terms of accuracy, the particle filter outperformed the two variants of the Kalman filter but at the expense of computational load. Another comparison of Bayesian trackers, exploiting range estimation via received signal strength of a radio signal was studied in (Khan et al., 2015). In this work, the accuracy of the particle filter was compared with and outperformed the extended and conventional Kalman filters. Most of these techniques require the tracked entity to be in the vicinity of the sensor providing the measurements. In applications where continuous human identification and tracking are required over a very large coverage area, these sensors fail to provide a reliable solution. In such scenarios in outdoor environments, the use of GNSS-based service is still the most sensible option.

2.5 Summary

Topological representations have been used widely in the robotics community in general and also in agricultural settings in particular, for specific aspects of state representation, planning, system engineering, and coordination. However, aspects of *scalability* to large application domains like agriculture are still limited, and in our work, we bring many of these individual aspects together into one concept of closely integrated state representation and estimation, and reasoning and coordination upon the very same formalism. We argue, that topological maps, integrating transition and navigation constraints, provide an ideal abstraction for logistical support in agricultural settings.

3 Topological Abstraction for Decomposition, State Representation, Process Modelling, and Coordination

Building integrated and deployable robotic systems is a challenge of software engineering in its own right. In particular, when it comes to supporting the deployment of heterogeneous fleets of robots with different form factors and abilities in a scalable way, working in dozens of hectares of farmlands, one needs to consider the reliability of individual robots of such a fleet, their probability of failure in executing tasks, and their orchestration and coordination in a principled way.

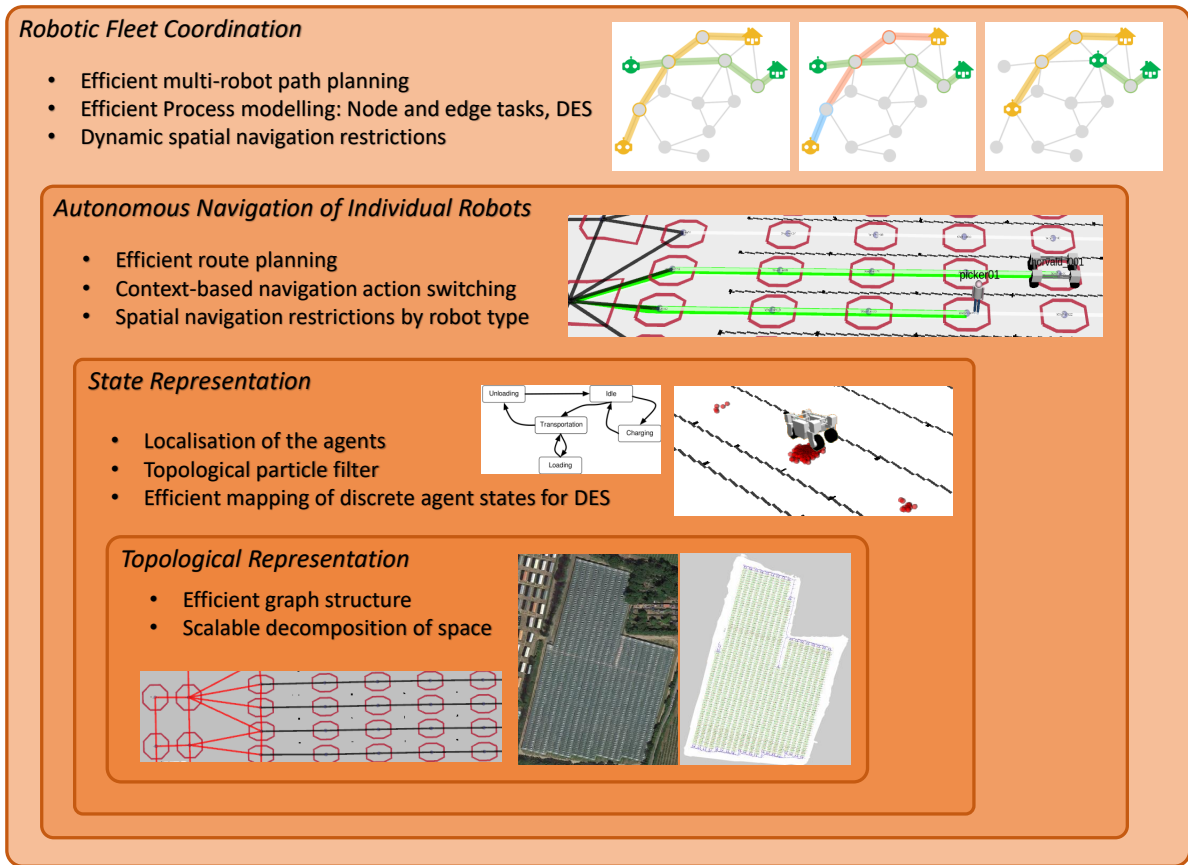


Figure 1: An overview of topological map representation as the backbone of a robotic fleet deployment providing decomposition, state representation, higher-level robot navigation planning and fleet coordination.

In this paper, we propose a *topological map* to be used as the unifying representation and computational model to facilitate both the orchestration of a fleet of otherwise largely decoupled individual robots, and their respective coordination (See Fig. 1). While individual robots will have specific implementations owing to their kinematic design, their sensor equipment, their navigation constraints, and their respective derived tasks and abilities, the abstraction of the current overall system’s operational state into a topological representation will be shown as being sufficient to facilitate both the coordination and management of such a fleet, as well as the modelling of overall system behaviour to allow for *ex-situ* modelling and analysis of operations.

It is this topological model that underpins the *decomposition* of the overall system architecture, the representations of the system’s overall synchronised *state*, i.e. the location of all involved actors (humans and robots), and the *coordination* of autonomous vehicles comprising the robotic fleet.

Both the task allocation and navigation coordination can be tied to the topological map and are driven by the state representations. Each task can be defined as specific actions at discrete nodes or along the edges and the coordinator allocates tasks to the robots based on the requirements of the tasks and the abilities of the robots. Robots receive route segments from a central coordinator to traverse routes within the topological maps, with the central coordinator ensuring that the planned routes are deadlock and collision-free, based on the current state representation of the location of humans and other robots.

3.1 Formal Definitions

A topological map is a discrete representation that can be viewed as a tuple $T \rightarrow \langle N, E, A, R \rangle$, where N are the nodes representing physical locations in Euclidean space, E are the edges which connect pair of nodes, A is the set of navigation actions that the robot can perform, and R is a set of boolean conditions that are used to define restrictions to the robot navigation on the topology. The framework allows the robot to use any navigation action, which is implemented for different purposes, that can be performed on the topology.

A node $n \in N$ is itself a tuple $n \rightarrow \langle q, p, Z, P \rangle$, where $q \in \mathbb{R}^4$ is the orientation represented as a quaternion, $p \in \mathbb{R}^3$ is the position in Euclidean space, Z is the set of Euclidean points outlining a polygon which defines the node's influence zone, and P is a logical predicate build from the atoms in R . Similarly, an edge $e \in E$ is a tuple $e \rightarrow \langle n_1, n_2, a, P \rangle$, where $n_1, n_2 \in N$ are the nodes connected by the edge e , $a \in A$ is the navigation action that the robot can perform to traverse e , and P is a logical predicate build from the atoms in R . Each condition in R can be evaluated as true or false in real-time as well as the predicate; if the predicate evaluates as false, the robot is not allowed to traverse the corresponding edge (or node). The framework allows the implementation of any number of conditions with its operation for real-time evaluation. The code for defining, visualising and using such a topological map is made publicly available to the community.¹

3.2 Decomposition

Using the topological model as a means of system decomposition allows for substantial separation of concerns, i.e., the software system facilitating the navigation of a particular robot between two neighbouring nodes in a topological map can be developed and optimised independently from the overall fleet architectures. Indeed, this system decomposition allows for very loose coupling between the individual robotics systems to a central coordination system, facilitating the support of heterogeneous fleets of robots. Different robots will have different sensors, form factors, and actuation and kinematic abilities. Specifically, on a system's level, a robot is represented by its (probabilistic) *location (state)* in the topological graph, its task- and form-factor specific *constraints*, capturing which edges in such a graph each robot can traverse, and a specification of the *tasks* each robot can fulfil in a farm environment (e.g. transportation, picking, scouting, etc). For example, Fig. 2 shows a unified topological representation that can be used for all fleet-level planning and coordination, while different types of robots operate on a subset of that topology based on their task- and form-factor-specific constraints.

The general nature of the topological map definition, of the actions and the constraints that can be applied to each nodes/edges, allows the use of any arbitrary robot and arbitrary tasks to be performed on the same topology. This level of decomposition is made possible thanks to the restrictions R , described in Section 3.1, which can be specified at the node/edge level and for specific robots/tasks. For example, one can define restrictions on a topological map such that on the same farm, some edges can be traversed by a robot that is tasked to perform picking or that has a gripper, while other edges are reserved for transportation robots only.

3.3 State Representation and Estimation

Adopting the topological abstraction model to represent the current overall state of the robotic fleet and its environments offers a unified representation, not only capturing the state of the robot but also of human actors in the environment. Human actors are represented in the very same way, with the respective (probabilistic) location and constraints also represented on this topological level, again allowing the integration of a variety of different sensors and tracking approaches to be unified into a central topological state.

Individual robots offer evidence about their localisation within the topological map to the central topological

¹https://github.com/LCAS/topological_navigation/

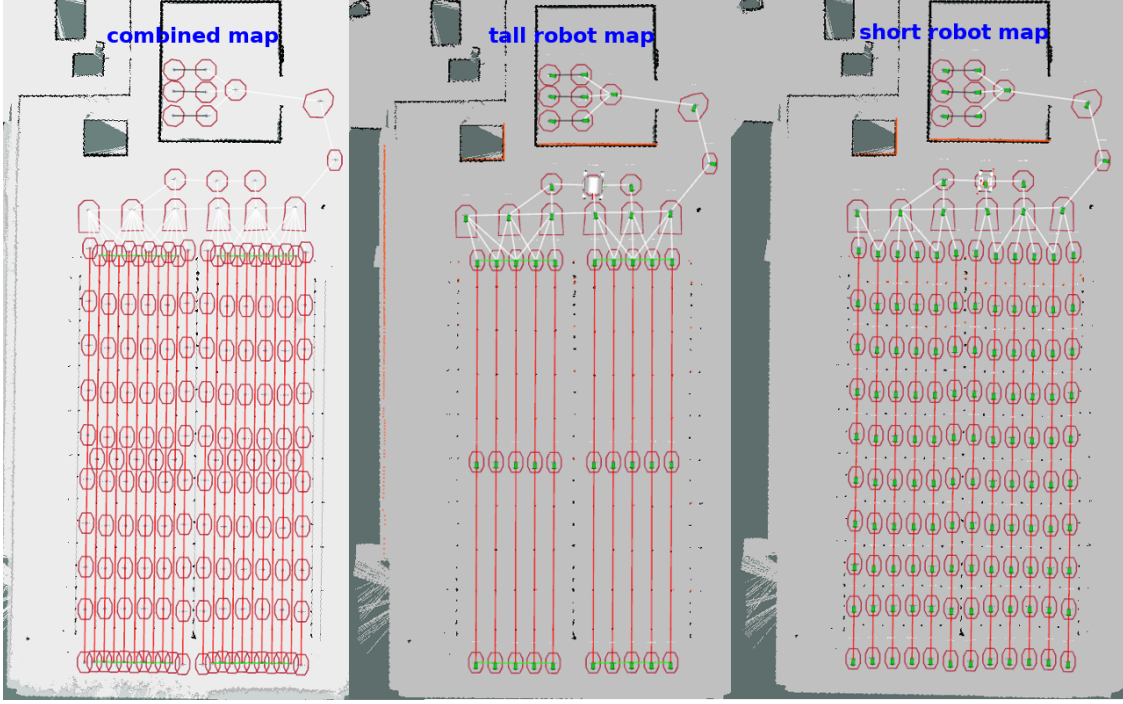


Figure 2: Demonstration of how restrictions are used within the unified topological representation to differentiate the navigable edges and nodes for robots with different form factors. On the left is the unified map which the server uses for general management of the fleet. The map in the centre shows only the nodes and edges available for the tall robot navigating over the tops of the raised beds. On the right shows the nodes and edges the short robot is able to traverse. The map on the right is also used for the Bayesian topological localisation of pickers.

localiser, different sensors to detect humans in an environment likewise are fused on the topological level, overall offering a probabilistic state representation of all actors in a farm environment.

