Evaluation of the performance of PoultryBot, an autonomous mobile robotic platform for poultry houses

Vroegindeweij, B. A., Blaauw, S. K., IJsselmuiden, J. M. M., & van Henten, E. J.

Evaluation of the performance of PoultryBot, an autonomous mobile robotic platform for poultry houses

Bastiaan A. Vroegindeweij[1,2], Sam K. Blaauw[2], Joris IJsselmuiden[2] and Eldert J. van Henten[2]

[1]Livestock Robotics, Ochten, 4051 DB, the Netherlands

[2]Farm Technology Group of Wageningen University, Wageningen, 6708 PB, the Netherlands.

Corresponding author: Bastiaan Vroegindeweij, bastiaan@livestockrobotics.nl

Abstract

Assessment of animal status, housing conditions and manually collecting floor eggs are the major daily tasks for poultry farmers. To assist the farmer in these tasks, PoultryBot, an autonomous mobile robot for use in poultry houses has been developed. In earlier research, several components of PoultryBot were discussed in detail. Here, performance of the robot is evaluated under practical conditions. For navigation, different paths were used to assess its navigation performance for various tasks, such as area sweeping and surveying close to walls. PoultryBot proved capable of navigating autonomously more than 3000 m, while avoiding obstacles and dealing with the hens present. The robustness of its navigation performance was tested by confronting PoultryBot with obstacles in different positions with respect to its path and using different settings of the navigation parameters. Both factors clearly influenced the driving behaviour of PoultryBot. For floor egg collection, detection and collection of eggs was assessed at 5 predefined egg positions lateral to the path of the robot. Over 300 eggs were tested; 46% were collected successfully, 37% was not collected successfully, and 16% were missed. The most observed failures occurred when the collection device was just next to the egg. It is thought that this problem can be solved by improving the control algorithm. The results demonstrate the validity of the PoultryBot concept and the possibility of autonomous floor egg collection in commercial poultry houses. Furthermore, they indicate that application of smart autonomous vehicles in dense animal environments is feasible.

Key words: Mobile robot; Poultry farming; Autonomous navigation; Floor egg collection; Mobile monitoring; Performance evaluation;

Nomenclature

False Negative	FN
False Positive	FP
True Negative	TN
True Positive	TP
False Discovery Rate	FDR

1 Introduction

In an era where automation and the use of robots is increasing, opportunities arise also to take over the repetitive or dirty tasks currently found in livestock farming. One of the major daily tasks of every poultry farmer, is observing and checking the health and well-being of the animals, and making sure all housing and control systems function properly. Due to the increase of sizes of farms, the time available per animal for observational tasks has decreased. At the same time, with changes in housing the behavioural freedom for the animals has increased. This has led to an increased need for flock observation and management since animal status now has a larger impact on production. Having a mobile platform that moves autonomously among the animals all day long provides the poultry farmer with more and potentially more objective information about the animals and their environment.

Besides information gathering, there is a growing interest in using automatic floor egg collection in the modern animal-friendly loose housing systems adopted for laying hens. Floor eggs originate from animals that prefer to lay their eggs in some other location rather than the provided nest space. Based on extensive research (like Blokhuis & Metz, 1995; Froehlich & Oester, 2001; van Niekerk & Reuvekamp, 1997), farm management has improved significantly in recent decades. Combined with improved animal training, this has greatly reduced the number of floor eggs.

However, despite proper animal training and management, these floor eggs still account for 0.1% to 2% of the daily production. In extreme cases, the number of floor eggs can even increase to 5% to 10% of total egg production. In all cases, the required manual collection of these eggs puts a significant load on the daily activities of the farm staff (Blokhuis & Metz, 1995; Claeys, 2007).

As part of the research project “Automation for Poultry Production” at Wageningen University, the first autonomous poultry house robot (PoultryBot), was developed to help the poultry farmer in his daily work in the modern aviary poultry house. More specifically, floor egg collection was used as an example case in the development and evaluation of PoultryBot. For floor egg collection, PoultryBot should move freely throughout the whole poultry house, while being aware of its location in the house and the location of nearby obstacles. Furthermore, the robot should be able to detect and collect floor eggs, regardless of their location in the poultry house.