The topological structure defined on the environment, being essentially a discretisation of the metric map, can be used to approximate the position of entities in the environment. The additional information provided by the topology structure helps to estimate the state of robots and humans on the map by reducing the search space and constraining the movements to the edges between nodes. This is useful especially in a structured environment – like polytunnels (also called hoop greenhouses) in agriculture – where a small Euclidean error between an estimated position and the true position on the metric map can represent a large distance in terms of the path the agent would have to travel to reach the correct position (see Fig. 3).

To localise agents on the topology we can obtain different types of inputs to identify their state according to the type of agent and the sensors available. For example, to localise a robot we can use the pose provided by the metric localisation stack, which is usually accurate at the centimetre level and map it to the closest topological node on the topology. Differently, to localise a human in the fields we typically need to rely upon partial and noisy observations coming from various sensors, and perform some sort of filtering and tracking of the target’s state.

In the remainder of the section, a state estimation system designed for localising farm pickers in a topological map will be described. This approach, integrated into the topological formalism presented, can be utilised for localising any agent based on heterogeneous observations. This is an example of how state estimation can be performed on our topological framework, and different methodologies can similarly be integrated.



Figure 3: A robot which, after reaching a picker at a wrong tunnel has to re-plan a path going out of the current tunnel first to enter the correct one. Note that the Euclidean error between the robot position and the picker is relatively small but, given the environment constraints, the robot has to travel 9 edges in the topology to reach the picker in its true position.

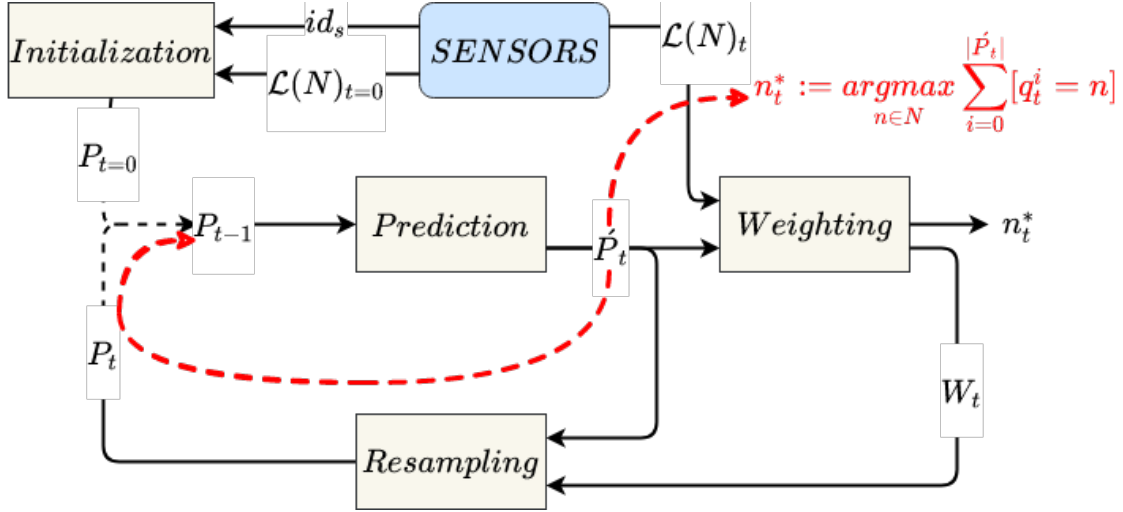


Figure 4: High-level sketch of the TPF update loop. The red dashed pathway outlines the operations performed to update the particles and estimate a node position for the target when no observations are coming from the sensors.

We attempt to tackle the problem of localising an agent on the topological map with an approach based on particle filtering, which we call a Topological Particle Filter (TPF). The TPF is first introduced in (Khan et al., 2020) and extended in (Polvara et al., 2021). The distribution of a set of particles P over the topology is used to approximate the probability distribution of the targets on the map. At timestep t , each particle $p_t^i \in P_t$, for $i = 1, 2, \dots, |P_t|$, in our TPF formulation is characterized by the state

$$p_t^i = \langle q_t^i, \vec{v}_t^i, \tau_t^i, T, V \rangle,$$

where $q_t^i \in N$ is the node the particle lies in, \vec{v}_t^i is the velocity vector of the particle, τ_t^i is the amount of time particle p_t^i has been in q_t^i , T is the topology and V is a fixed window size used to estimate the particle's velocity. The definition of the particle uses the Markov assumption, i.e., a particle state at time $t+1$ depends uniquely on its state at time t

$$Pr(p_{t+1}^i | p_{t=0}^i, \dots, p_t^i) = Pr(p_{t+1}^i | p_t^i). \quad (1)$$

The observations from sensors are obtained in the form of likelihood distributions $\mathcal{L}(N)$ over the topological nodes N in the map. Moreover, each sensor informs the TPF of whether the observations provided uniquely

identify the specific instance of the target to localise or not by advertising the variable $id_s \in \{\top, \perp\}$.

The TPF updates the estimate of the target in a series of sequential steps, outlined in Fig. 4:

1. *Initialization* upon receiving the first observation from a sensor it initializes the particles;
2. *Prediction* predicts the set of particles \hat{P}_t at time t given the previous state P_{t-1} ;
3. *Weighting* weights each particle with the likelihood value received from the sensor and generates the current estimate n_t^* of the agent position;
4. *Resampling* integrates the weights and the predicted particles to generate the new particle distribution P_t .

The TPF can perform predictions in absence of new observations from the sensors by simply repeatedly applying the forward *Prediction* model to generate the next step particles and producing the estimate target position as the node with the most particles, as highlighted in Fig. 4 with the sketched path in red. For the implementation details of each step of the TPF, the reader is invited to read (Khan et al., 2020; Polvara et al., 2021).

3.4 Modelling and Optimising System Interactions using Discrete Event Simulations

Evaluating the system performance in a simulated environment, before the actual field deployment can help in optimising individual modules as well as the interfaces and interactions between these modules. In these simulations, the state changes of the different *entities* (units which have different states) are modelled as functions of time and the interactions between the interacting entities. Simulations can be broadly classified into continuous time simulations and discrete event simulations. Continuous time simulations track the state changes of the entities at fine and constant temporal resolutions. In contrast, a discrete event simulation (DES) focuses mainly on *events* (an instance of system state change) and models them as a set of *operations* or abstract step changes that may take arbitrary time intervals depending on the processes involved. By computing these step changes, a DES jumps through discrete event times and updates the state of the individual entities and the whole system. Also, the complex physics modelling of the entities is not considered in the DES modelling. Thus, DES has a very low computational requirement and can simulate complex system interactions in a fraction of the time of what a continuous-time simulation might take. A detailed overview of the DES concept is available in (Schriber et al., 2017).

Due to the discrete step changes and events in the modelling, DES has high synergy with the topological map representations that discretise the environment to discrete nodes and edges connecting them. e.g., an agent’s navigation from a node to an adjacent node through an outgoing edge can be modelled as a single operation, that changes the state of that agent. This state change could be the change in the position as well as progress in another task it is involved in (e.g. moving closer towards a distant target location).

3.5 Fleet Coordination

Based on the topological models or navigation constraints, and a synchronised state representation, the fleet of robots needs to be coordinated to accomplish tasks in collaboration with humans (whose locations are part of the state representation). Both the task allocation and navigation coordination can be tied to the proposed topological map and are driven by the state representations. Tasks are defined as specific actions at discrete nodes or along the edges and the coordinator allocates tasks to the robots based on the requirements of the tasks and the abilities of the robots.

Coordinating the long-term operations of a fleet of autonomous robots in a dynamic environment is a multi-faceted challenge. We identify three major groups of management tasks that combined cover all aspects needed to handle a fleet of robots efficiently:

1. Fleet management:
 - monitoring the state of agents in the environment
 - managing the membership of the active robotic fleet
 - managing in-field charging facilities
2. Task management:
 - registration of new tasks
 - task allocation to robots
 - task scheduling
 - monitoring the progress of allocated tasks
3. Traffic management:
 - individual and combined route planning
 - dead-lock detection and mitigation

In this work, we have opted for a centralised approach as the major focus is to have a proof-of-concept implementation and deployment of a fully autonomous robotic fleet with human agents in the environment, with the topological map representation underpinning the modules addressing these fleet management challenges. The centralised approach also helped to have complete observability of the agents in the system, and to ensure optimal decision making. These fleet management modules could be extended to a decentralised approach with additional communication protocols to reach consensus on decision making.

The first one of these core fleet management challenges is *fleet management*, which starts with the continuous monitoring of the states of the robots. For coordination, the state of the robot may include its position in the topological map of the environment (as discussed in Sec. 3.3), its capabilities (for executing different tasks), battery charge, and on-board device health status. The long-term operation of a robot can result in changes in its state. By continuously monitoring these, the coordinator can maintain an *active fleet* of healthy robots to perform any impending tasks by dynamically unregistering any robots with detected faults or operational anomalies and registering the robot back into the active fleet once it is fixed. For example, a robot with a low battery charge may have to be taken off from the active fleet (i.e. is no longer available to perform tasks in the field) and sent to a charging station. This robot can re-join the active fleet after getting charged as shown in Fig. 5.

The second core challenge is *task management*, especially in a dynamic environment with chances of new tasks appearing randomly and with a heterogeneous fleet of robots to perform different types of tasks. In our framework, any task in an environment involves performing task-specific actions and can be classified either as a node task or an edge task based on the location this is performed. Node tasks involve the execution of task-specific actions at one or more topological nodes (i.e. actions are limited to discrete locations) and edge tasks involve the execution of task-specific actions along one or many consecutive topological edges (i.e. actions are continuous over space and time). Thus, a task $t_j \in T$ can be defined as a tuple $\langle n_j, tt_j \rangle$, where t_j is the j^{th} task, T is the group of all tasks, n_j is an ordered list of topological nodes and tt_j is the task type. For a node task n_j represents the discrete locations to execute the task actions and for an edge task n_j is a tuple $\langle n_j^s, n_j^f \rangle$, where n_j^s is the node from which the edge task starts and n_j^f is the node at which the task finishes.

A robot assigned to a task should be able to navigate to the topological nodes and edges involved and should be configured with the required on-board hardware and software capabilities to perform the actions

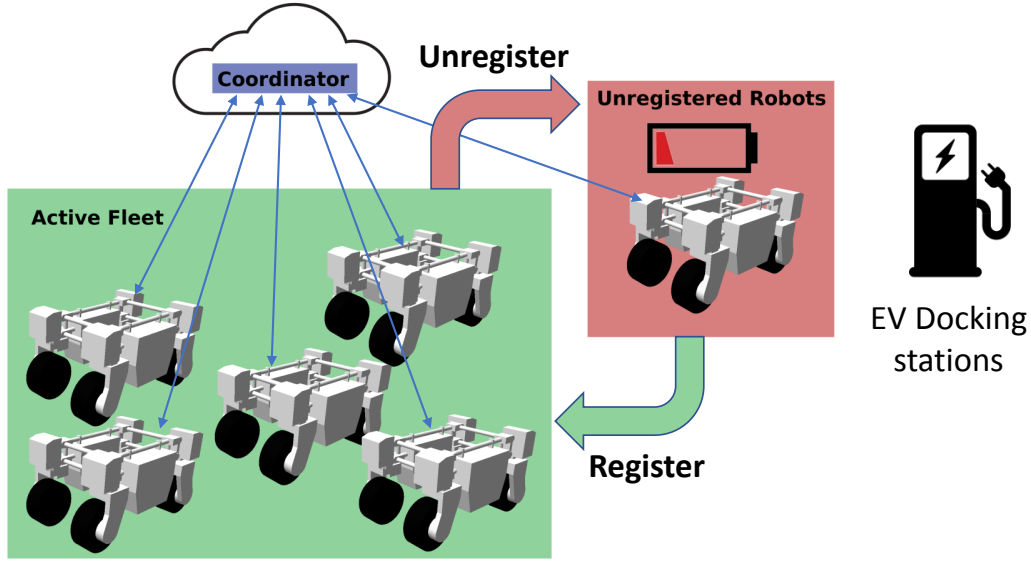


Figure 5: An example of dynamic active fleet management - *Unregister* robots for charging and *Register* after charging. Tasks will be allocated only to the active fleet.

corresponding to the task. In our task allocation formulated as an optimisation problem for sequential single-item allocations, the objective is to minimise the topological route distance for a robot to navigate to the task starting topological node (n_j^s) to perform the task, and the constraints include the presence of on-board sensors and actuators required for the task-specific actions and access for the robot to the task location (i.e. absence of any topological restrictions).

This leads to the third among the core challenges *traffic management*, which handles path planning for the fleet. While multi-robot systems are highly recommended for sharing workload and robustness, the most critical challenge is to ensure the safe navigation of the individual robots and make sure the robots do not run into each other while moving around the environment. Having a centralised point of control allows us to take all relevant agents into account when planning routes, including both the robots' and humans' locations.

Finding a route from the robot's current location to the task location on a discrete topological map, considering restrictions on nodes and edges is computationally less complex than planning on a metric map. However, to further ensure efficient routing for our topology (which contains long parallel rows uniquely to poly-tunnels), we developed a new route planner which was optimised for this style of environment.

Our route planner (named the *fragment planner*), works to avoid deadlocks by having robots wait in front of bottlenecks if the pathway is occupied or allocated by another robot. It begins with blocking access to occupied nodes by removing their connecting edges from the map, where nodes are considered occupied if they are the closest node to any agent which may act as a physical obstacle to the fleet. Each agent is then identified as mobile or static by whether they have a navigation goal, and each mobile agent individually searches for a route by unblocking nodes explicitly required for the task and performing a route search. Mobile agents are converted to static agents if their route is empty and each static agent is assigned their closest node as their route.

The planner resolves potential conflicts by first identifying intersections in the mobile agents' routes. Each



Figure 6: Example of fragment planner operation: (left) Optimal routes are identified by the planner; (centre) Routes are broken up into fragments beginning at the node before an intersection if they are not the closest agent to the conflict; (right) Agents move to next fragment of route once the path is clear.

agent’s route is then divided up into fragments where each new fragment is begun with the node before an unattainable critical node, where we define attainability based on agent proximity. This planner results in agents’ given routes which can be efficiently generated, and completed in stages, allowing robots to pause their routing before colliding with other moving robots.

Whilst the robot’s local navigation will prevent the robots from physically colliding with one-another, limiting our reliance on this system with more efficient topological route planning improves the productivity of the system as a whole.

4 Validation Use-Case: Supporting Fruit-Pickers in Horticulture

20-30% of the labour cost associated with strawberry picking is spent walking trays of picked fruit and empties back and forth from the picking locations to local storage areas, typically situated near the edge of the field and easily accessible by transportation vehicles. In addition, around 10% of the field area can be designated for such transportation needs. For this purpose, we propose to employ a fleet of transportation robots equipped with a dedicated picking tray storage system allowing the movement of full/empty trays around the field between the human workers and storage units. The transportation robots are equipped with on-board multi-modal sensing to support the safe and autonomous navigation of the robots and their collaboration with humans.

In this, each fruit picker is provided with a smart trolley that can carry up to four trays of strawberries. Once the trays on the trolley are (almost) full, a robot can be requested. The idle robots will already be loaded with empty fruit trays by support workers and will be waiting at their base stations. The fleet coordinator, when it receives the call from the picker, will allocate a suitable robot which will then go to the picker. Once the robot arrives at the location of the picker, it is stocked with full strawberry trays while the trolley is loaded with empty trays provided by the robot. The fruit picker can now keep on picking while the robot transfers the full strawberry trays to the local storage. With the robot’s task being to collect trays inside the polytunnels and move them to a collection area outside, the robot spends a significant time traversing these tunnels. For this cycle to work, the system must

- have a human computing interface (HCI) device to register the calls for robots,
- localise the pickers accurately in the environment,
- have an autonomous fleet coordinator to monitor the calls from the pickers and allocate and schedule the tasks,

- have a multi-robot traffic management system for avoiding any physical interference, and
- have fine-tuned robot navigation in the long and narrow navigation rows.

The human picker interacts with the system through a *Smart-Trolley* device (see Sec. 4.2) which allows making calls for an available robot from any location in the field. The centralised fleet coordinator system (Sec. 3.5) employs the proposed topological representation to track the locations of the robots (see Sec. 4.1) and smart trolleys (i.e. human pickers), assign tasks based on the user requests and orchestrates autonomous navigation by devising optimal and non-conflicting paths. A digital twin of the operational environment (i.e. strawberry farm, see Sec. 4.3) with models of robots and human pickers is also developed to facilitate development, optimisation and the final deployment of the fleet in the real scenario. The individual components of the system are described as follows.

4.1 Thorvald robot

The core robotic platform employed in this work is Thorvald - a modular, general-purpose, lightweight and flexible robotic system designed to address multiple applications of field robotics in agricultural environments (Grimstad and From, 2017). The system consists of a set of re-configurable modules which can be combined freely to create robots of varying sizes and with varying kinematic properties suitable for a particular task. To support strawberry production, we use a four-wheel drive and steer (4WD4S) wheel setup with a narrow base configuration (1.0 m wide) which can fit into a single polytunnel row to support in-field logistics tasks. The robot’s battery provides roughly 10 hours of autonomy without payload. The robot is controlled through a built-in computer running Linux OS and Robot Operating System (ROS).

The robot navigates autonomously by employing the proposed topological representation (see Sec. 3.3) and executing a navigation plan devised by the fleet coordination component. The topological representation allows for specifying different navigation actions depending on the characteristics of the environment. The actions that were implemented for the in-field logistics scenario in strawberry polytunnels include $A = \{move_base, row_traversal, row_change\}$; these are used for navigation in open space, traversing a row of the tunnels and moving from one row to another correspondingly. To localise the robot in the topological map, we exploit the output from an off-the-shelf localisation module as the input to define the node where the robot is located or the edge the robot is traversing. We used the *robot_localisation* package (Moore and Stouch, 2014) combining the wheel odometry and the RTK-GPS, obtaining an accuracy of around 2 cm. To incorporate the RTK-GPS data into robot map coordinates, the same fixed datum configuration is used for all the sessions.

4.2 Smart-Trolley

To enable interaction between the robots and farm workers, we propose to employ a *Smart-Trolley* device which is a standard transportation cart used in the soft fruit industry equipped with an electronic system providing geo-location information and a user interface for interaction with the fleet coordination system. The device presents 3 easily accessible buttons which allow the workers to *call a robot* at the worker’s location to pick-up a tray, *cancel* the call, and *signal the loaded tray* so that the robot can transport the produce to the storage location. An LCD screen is used to show the status of the requests and tasks issued by the coordinator. The geo-location information from a GNSS unit is sent to the coordinator for tracking the topological location of each worker/trolley through the Topological Particle Filter as outlined in Sec. 3.3. The device is controlled by a Raspberry PI Zero single-board computer together with a Waveshare SIM7600E-H WiFi/GNSS unit. Although the same functionalities could be provided by a dedicated app installed on smartphones, these are difficult to operate by farm workers who often wear gloves and are unreliable when affected by dirt or weather.

Integrating multiple signals from many such smart devices in the field with our centralised TPF tracking



Figure 7: The Thorvald robot used for a fruit transportation task.

framework allows us to keep track of their trajectory over time maintaining separate candidate positions for each picker. There are still many challenges, however, given by the noise and inaccuracies of the GNSS signals (exacerbated in polytunnels by the overhead metal structure) which can sometimes locate a picker in the wrong row or tunnel. To compensate for such disturbances, the proposed tracking framework can easily incorporate other localisation sources such as LiDAR-based leg detectors utilising the robot’s onboard sensors (e.g. (Polvara et al., 2021)).

4.3 Strawberry Farm Environments

We deployed our picker supporting the fleet in two different real environments including a small strawberry production facility at Riseholme campus, and a large commercial farm in Kent, UK. The sites differ not only in scale but also in terms of the challenge they pose to robotic operations. The strawberry production facility at the Riseholme campus is modelled after typical commercial strawberry farms and was specifically built for robotic deployment. It features two polytunnels of 24m in length each including 5 rows of strawberry plants grown on tabletops which are mounted on steel poles for easier maintenance and harvesting. The ground is gravel, even and easily traversable both by humans and robots. The site is professionally maintained which includes automatic irrigation, regular spraying, weeding and thinning operations. The commercial fruit production farm in Kent, UK has 20 polytunnels of lengths ranging from 130m to 240m, each with 5 rows of raised beds of strawberry plants. This environment is more challenging for autonomous robot operations. The ground is mostly compacted soil with many irregularities caused by water and growing weeds. The arrangement of table tops can be irregular in places and crop canopy significantly obstructs the traversable paths. The topological representations used for the fleet deployments at Riseholme are shown in Fig. 9 (top-left) and those of the Kent environment are shown in Fig. 9 (top-right) and Fig. 9(bottom).



Figure 8: The *Smart-Trolley* device for the human pickers. From left to right, the buttons' functions are: calling (green button), cancelling the call (red button), and signalling a loaded tray (blue button). An LCD screen shows the status of the tasks.

The topological map for the Riseholme farm consists of 10 nodes per lane approximately at a distance of 3m from each other, giving 60 nodes per tunnel which together with 17 service and drop-off nodes results in 137 nodes in total. The topological map for the Kent farm consists of topological nodes ranging from 46 to 85, approximately at a distance of 3m from each other inside the polytunnel lanes, giving 5425 discrete points in total. Some of the evaluations are carried out in a smaller section of the commercial field with four polytunnels of lengths 130m, with 46 topological nodes per lane resulting in 184 discrete points per tunnel.

To facilitate seamless optimisation, deployment and evaluation of the proposed system and its components, we have also developed a digital twin of all three farms, developed in Gazebo, featuring the realistic appearance of its polytunnel infrastructure, Thorvald robots and human pickers.