Several other applications exist where robots were freely acting in a complex environment, including interaction with dynamic objects such as humans, animals or plants. For example, the robots Rhino and Minerva acted as tour guides in museums (Burgard et al., 1999; Thrun et al., 2000), while Spencer guided passengers in an airport terminals (Triebel et al., 2015). In the agricultural domain, which is characterised by its complexity and limited structure (Nof, 2009), significant effort has been spent on autonomous robots for field work (Bakker, 2009; Deepfield Robotics, 2016; Hiremath, Evert, Heijden, Braak, & Stein, 2012) but also in orchards or greenhouses (Bac, van Henten, Hemming, & Edan, 2014; Bayar, Bergerman, Koku, & Konukseven, 2015; Shalal, Low, McCarthy, & Hancock, 2015a, 2015b). Several of the methods used in developing these robots can be considered as being useful for PoultryBot, such as the particle filter for localisation and the vision approaches used for fruit detection in horticulture (Bac, Hemming, & van Henten, 2013; van Henten et al., 2002). Their usefulness in the challenging environment of an aviary poultry house, however, still had to be proven.

With respect to livestock farming, some simple autonomous vehicles with fixed paths are used in dairy husbandry (Lely, 2015). In the domain of intensive animal production, a few research activities have been carried out, such as a student project at the KU Leuven, Belgium (Aertsen et al., 2012), some preliminary investigations on a mobile monitoring robot from Australia (Qi, Brookshaw, Low, & Banhazi, 2013; Haixa Qi et al., 2013) and a project on monitoring animal health and well-being using mobile and aerial robots at the Georgia Institute of Technology, USA. PoultryBot can be differentiated from previous examples of livestock robots by: 1) having more advanced systems for localisation and navigation, such that it can move freely and goal-driven throughout its environment; 2) being able to detect and interact with objects of interest; 3) being a test bed for an integrated system in a practical poultry house.

Previous work has introduced the concept of PoultryBot, and described and evaluated several of its main features. In Vroegindeweij, Ijsselmuiden, and van Henten (2016), a localisation system based on the particle filter approach (Thrun et al., 2000; Thrun, Burgard, & Fox, 2005) that originated from museum robot Minerva, was described and evaluated in a poultry house without hens. Vroegindeweij, van Willigenburg, Groot Koerkamp, and van Henten (2014) addressed the problem of path planning for the collection of floor eggs by presenting a new algorithm for non-uniform repetitive area coverage, when to the best of our knowledge, no such method existed at that time. Based on the use of multispectral features for fruit detection in harvesting robots from horticulture (Bac et al., 2013; van Henten et al., 2002), Vroegindeweij, van Hell, IJsselmuiden, and van Henten (2018) presented and tested an approach for the discriminating between the various object types in the poultry house that are relevant for the functioning of PoultryBot. Finally, in Vroegindeweij, Kortlever, Wais, and van Henten (2014), a description and evaluation of an actuator for floor egg collection was presented. Therefore, while individual aspects of this robotic system have been tested, to prove that the proposed concept and methods work, they have to be tested in an integrated manner under (near) practical conditions.

The objective of the current paper is to evaluate the performance of PoultryBot in a near practical environment. As an initial performance benchmark, a number of requirements for a future implementation of PoultryBot in commercial poultry houses can be indicated. Firstly, the robot should be able to operate autonomously, such that human intervention of the farmers are hardly required. To achieve this, it should drive collision-free through the poultry house, while being capable of handling various path types, such as traversing large areas to move from spot to spot or driving close to a wall to reduce floor laying in these areas. Furthermore, as object density in the poultry house is high and floor eggs can be found close to obstacles, PoultryBot should be able to closely approach obstacles without colliding with them. As PoultryBot's given path is task-oriented, this path should be followed as much as possible, with the freedom to avoid obstacles when required for safe navigation. Regarding its localisation, an error of < 0.1 m for 95% of the time is required to match the collected information to the correct physical location for mapping purposes. For floor egg collection, PoultryBot should detect at least 95% of the eggs present in its vicinity, with less than 5% of its detections being a false positive. Furthermore, all detected eggs within 1 m from PoultryBot should be collected, irrespective of their location.

To determine to which degree PoultryBot could comply with these requirements, PoultryBot's capabilities were evaluated under real-life conditions (including the presence of live animals) in a test environment similar to a commercial poultry house. Besides PoultryBot's performance, also the limitations and bottlenecks of the current approach were investigated.

2 Robot configuration

This section describes the configuration of PoultryBot used during the experiments. This also includes a description of the core methods used for localisation, path planning, object detection and navigation. Finally, the resulting behaviour of PoultryBot for navigation and egg collection is described. PoultryBot itself is shown in Fig. 1 during a test among hens.

Fig. 1: PoultryBot among hens in the test environment

2.1 PoultryBot platform

PoultryBot (Fig. 1) is based on the EyeSonic and SmartTrike field robots (Aelfers, van Esbroeck, van Hell, Raedts, & Russchen, 2015; SmartTrike, 2015; Wageningen University, 2009), and is about 1.1 m long, 0.45 m high. PoultryBot's width varies between 0.3 m at the rear and 0.55 m at the front and it is not symmetrical around its longitudinal axis. For stability and to overcome uneven and loose surfaces, PoultryBot has 3 driven pneumatic wheels, of which one is also steered, all controlled by two Roboteq AX3300 motor controllers (Roboteq Inc., Scottsdale, Arizona, USA). Individual wheel speeds were calculated using standard mobile robot kinematics (Siegwart et al., 2011) and controlled using an open control loop on a wheel power setpoint. For steering, a steering setpoint was used with feedback control using a potentiometer on the wheel orientation. To register the robot's behaviour and its environment, sensors including HEDL 5540 wheel encoders (maxon motor ag, Sachseln, Switzerland), an Xsens MTi-300 motion tracker (Xsens, Enschede, The Netherlands) a DMK 23UX174 camera (The Imaging Source LLC, Charlotte, North Carolina, USA), and a Sick LMS 111 laser scanner (Sick AG, Waldkirch, Germany) were mounted on the platform. The laser scanner was placed at 0.37 m above the ground. This position reduced the number of detections representing hens while at the same time the overall height of the platform remained within the height limitation of 0.45m imposed by the poultry house interior. Power was supplied by a set of batteries, with a 12 V d.c. pack for the electronics and sensing and a 24 V d.c. pack for the motors.

As the main task of PoultryBot is collecting floor eggs, a bent helical spring was mounted in front as collection device (see Fig. 1). A detailed explanation and performance evaluation of this device (with over 95% of the eggs successfully collected) is given in Vroegindeweij, Kortlever, et al. (2014). A drive motor was added to rotate the collection device, to both improve the collection results and to facilitate unloading. To increase manoeuvrability, a lifting mechanism was included to lift the collector when no eggs had to be collected. The collection device itself was controlled using a Roboclaw 2 x 15A motor controller (BASICMICRO.com, Temecula, California, USA), with a given lifting speed and feedback from a potentiometer on the position of the collection device. The control logic was given by a state machine that covered the current status and desired action of the collection device.

2.2 Software and control architecture

The on-board PC, running Windows 7 and NI LabVIEW 2013 (National Instruments, Austin, Texas, USA), was used to communicate with all peripherals, process incoming data and issue control commands. A distributed software architecture performed all acquisition, processing and sending of information in parallel, and always made the most recent data available for all processes. In this architecture, each physical task was performed by an independent node, like reading a sensor, processing data to obtain information, sending a control command or communicating with the user. This node was set to a fixed processing rate and obtained the latest data available from an internal data server. On overview of the organisation of the high-level nodes is shown in Fig. 2. Here, the top row in the figure contains all processing nodes, that receive data from the central data server (such as robot location and sensor information) and return their processing results (like robot location or navigation commands). The bottom row contains interfacing nodes that perform readout of sensors (receiving data from the laser scanner and the camera), write logdata and set action commands (like drive at a certain speed). All of this was coded using LabVIEW 2013, and most of these nodes contain several sub nodes for further organising the control process. Some computationally intensive operations (such as the raycast in the localisation method) were performed using a C++ library. Data acquisition and processing speed was set to 10 Hz for most nodes, except for those having a safety-critical task. These nodes ran faster, namely at 20 Hz. Furthermore, all data from all sensors was logged at 10 Hz, together with data like the estimated location and the speed commands for the wheels.