4.4 Topology optimisations and DES

The strawberry polytunnels with raised tables are designed for the convenience of human workers and manually driven small tractors used for crop care applications. The performance of a fully autonomous robotic fleet in these environments may not be optimal due to the size of robots causing traffic congestion when their autonomous navigation paths cross or overlap. By detecting areas where such traffic congestion occurs and mitigating them by making modifications in the environment such as providing more space for parallel lanes for more than one robot to navigate in parallel. Combining with the topological maps, DES (see

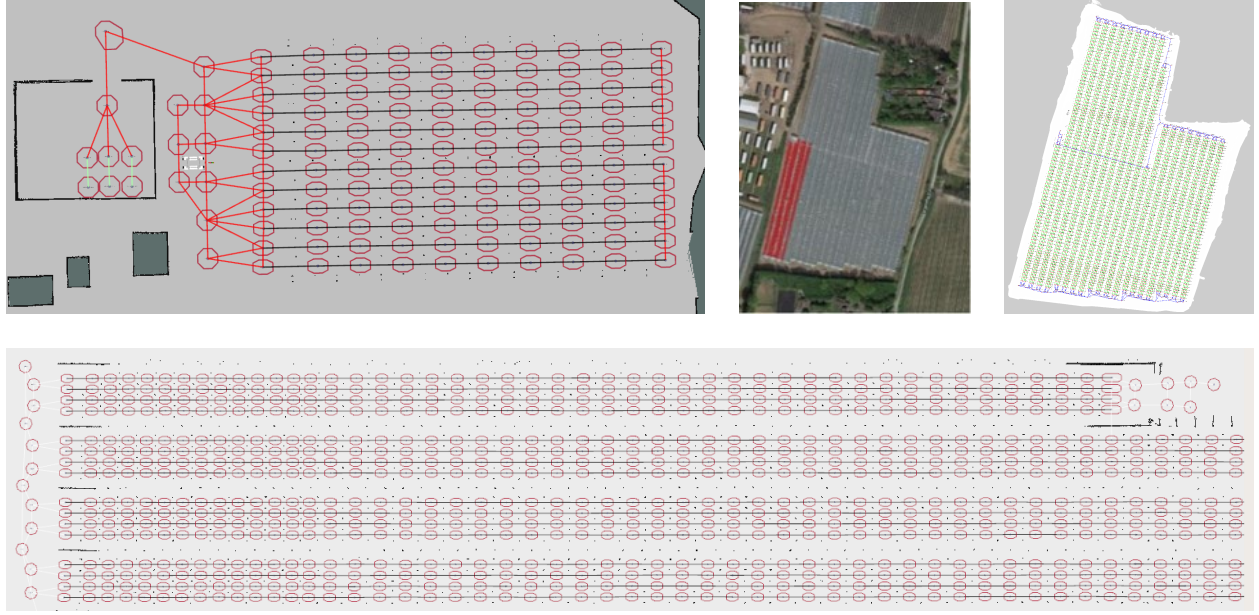


Figure 9: (Top-left) Topological representation of the Riseholme environment featuring two 24 m long polytunnels and a Food Handling Unit; (top-centre) aerial view of the commercial strawberry field; (top-right) topological map for the full field; and (bottom) the section of the topological map for the area highlighted in red in the aerial view (top-centre). All topological maps have head lanes at both ends.

Sec. 3.4 and (Das et al., 2018)) can be used as a computationally low-cost tool for inspecting the effectiveness of different modifications to improve robotic fleet operations before they are carried out in the field. Also, DES can be used to optimise the placement of some service nodes for specific robotic applications. Focusing on the logistics tasks, DES is used in this work to identify traffic bottlenecks and to evaluate the effectiveness of different mitigation strategies.

5 Evaluation

5.1 Localisation of Human Pickers

In this section, we show the experimental evaluation in localisation and tracking of pickers in the polytunnel environment from multi-modal sensory observations using the Topological Particle Filter (TPF) (described in Sec. 3.3) to effectively exploit the graph structure of the navigable space. While the TPF formalism has been evaluated in (Polvara et al., 2021) in simulated field environments, this section presents the results of applying the methodology to data collected in during two distinct real-world deployments in the strawberry fields in the Riseholme and Kent farms.

In all the experiments the humans were moving along the polytunnels – simulating the typical speed of strawberry pickers given that the data were recorded out-of-season for picking strawberries – while carrying a smart trolley device (described in Sec. 4.2) able to detect the GNSS position of the human. A Thorvald robot was present in the fields alongside the humans to support their picking operation. The LIDAR sensors signal on the robot base was used, other than for robot localisation, to perform legs detection of people in the field which feeds directly in the TPF, together with the GNSS signal, for the task of pickers localisation. The data collected in-field has been manually annotated at a later stage by indicating the topological position of the picker at each time-step of the recordings. Given that we are performing localisation at the topological

Table 1: Performance for each real-world scenario. Results are reported as mean and standard deviation. The best method is in bold.

Method	Euclidean Error[m]	Topological Error[nodes]
Riseholme run 1		
closest-node	2.95(1.27)	5.66(3.20)
TPF(<i>ours</i>)	1.06(1.56)	0.41(0.66)
Riseholme run 2		
closest-node	3.11(2.58)	2.77(2.54)
TPF(<i>ours</i>)	3.68(2.55)	3.72(2.68)
Riseholme run 3		
closest-node	5.98(2.26)	6.51(2.61)
TPF(<i>ours</i>)	5.75(2.29)	4.35(2.55)
Riseholme run 4		
closest-node	4.55(2.76)	3.20(2.62)
TPF(<i>ours</i>)	3.75(2.77)	2.91(2.72)
Kent run 1		
closest-node	4.04(2.55)	16.57(25.19)
TPF(<i>ours</i>)	2.80(2.4)	11.94(13.36)
Kent run 2		
closest-node	2.02(1.44)	9.44(13.13)
TPF(<i>ours</i>)	2.01(2.13)	1.61(2.55)

level only, the ground truth position of the picker is indicated by the name of the topological node that the picker is traversing. At each experimental run, more than one person was in the fields traversing the tunnels, however, only one person per run is the target of the picker localisation and has been annotated as ground truth.

In the experiments performed, we considered two metrics for assessing the performance of our solution in localizing the fruit pickers: the *topological error* and the *Euclidean error*. The first represents the number of nodes in the shortest path in the graph (i.e., the topological map) between the estimated node and the closest node to the real position of the fruit pickers (highlighted in Fig. 3). The Euclidean error is the Euclidean distance between the estimated node projected on a metric map and the ground truth position of the picker. It's important to notice that, given the topology of the map constraining the robot's movements only along edges connecting two nodes, a small Euclidean error does not always correspond to a small topological error.

Table 1 shows the results of our method in different runs in the real field environment compared with the baseline, predicting the position of the picker as the closest node in Euclidean distance to the GPS signal. It's important to note that the GPS signal's noise is substantially large in most parts of the runs, resulting in the *closest-node* approach frequently mistaking the row or even tunnel the picker is traversing. This error also negatively affects our proposed method, the *TPF*, which however achieves considerably higher performance at localising the target picker by integrating the LIDAR leg detections for a more accurate estimation of the position of the pickers. The presence of multiple people in the field also is the cause of mispredictions if the leg detector is not able to identify the different people from the LIDAR data, and consequently, the TPF cannot distinguish whether an observation from the leg detection module is the one carrying the GPS-enabled smart trolley device or just another person (i.e. a false positive). An instance of such a false positive error can be seen in Table 1 / Fig. 10 *Kent run 1* where for an entire tunnel the TPF follows the wrong person, as detected by the leg detector since they were closer to the GPS' noisy signal at the entrance of the tunnel.

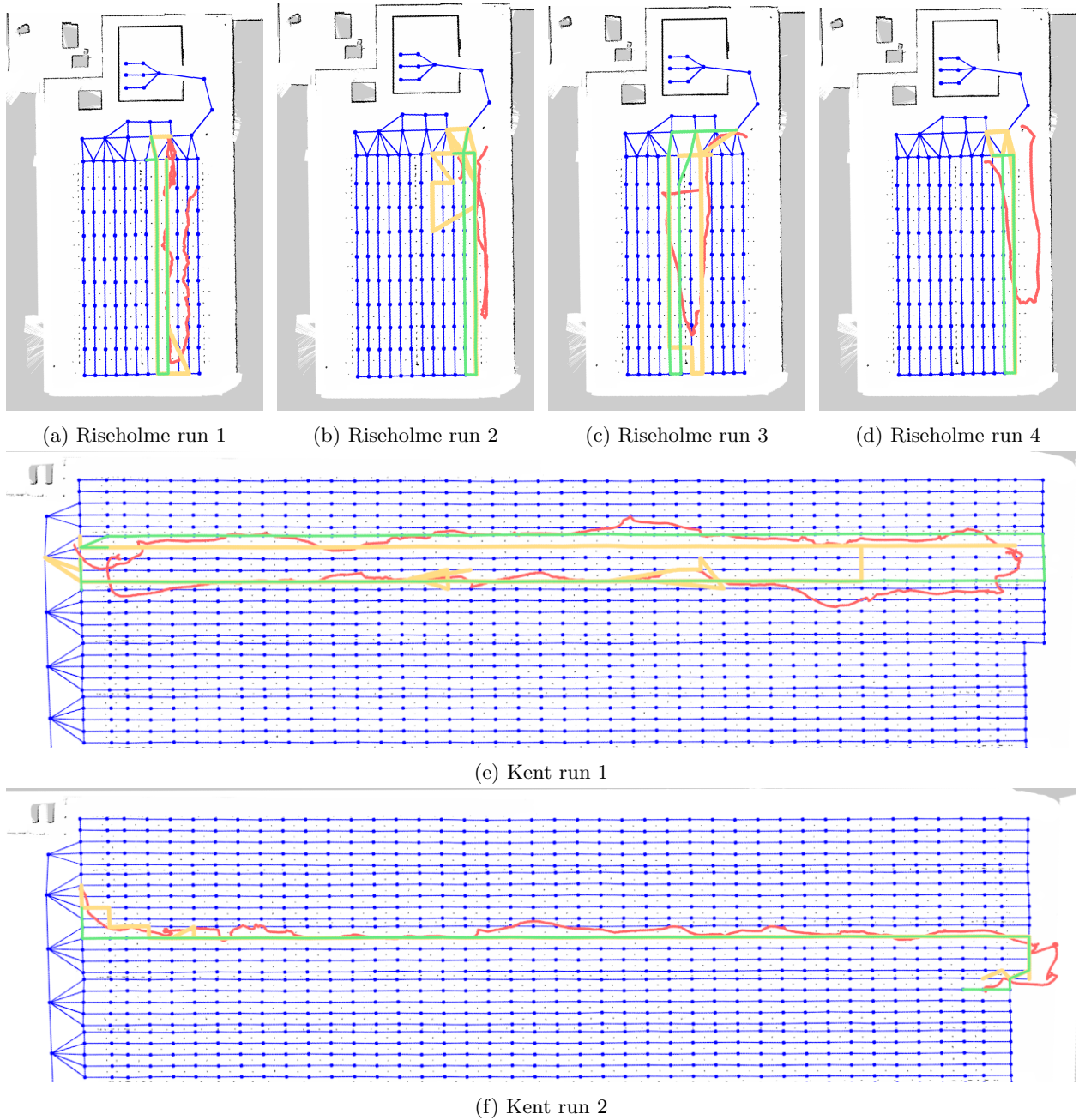


Figure 10: Trajectories of ground truth position (green), GPS signal (red) and position estimated by the TPF (yellow) for different experiments in the real scenario at Riseholme and Kent farms.