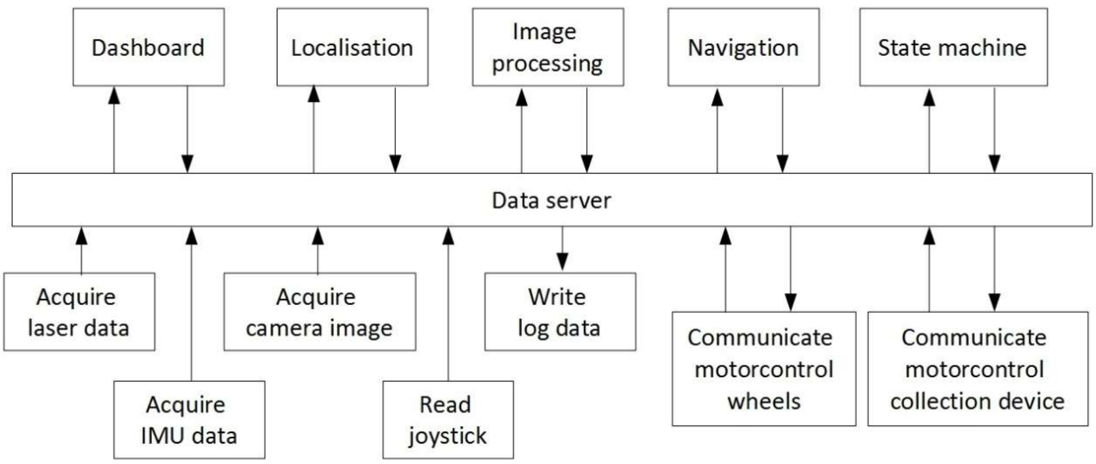

Fig. 2: Software architecture of the PoultryBot. Each block represents an individual node in the software. The top-levels blocks all do data processing for control purposes, the middle layer contains the data server while the lower level blocks represent interfacing with robot hardware.

2.3 Localisation method

Localisation of PoultryBot was achieved using a particle filter (Vroegindeweij et al., 2016), which was based on Thrun et al. (2005). In a particle filer, the pose of the robot is represented by a set of particles in the environment, with each particle containing a possible value for location and orientation of the robot. In each iteration of the algorithm, for each particle first a new position is predicted using information on the robot's displacement. This is followed by an update step, which evaluates for each predicted pose the correspondence between an actual measurement of the robot's environment applied to the predicted pose and a map of the robot's environment. Using the degree of correspondence as a measure of the likelihood of each particle's pose, a new set of particles is sampled from the current set of particles, which then describes the new pose estimate of the robot.

In our implementation, the prediction step used a combination of the best available information from odometry (encoder) data of all 3 wheels and the orientation data from the Xsens MTi 300. Data from all sources was checked on availability and reliability by discarding extreme readings and ignoring sources with missing data. The vehicle displacement and rotation were then determined separately for the rear wheels and the front wheel using standard mobile robot kinematics (Siegwart, Nourbakhsh, & Scaramuzza, 2011) and the change in compass orientation. If multiple data sources provided reliable data, a Kalman filter on displacement and rotation was used to fuse all reliable data into a single displacement prediction. During testing, this approach proved robust against internal communication failures and significant slip of individual wheels. The update step incorporated data from the Sick LMS 111 laser scanner. Data from the laser scanner were matched to a ray cast on a pre-defined map of the poultry house containing all fixed obstacles in the environment. As this method explicitly accounted for the possibility of random and shorter-than-expected distance readings, the presence of hens in the environment did not cause any problems. For the update step, the settings for the 'beam model' from Vroegindeweij et al. (2016) were used. Although these settings were based on a situation without hens in the environment, they showed the best performance during initial testing in the experimental environment with animals, and were therefore used in this research as well. For more details on the localisation system, see Thrun et al. (2005) and Vroegindeweij et al., (2016).

2.4 Path planning

In Vroegindeweij, van Willigenburg, et al. (2014), a method was described for coverage path planning for the collection of floor eggs. This non-uniform area coverage path planner was based on a dynamic programming approach and a map containing the probability of floor egg occurrence. The resulting path consists of a set of waypoints. As this path planning method did not account for the kinematics of the robot, the resulting paths contained waypoints 0.4 m apart, some of them connected by sharp turns (see Vroegindeweij, van Willigenburg, et al. (2014) for more details on the approach and results). Initial tests under practical conditions showed PoultryBot had severe difficulty in following such paths, and it was thus decided to use simpler, (manually defined) paths for the experiments in this work. These also consisted of waypoints, but now placed further apart while avoiding sharp turns. Adapting the method from Vroegindeweij, van Willigenburg, et al. (2014) and/or post-processing the path for this purpose is expected to produce paths the robot is capable of following and is a topic for future work. Details on the paths used in the experiments are given in the description of the experimental evaluation, in sections 4.1.1, 4.2.1, and 5.1.