Figure 11: Infield logistics with two robots: (top-left) a robot is sent to a picker who requested logistic support using the Smart-Trolley device; (top-right) the robot arrived at the picker; (bottom-left) the picker loads trays on the robot; and (bottom-right) the robot starts navigating to drop-off node.

5.2 Testing of the complete autonomous logistic support system

The original workflow of the logistic support system (see Fig. 11) involved the pickers calling for a robot when they have a full tray of fruits ready for collection, and the fleet coordinator assigning a robot which then travels to the picker with empty trays. In this, the pickers will have to wait from the time when they made the call to the time the assigned robots reach them. Assuming the robots are travelling at the same speed as the human workers, this can save half of the overall time the pickers may otherwise spend on transportation alone (Das et al., 2018). However, the commercial polytunnel environment with long and narrow paths requires the robots to navigate hundreds of metres to reach the picker to collect the trays. The head-lanes connecting the long polytunnel rows are shared by all robots and can be a bottleneck in reducing navigational efficiency. The fleet coordinator manages the robot traffic by limiting the number of robots allowed to pass through these bottlenecks (see Sec. 3.5). However, when multiple robots have to pass through a section with a navigation bottleneck, this traffic management strategy introduces additional delays for the robots which are blocked from passing the bottleneck. If a robot going to a picker is blocked, the pickers will have to wait for a longer time for the robot to arrive, which cancels the intended productivity gain from the logistic support system. To reduce this delay, an alternate workflow is implemented, where a picker starts with three or four empty trays on a trolley, makes a call for a robot after the first tray is full and continues picking until the assigned robots reach them. In this modified workflow, the continuous localisation of the smart-trolley device using the TPF is utilised to update the goal node associated with the task assigned to the robot was updated whenever the picker's location was updated. This modification completely removes the unnecessary waiting period for the pickers, and the pickers can make the best use of the time by continuing to pick and fill up more trays.

An autonomous robotic fleet consisting of two Thorvald robots for logistic support was tested by two pickers in four polytunnels of approximately 130m in length in the commercial production field in Kent (the highlighted area in Fig. 9(top-centre)). This test was split into two trials to evaluate two different workflows, covering three polytunnels in total. All three polytunnels had approximately the same yield. The first trial involved

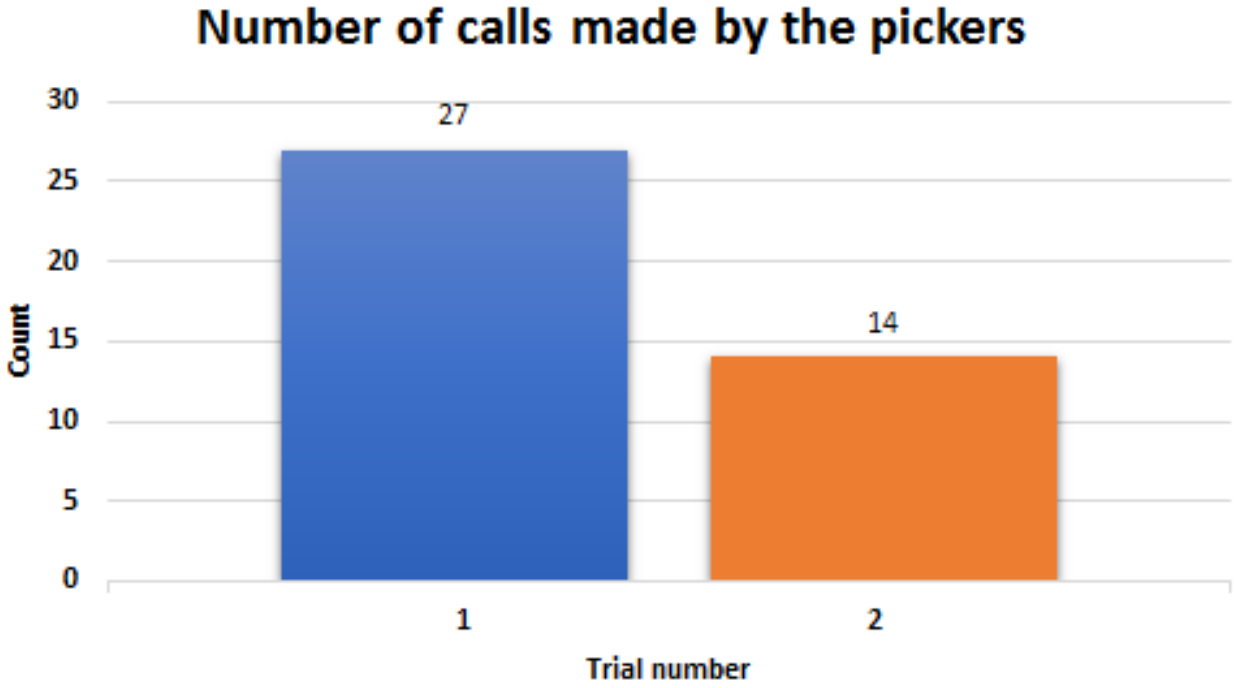


Figure 12: Total number of calls made by the pickers in the trials

the two polytunnels closer to the service nodes (robot's base stations and fruit drop-off point), while the second trial involved the next polytunnel resulting in a marginal increase in the distances to travel and more navigation congested head-lanes. Fig. 12 shows the number of calls made by the pickers during these trials. There is no significant change in the number of calls per tunnel (13.5 and 14 respectively) between these two trials. Fig. 13 shows the time the robot took to reach the pickers after they were assigned the tasks in each of the calls for both trials. In the first trial, this time is equivalent to the unnecessary wait time, which is completely avoided in the second trial with the modified workflow. Although not evident in these tests, in hindsight, the farther polytunnels would have resulted in longer travelling time for the robots. In the second workflow, it would have resulted in pickers filling up more trays by the time the robots reach them, this, in turn, can reduce the number of calls from the pickers and the overall robotic traffic congestion.

One observation made during the trials is regarding the localisation accuracy of the TPF and its effect on the navigation goals set for the robots. Although the TPF consistently retains the position within the correct row by enforcing the topological constraints, the estimated closest node of the picker is not accurate (see Table 1) resulting in setting the robot a goal node that is approximately two nodes away on average. Although the pickers have to carry the trays over this distance to the robot, it is substantially smaller compared to the distance to the actual drop-off point. Another observation from these tests is the traffic congestion along the head-lanes which is a clear bottleneck affecting the overall system performance. With a larger number of robots in the fleet to support more human pickers, this will be more substantial. This can be addressed by further optimisation of the topology of the environment, as well as the topological map for the best use of such an automated robotic fleet. This is presented in Sec. 5.3, by running DES of the system to evaluate the effect of different possible modifications of the topological map in the overall picking process.

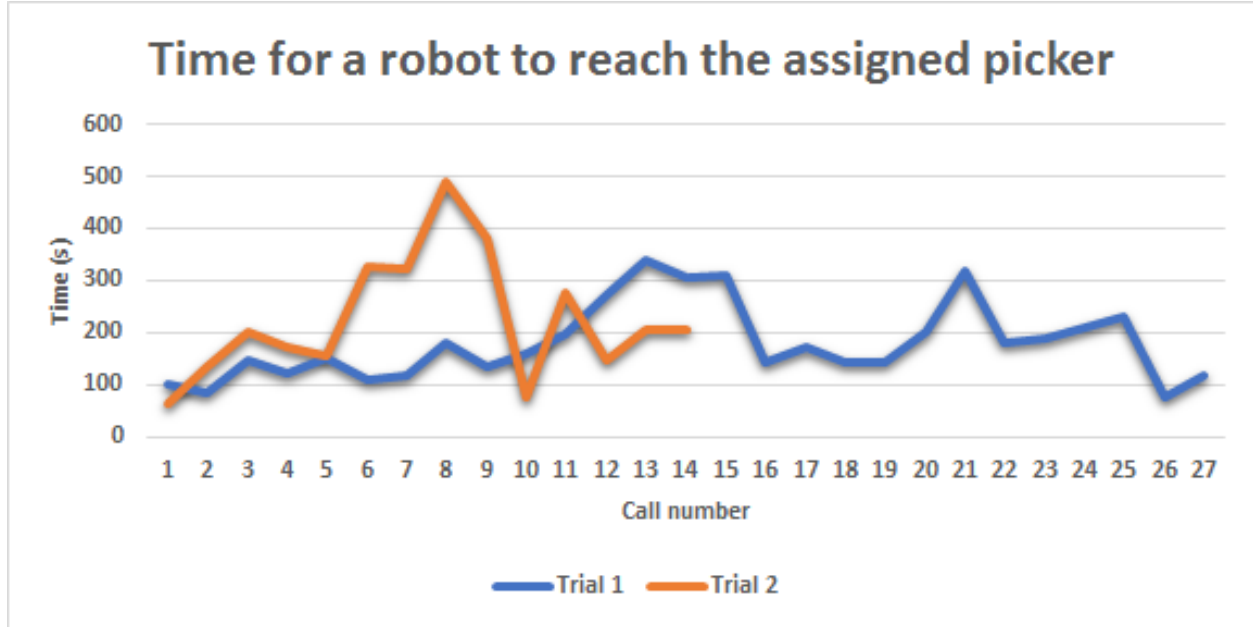


Figure 13: Time the robots took to reach the picker, once they were assigned to the pickers after each call in both trials. The large values indicate the delays caused by traffic congestion.

5.3 Optimisation of topology for in-field logistics tasks using discrete event simulations

It was shown in (Das et al., 2018) that DES could be used for simulating picking and in-field logistics operations on a topological map representation and for optimising the robotic-fleet size for a given field and the number of human farm workers. Here, we extended these simulations to analyse the effect of changes in the topological map on the robotic-fleet traffic while providing in-field logistic support for the picking operations. The optimisation of the topological map is significant as many robotic field operations involving navigation could be shortened with cumulative benefits. For example, by reducing the time a robot would take to reach a picker, the time the picker would wait for the robot could be reduced. Such traffic optimisations could reduce the time the picked fruits are kept outside (by bringing them to the drop-off point earlier, they could be moved to cold storage as soon as possible). The parameters used for these simulations were derived from (Harman and Sklar, 2021) and (Das et al., 2018) and are presented in Table 2.

In these simulations, the topological maps of three different fields of varying complexity were modified and their effects on the overall picking and logistics operations are analysed. The topological maps considered are (i) the Riseholme map, (ii) Four polytunnels in the Kent map, and (iii) the Full Kent map. Inspired by traffic management strategies in literature, such as entry and exit restrictions (Ryan, 2008), multi-robot path planning (Madridano et al., 2021), agent-based technologies for traffic and transportation (Bazzan and Klügl, 2014), and robotic forklifts for intelligent warehouses (Vivaldini et al., 2010), we identified the following topological modification strategies.

- adding parallel lanes in high-demand areas
- setting multiple storage locations (in-field drop-off points)
- setting multiple robot-waiting-stations
- adding cross lanes in long rows
- allocating multiple storages at different regions of the field

Table 2: Parameters of DES. Note: $X \sim \mathcal{N}(\mu, \sigma^2)$

Parameter	Value	Note
picker_picking_rate	$\mu = 0.4 \text{ m/s}, \sigma = 0.02$	described the average speed with which pickers move while picking in a row
picker_transportation_rate	$\mu = 1.0 \text{ m/s}, \sigma = 0.04$	speed of picker when carrying crate
picker_max_n_trays	$\mu = 1.0 \text{ trays}, \sigma = 0.04$	maximum number of trays on the trolley
picker_car_n_trays	4	picker calls for a robot after filling these many trays
picker_unloading_time	$\mu = 10.0 \text{ s}, \sigma = 0.2$	how long it takes a picker to unload a tray at storage or load a tray on the robot
tray_capacity	1200.0 g	the maximum weight the tray could carry
robot_transportation_rate	1.0 m/s	an array of floats, speed of a robot (values for all robots, or single value to be copied)
robot_max_n_trays	4.0 trays	maximum number of trays on the robot
robot_unloading_time	10.0 s/tray	time to unload a tray at the drop-off point
yield_per_node	$\mu = 200.0 \text{ g}, \sigma = 5.0$	yield between two adjacent topological nodes, the distance between two nodes is about 2-3m

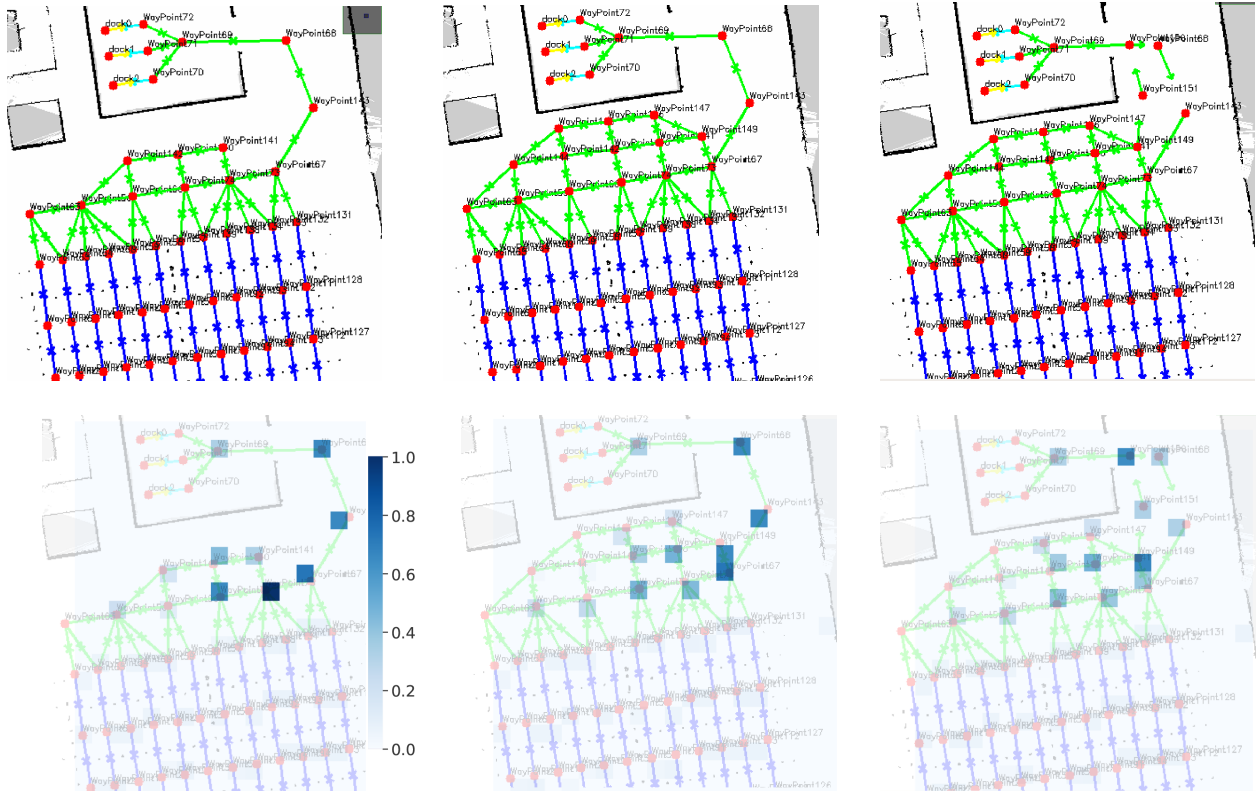


Figure 14: Riseholme topological maps on top and corresponding node usage on the bottom (Deep blue colour indicates nodes in high demand). (left) Original map; (centre) topological map with a parallel lane near the head-lanes; and (right) a parallel head-lane and a roundabout near the entrance of cold storage.

After the topological map modification, we ran 10 trials of the DES of the full picking and logistics operations on the modified maps with 4 pickers and 2 robots in the modified map to obtain different metrics: node_usage and simulation_time. Here, node_usage was the relative frequency of how often the node has been used, and simulation_time was the total simulation time used for a whole picking and logistic process.

5.3.1 Modification on Riseholme field

For the Riseholme field, initially the nodes in high-demand were identified as possible bottlenecks. After this, two strategies were used to modify the topology to address these bottlenecks:

- Adding parallel lanes in high-demand areas, i.e. near the head-lane as shown in Fig. 14 (centre-top)
- Using a roundabout near the entrance to the storage node as shown in Fig. 14 (right-top)

The original and modified topological maps along with the results (node usage heatmaps) shown in Fig. 14. In this figure, the topological maps are shown on top and their corresponding node usages are shown directly below each of them. From the figure, it can be noticed that:

- Parallel lane nearby waiting stations (high-demand areas) reduces traffic demand and congestion
- A roundabout near the entrance of cold storage (single track) further reduces traffic congestion

5.3.2 Modification on Kent 4 polytunnels field

The complexity of the topological map was further increased at the Kent 4 polytunnels field, which nearly had 10 times the route length compared to the Riseholme map. Fig. 15 shows the original topological map and the high node usage along the head-lane. The drop-off point is the outermost node on top and the robot-waiting-station nodes are on the lane adjacent to the route to the drop-off point. To address this bottleneck, three topology modification strategies were used:

- Adding parallel lanes in high-demand areas (Fig. 16 left-top)
- Setting multiple robot waiting stations (Fig. 16 centre-top)
- Setting multiple storage locations (Fig. 16 right-top)

Ten trials of DES were completed with the modified topological maps and the results are shown in Fig. 16. By comparing these results against the node usage of the original map in Fig. 15, it can be seen that:

- Parallel head-lane reduces traffic demand and congestion along the head-lane
- More robot waiting stations near the head-lanes reduce traffic congestion near the drop-off node
- A parallel lane near the storage further reduces traffic congestion, but still with a bottleneck at the node connecting to the drop-off point

5.3.3 Modification on Kent full field

Finally, the complexity of the topological map was increased to the full field, which is nearly 8 times the 4 tunnels considered previously. Compared to the Kent 4 tunnels map, the tunnels are much longer. This can result in long navigation routes for the robots to reach the pickers. Also, depending on the number of pickers, the picking process may spatially move across the field, causing variations in the service distances. To address these challenges, two strategies were used to modify the topology:

- adding cross lanes in long rows (Fig. 17)
- using multiple storage nodes in different areas of the field (Fig. 18)

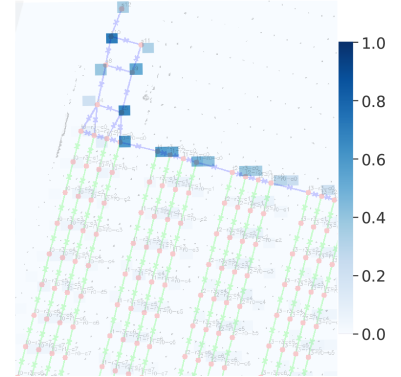
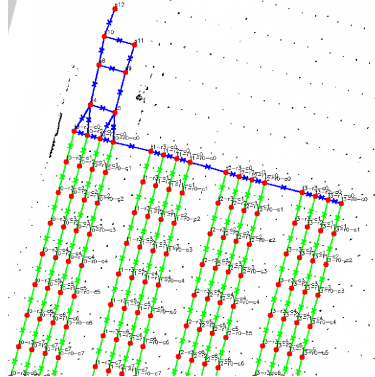


Figure 15: (left) Four polytunnels in Kent field, noted by red colour; (centre) the original topological map; (right) node usage with deep blue colour indicating high demand.

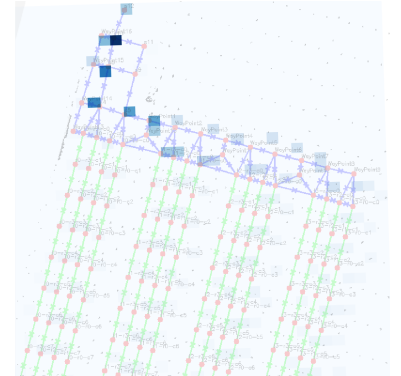
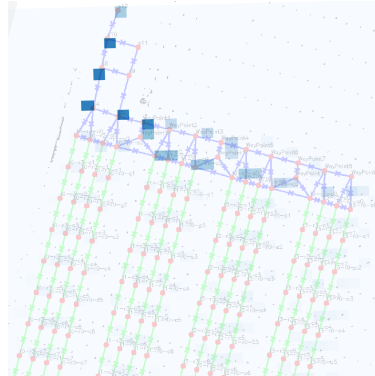
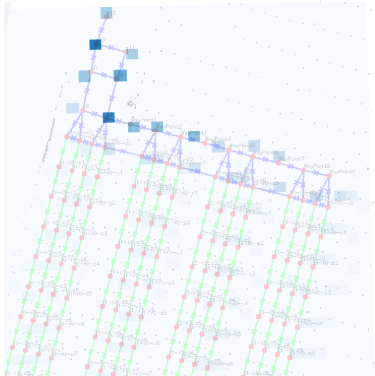
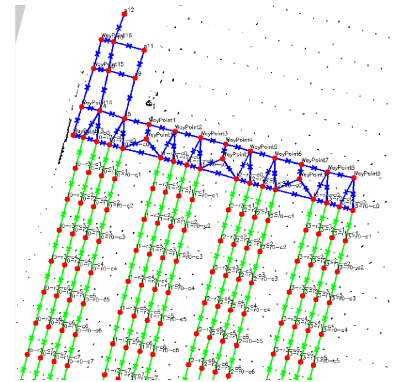
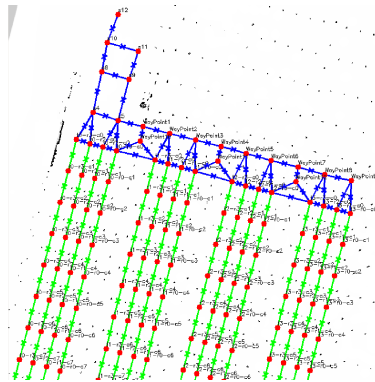
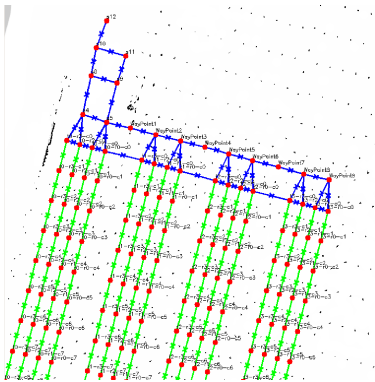


Figure 16: Modified Kent 4 polytunnels maps on top and corresponding node usage on bottom: (left) With parallel head-lanes (blue lines); (centre) topological map with parallel head-lanes and more robot waiting stations between the head-lanes; (right) parallel lanes near the drop-off/storage node along with parallel head-lanes and more base stations.

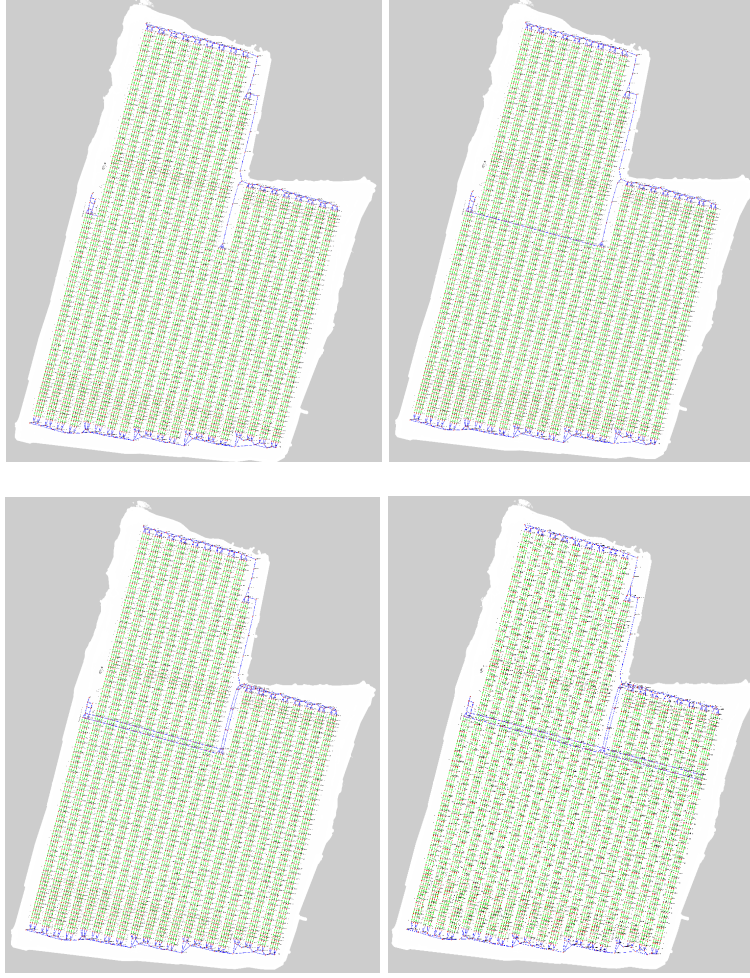


Figure 17: (Left-top) no cross path in the middle; (right-top) single-lane cross path in the middle to the centre of the field; (left-bottom) double-lane cross path in the middle to the centre of the field; and (right-bottom) double-lane cross path in the middle across the full field. All topological maps have parallel head-lanes at both ends.

Table 3: Logistics performance under different cross-path strategies. 2 pickers and 2 robots.

Cross path lane across field	simulated_time (s)	simulation_time (s)
None	109161.8 \pm 829.8	794.7 \pm 37.4
Single lane to the centre	75489.3 \pm 525.0	502.0 \pm 32.2
Double lane to the centre	66606.2 \pm 303.7	295.5 \pm 12.4
Double lane across the full field	60443.6 \pm 255.7	147.6 \pm 5.5

In Fig. 17, different cross lanes were used in the row middle so the robots do not have to travel along the long full rows to travel from one row to another. The results are summarised in Table 3. From these, it can be seen that adding cross lanes significantly reduces the robot service distance and thereby reducing traffic congestion and improving overall logistic efficiency. With a single lane across from the left side to the centre of the field resulted in a reduction of about 42% in process time. However, when using the double lane across full field strategy, the system has the fastest process time, with a reduction of about 54%. This large performance improvement is because the top-right rows of the field become noticeably short and the robots travel more often on the double lanes reducing traffic congestion.

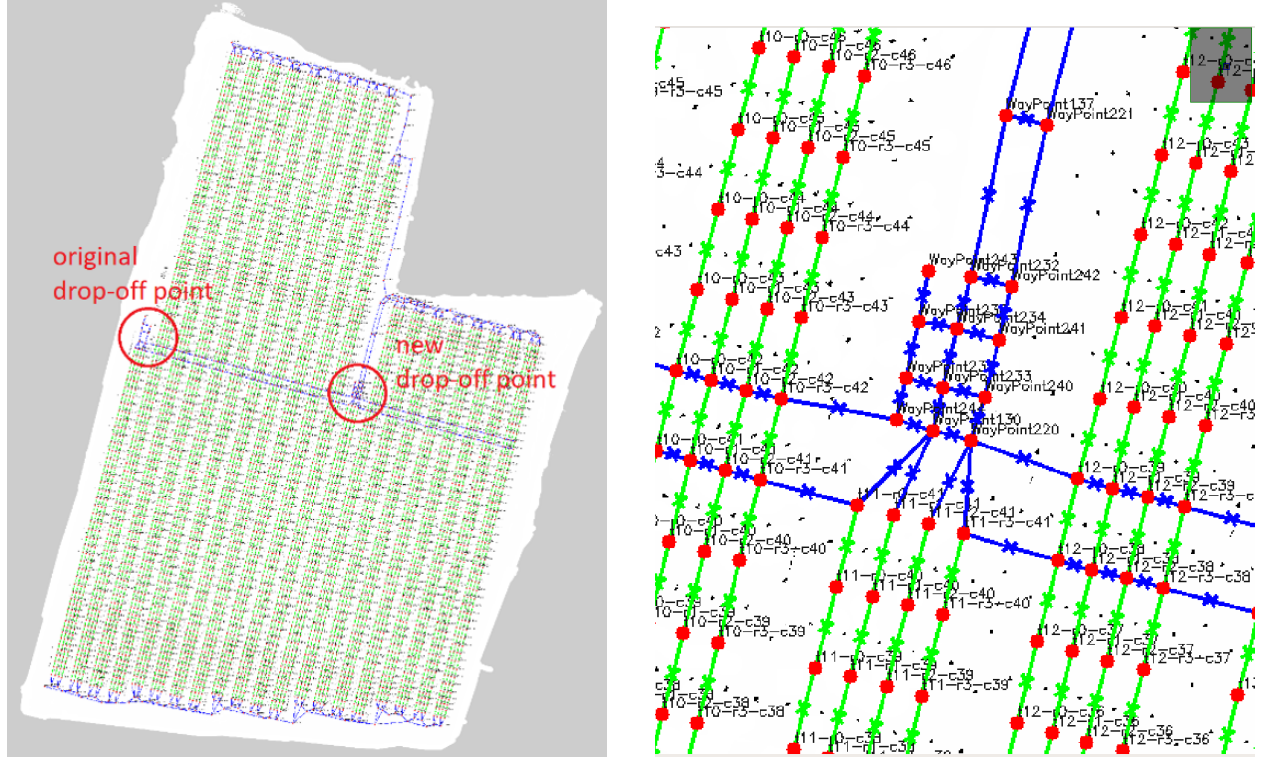


Figure 18: (Left) Modification of the full Kent map with double-lane cross-path and additional drop-off points; (right) Zoomed in details of the nodes and edges near the new drop-off point (the leaf node).

Table 4: Logistics performance under the different number of drop-off points strategy. 2 pickers and 2 robots.

Drop-off points	simulated_time (s)	simulation.time (s)
Single point (Left)	60318.6±311.7	149.2±5.1
Single point (Centre)	53140.6±620.9	113.1±6.2
Two points (Left and centre)	58669.2±352.4	100.6±2.0

In Fig. 18, different storage locations are used to see which location is better as a drop-off point. The original drop-off point is located at the middle of the field's left edge, and then a new drop-off point is added to the middle free space of the field. Ten trials of full DES were run with the modified map. Results in Table 4 denote that having two drop-off points substantially reduced the traffic congestion but is not as effective as having one drop-off point in terms of process time. The reason could be that a robot may select a faraway drop-off point, if the closest one is used by another robot, resulting in longer routes.

Looking at the results from both sets of modifications in Table 3 and Table 4 it is evident that:

- cross lanes help to reduce traffic congestion and improve overall logistic efficiency
- two drop-off points significantly reduce traffic congestion

6 Conclusions

In this paper, we propose a topological map to be used as a unifying representation of the system’s overall state which includes the location of all actors (humans and robots), and a computational model to facilitate both the orchestration of a fleet of robots and their respective coordination. The practical use of the proposed framework was demonstrated in a horticultural use-case with a fleet of robots supporting the work of human fruit pickers. The critical components of the system were analysed and evaluated in deployment in both realistic digital twin and real-life soft fruit farms of different scales, demonstrating the scalability and effectiveness of the proposed framework. Such a flexible representation allows for the rapid deployment and optimisation of heterogeneous robotic fleets in complex scenarios which are of great importance in the future success of robotic systems in agriculture. But the presented framework is more general and should be easy to adopt in other multi-robot/multi-human scenarios such as warehouse logistics, cleaning and maintenance of public spaces, etc.

The topological map representation discretises the environment to discrete nodes and edges connecting them, making it highly suitable for (i) reducing the spatial state space which in turn reduces the complexity of any planning within that state space and (ii) representing the topological features and constraints of the environment which can then be easily incorporated into navigation and task planning. Firstly, we demonstrated the exploitation of these features in an improved localisation of human agents into a finite state space consisting of the topological nodes, using observations from a low-cost GNSS receiver and by incorporating the topological constraints of the environment and the distributed sensing by the robots into the TPF. This accurate localisation on the topological map opens up immense opportunities to deploy robotics fleets into open-field environments. However, topological localisation could be also implemented in an indoor environment, by replacing the GNSS with other modalities of indoor localisation technologies. Secondly, we demonstrated that the topological map and localisation can be the backbones of a fully autonomous robotic fleet by deploying an in-field logistic support system in environments with different sizes and complexities. This system incorporates different layers of multi-robot coordination strategies in a human shared workspace such as topological navigation of individual robots, robotic fleet management, task management and fleet traffic management (see Sec. 3.5 for intricate details). Moreover, we also showed that such a system can be adapted for dynamic changes by modifying task definitions based on agent state changes (e.g. updating the location of an already assigned collection task based on the picker location). Thirdly, we showed that the topological map framework is a tailor-made environment descriptor for running DES to analyse the effect of topology modifications in the overall process while deploying an autonomous robotic fleet. We demonstrated this by looking at different modifications applied to three environments, starting from a small research facility to a large field used for commercial soft-fruit production.

The discretisation of the environment, especially the number of nodes in an operational route, however, has increasing computational complexities in route planning and centralised traffic management when the number of nodes is increased, and decreases the spatial resolution of operations when the number of nodes is decreased. The robotic fleet was tested only in four tunnels of a commercial soft-fruit production site. However, when deployed on the full farm-scale, the centralised fleet management will be a single point of failure and will have high computational complexity with high spatial coverage of the topological map and a large number of agents. In future, we will investigate the development of decentralised fleet management strategies to address these challenges. The topological modifications with the full process simulated in DES have a high potential to design new facilities, or to modify an existing facility to accommodate autonomous robots operating efficiently bringing in overall process improvements. The topology modification strategies tested in this paper, however, are limited to specific test cases. In future, we will also investigate automating these modifications and evaluation processes to incorporate a wider range of strategies and to pick the best set of modifications automatically for a given environment.

Acknowledgements

This work is part of the RAS-Berry (Robotic and Autonomous Systems for Berry Production, <https://rasberryproject.com/>) research programme, partly funded by Innovate UK under grants 103339, 105151 and 51367. This work is also partly supported by funding from Research England (E3) and Berry Garden Growers.

References

- ANYbotics (2023). Install inspection robot - anybotics. <https://www.anybotics.com/install-inspection-robot/#accurate-navigation>.
- Bac, C. W., Henten, E. J., Hemming, J., and Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, 31(6):888–911.
- Bachche, S. (2015). Deliberation on design strategies of automatic harvesting systems: A survey. *Robotics*, 4(2):194–222.
- Bakker, T., van Asselt, K., Bontsema, J., Müller, J., and van Straten, G. (2011). Autonomous navigation using a robot platform in a sugar beet field. *Biosystems Engineering*, 109(4):357–368.
- Ball, D., Ross, P., English, A., Milani, P., Richards, D., Bate, A., Upcroft, B., Wyeth, G., and Corke, P. (2017). Farm workers of the future: vision-based robotics for broad-acre agriculture. *IEEE Robot. Autom. Mag.*, pages 97–107.
- Baxter, P., Cielniak, G., Hanheide, M., and From, P. (2018). Safe human-robot interaction in agriculture. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, HRI '18*, pages 59–60, New York, NY, USA. ACM.
- Bazzan, A. L. and Klügl, F. (2014). A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3):375–403.
- Bellotto, N., Cosar, S., and Yan, Z. (2018). *Human Detection and Tracking*, pages 1–10. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Bellotto, N. and Hu, H. (2007). Multisensor data fusion for joint people tracking and identification with a service robot. In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1494–1499.
- Bellotto, N. and Hu, H. (2010). Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters. *Autonomous Robots*, 28:425–438.
- Binch, A., Das, G., Fentanes, J., and Hanheide, M. (2020). Context Dependant Iterative Parameter Optimisation for Robust Robot Navigation. In *Proc. 2020 IEEE International Conference on Robots and Automation (ICRA)*.
- Blanco, J.-L., Fernandez-Madrigal, J.-A., and Gonzalez, J. (2008). Toward a Unified Bayesian Approach to Hybrid Metric–Topological SLAM. *IEEE Transactions on Robotics*, 24(2):259–270.
- Bochtis, D., Sørensen, C., and Vougioukas, S. (2010a). Path planning for in-field navigation-aiding of service units. *Computers and Electronics in Agriculture*, 74(1):80–90.
- Bochtis, D. D., Sørensen, C. G., and Vougioukas, S. G. (2010b). Path Planning for In-Field Navigation-Aiding of Service Units. *Comput. Electron. Agric.*, 74(1):80–90.
- ClearpathRobotics (2022). Web ui autonomous mode - outdoornav software. https://clearpathrobotics.com/assets/manuals/outdoornav/archived/0.6.0/ui_autonomous_mode.html.

- Conesa-Muñoz, J., Bengochea-Guevara, J. M., Andujar, D., and Ribeiro, A. (2015). Efficient distribution of a fleet of heterogeneous vehicles in agriculture: A practical approach to multi-path planning. In *Proc. 2015 IEEE Int. Conf. Auton. Robot Syst. Compet. ICARSC 2015*, pages 56–61.
- Das, G., Cielniak, G., From, P. J., and Hanheide, M. (2018). Discrete Event Simulations for Scalability Analysis of Robotic In-Field Logistics in Agriculture – A Case Study. In *IEEE International Conference on Robotics and Automation, Workshop on Robotic Vision and Action in Agriculture*, Brisbane, Australia.
- Duckett, T., Pearson, S., Blackmore, S., and Grieve, B. (2018). *Agricultural Robotics: The Future of Robotic Agriculture*. UK-RAS White Papers. UK-RAS Network.
- FawkesRobotics (2018). fawkesrobotics/fawkes_msgs: Ros messages for fawkes integration. https://github.com/fawkesrobotics/fawkes_msgs.
- Feng, G., Qixin, C., Yongjie, C., and Masateru, N. (2008). Fruit location and stem detection method for strawberry harvesting robot. *Transactions of the Chinese Society of Agricultural Engineering*, 10.
- Gebresenbet, G. and Bosona, T. (2012). Logistics and Supply Chains in Agriculture and Food. In Groznik, A., editor, *Pathways to supply Chain Excell.*, chapter 8, page 218. InTech.
- Glas, D. F., Kanda, T., Ishiguro, H., and Hagita, N. (2009). Simultaneous people tracking and localization for social robots using external laser range finders. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 846–853.
- Glas, D. F., Morales, Y., Kanda, T., Ishiguro, H., and Hagita, N. (2015). Simultaneous people tracking and robot localization in dynamic social spaces. *Auton. Robots*, 39(1):43–63.
- Gozde Defterli, S., Shi, Y., Xu, Y., and Ehsani, R. (2016). Review of robotic technology for strawberry production. *Applied Engineering in Agriculture*, 32:301–308.
- Grimstad, L. and From, P. J. (2017). The Thorvald II Agricultural Robotic System. *Robotics*, 6.
- Grimstad, L., Pham, C., Phan, H., and From, P. (2015). On the design of a low-cost, light-weight, and highly versatile agricultural robot. In *Proc. Int. Work. Adv. Robot. its Soc. Impacts*, pages 1–6.
- Gritti, A. P., Tarabini, O., Guzzi, J., Di Caro, G. A., Caglioti, V., Gambardella, L. M., and Giusti, A. (2014). Kinect-based people detection and tracking from small-footprint ground robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4096–4103.
- Hanheide, M., Göbelbecker, M., Horn, G. S., Pronobis, A., Sjöö, K., Aydemir, A., Jensfelt, P., Gretton, C., Dearden, R., Janicek, M., Zender, H., Kruijff, G.-J., Hawes, N., and Wyatt, J. L. (2017). Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 247:119–150.
- Harman, H. and Sklar, E. I. (2021). A practical application of market-based mechanisms for allocating harvesting tasks. In Dignum, F., Corchado, J. M., and De La Prieta, F., editors, *Advances in Practical Applications of Agents, Multi-Agent Systems, and Social Good. The PAAMS Collection*, pages 114–126, Cham. Springer International Publishing.
- Imperoli, M., Potena, C., Nardi, D., Grisetti, G., and Pretto, A. (2018). An effective multi-cue positioning system for agricultural robotics. *IEEE Robotics and Automation Letters*, 3(4):3685–3692.
- Jensen, M. A. F., Bochtis, D., Sørensen, C. G., Blas, M. R., and Lykkegaard, K. L. (2012). In-field and inter-field path planning for agricultural transport units. *Computers & Industrial Engineering*, 63(4):1054–1061.
- Khan, M. W., Das, G. P., Hanheide, M., and Cielniak, G. (2020). Incorporating spatial constraints into a bayesian tracking framework for improved localisation in agricultural environments. In *Proc. IROS*, pages 2440–2445.

- Khan, M. W., Salman, N., Ali, A., Khan, A. M., and Kemp, A. H. (2015). A comparative study of target tracking with kalman filter, extended kalman filter and particle filter using received signal strength measurements. In *2015 International Conference on Emerging Technologies (ICET)*, pages 1–6.
- Kondo, N., Monta, M., and Arima, S. (1998). Strawberry harvesting robot on hydroponic system. *IFAC Proceedings Volumes*, 31(5):181 – 185. 3rd IFAC Workshop on Artificial Intelligence in Agriculture, Makuhari, Japan, 24-26 April 1998.
- Lacerda, B., Faruq, F., Parker, D., and Hawes, N. (2019). Probabilistic planning with formal performance guarantees for mobile service robots. *The International Journal of Robotics Research*, 38(9):1098–1123.
- Li, X., Zhang, Y., Marsic, I., and Burd, R. S. (2017). Online people tracking and identification with rfid and kinect. *ArXiv*, abs/1702.03824.
- Madridano, Á., Al-Kaff, A., Martín, D., and de la Escalera, A. (2021). Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications*, 173:114660.
- Minyoung Kim, Kumar, S., Pavlovic, V., and Rowley, H. (2008). Face tracking and recognition with visual constraints in real-world videos. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.
- Moore, T. and Stouch, D. (2014). A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer.
- open rmf (2023). Ros package: rmf.traffic.editor. https://index.ros.org/p/rmf_traffic_editor/.
- OpenStreetMap-ROS (2021). route_network - ros wiki. http://wiki.ros.org/route_network.
- Park, U., Choi, H., Jain, A. K., and Lee, S. (2013). Face tracking and recognition at a distance: A coaxial and concentric ptz camera system. *IEEE Transactions on Information Forensics and Security*, 8(10):1665–1677.
- Pedersen, S. M., Fountas, S., Have, H., and Blackmore, B. (2006). Agricultural robots—system analysis and economic feasibility. *Precision Agriculture*, 7:295–308.
- Peng, C. and Vougioukas, S. G. (2020). Deterministic predictive dynamic scheduling for crop-transport co-robots acting as harvesting aids. *Computers and Electronics in Agriculture*, 178:105702.
- Polvara, R., Del Duchetto, F., Neumann, G., and Hanheide, M. (2021). Navigate-and-Seek: A Robotics Framework for People Localization in Agricultural Environments. *IEEE Robotics and Automation Letters*, 6(4):6577–6584.
- Rajendra, P., Kondo, N., Ninomiya, K., Kamata, J., Kurita, M., Shiigi, T., Hayashi, S., Yoshida, H., and Kohno, Y. (2009). Machine Vision Algorithm for Robots to Harvest Strawberries in Tabletop Culture Greenhouses. *Engineering in Agriculture, Environment and Food*, 2(1):24–30.
- Ranganathan, A. and Dellaert, F. (2011). Online probabilistic topological mapping. *The International Journal of Robotics Research*, 30(6):755–771.
- Ryan, M. R. K. (2008). Exploiting subgraph structure in multi-robot path planning. *Journal of Artificial Intelligence Research*, 31:497–542.
- Schriber, T. J., Brunner, D. T., and Smith, J. S. (2017). Inside discrete-event simulation software: how it works and why it matters. In *Proc. 2017 Winter Sim. Conf.*, Las Vegas, NV, USA.
- Seyyedhasani, H., Peng, C., Jiunn Jang, W., and Vougioukas, S. G. (2020a). Collaboration of human pickers and crop-transporting robots during harvesting – part i: Model and simulator development. *Computers and Electronics in Agriculture*, 172:105324.

- Seyyedhasani, H., Peng, C., Jiunn Jang, W., and Vougioukas, S. G. (2020b). Collaboration of human pickers and crop-transporting robots during harvesting – part ii: Simulator evaluation and robot-scheduling case-study. *Computers and Electronics in Agriculture*, 172:105323.
- Street, C., Pütz, S., Mühlhig, M., Hawes, N., and Lacerda, B. (2022). Congestion-aware policy synthesis for multirobot systems. *IEEE Transactions on Robotics*, 38(1):262–280.
- Vivaldini, K. C., Galdames, J. P., Bueno, T. S., Araújo, R. C., Sobral, R. M., Becker, M., and Caurin, G. A. (2010). Robotic forklifts for intelligent warehouses: Routing, path planning, and auto-localization. In *2010 IEEE International Conference on Industrial Technology*, pages 1463–1468. IEEE.
- Wenmiao Lu and Yap-Peng Tan (2001). A color histogram based people tracking system. In *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)*, volume 2, pages 137–140 vol. 2.
- Wurman, P. R., D’Andrea, R., and Mountz, M. (2008). Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. *AI Mag.*, pages 9–20.
- Yan, Z., Duckett, T., and Bellotto, N. (2017). Online learning for human classification in 3d lidar-based tracking. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 864–871.
- Zhu, Z., Das, G., and Hanheide, M. (2023). Autonomous topological optimisation for multi-robot systems in logistics. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC ’23*, page 791–799, New York, NY, USA. Association for Computing Machinery.