To control the floor egg collection process, for each new egg found two additional waypoints are inserted to the waypoint list. The first is the position where the collection operation should start, while the second indicates the position where the collection operation should stop. Both waypoints are placed on the line between the robot's position and the location of the egg, the first at 0.3 m before the egg and the second at 0.3 m after the egg. If these new waypoints are placed within 0.3 m from an existing waypoint, the existing waypoint is replaced by the waypoints for egg collection.

2.5 Object detection

Detection of objects relevant for the functioning of PoultryBot was carried out at multiple levels, depending on the purpose. For navigation, (large) objects surrounding the robot were registered by the laser scanner. Subsequently, their locations with respect to the robot were fed into the navigation algorithm, and used in determining speed and steering commands, as described in section 2.5. For floor egg detection, the DMK 23UX174 monochrome camera with a 470 nm band pass filter attached to a lens with 5 mm focal distance was used. The image processing pipeline was based on (Vroegindeweij et al., 2018), with additional filtering on object shape and size to improve the performance for egg detection. The processing pipeline operated as follows. Using calibration images, first the vignette effect originating from the combination of lens and wavelength filter was corrected, using the method of (Zheng, Yu, Kang, Lin, & Kambhamettu, 2008). Next, pixels likely to correspond to eggs had the highest intensity values, so a high-pass threshold was applied. Using multi-stage morphological processing such as removing small particles and selecting particles that match the expected size (1200 - 5000 pixels), shape (Heywood circularity factor between 1.2 and 4) and position in the image of an egg (towards the lower half of the image), blobs expected to represent eggs were segmented. Finally, for each blob found its global position in the environment was determined using the pose estimate of the robot and a calibration of the camera based on the homography matrix (Dubrofsky, 2009; Wang, Hu, & Wu, 2004). For each detected egg, its estimated global position was used to control egg collection, by adding special waypoints for navigation during collection.

2.6 Navigation and driving

To convert the globally planned path (consisting of waypoints) into motions, while accounting for all obstacles present, the navigation method described by Schlegel (1998), was implemented. This method allows for close approximation of obstacles since it uses the exact robot contour, instead of the commonly used circular approximation. Furthermore, it considers both forward and backward movements and allows for online adaptation of the goal position when new target locations, such as waypoints for egg collection, emerge. Since PoultryBot is a relatively large and rectangular-shaped robot operating in a dense environment, such a method is needed to manoeuvre through narrow passages and collect eggs at all possible locations (including in corners and next to obstacles).

In the method of Schlegel, each combination of robot speed and steering angle that is allowed from a kinematic perspective is converted into a curvature that describes the related robot trajectory. A robot-based obstacle grid is then used to pre-calculate the available free space for each possible combination of obstacle location and allowable curvature. For each iteration of the navigation algorithm, the obstacle grid is filled with the current location of obstacles with respect to the robot. As the obstacle grid can be filled from any source, combining information from a map and distance sensors becomes a trivial task. For PoultryBot, the obstacle grid is filled with information on pre-defined map obstacles and the most recent reading of the laser scanner.

Next, given the current combination of obstacle presence, vehicle speed, and driving direction, the most suitable control option was selected from all curvatures that were allowed using a heuristic. For each allowed control option this heuristic weighed the normalised values for free distance ahead, heading towards the goal position, the closeness to obstacles and goal position, and speed. The highest weight was for "heading towards goal" (which was thus favoured most), followed by "free space" and "avoiding obstacles", while speed and goal approximation were less important. The result was a driving behaviour that tended to steer PoultryBot quite directly towards the target position, although sometimes it had difficulty avoiding obstacles, especially if the target position was further away. Navigation settings were the same for regular driving and egg collection, although speed during egg collection was lower.

2.7 PoultryBot's driving behaviour

Combining the elements for localisation, path planning, navigation, and object detection led to a driving behaviour that can be described as follows. After switching to autonomous mode at its start position, PoultryBot drove from waypoint to waypoint, which had to be passed within a given distance and in a specified direction. While driving, PoultryBot tried to avoid the obstacles present, based on mapped locations of fixed obstacles and distance readings towards obstacles from the laser scanner. Given its current position, the next waypoint and the information on obstacles positions, Poultrybot searched for the direction towards its goal which could be followed for the longest period of time. Although obstacles should be avoided, PoultryBot was allowed to approach them closely if they were densely present in the direction of target waypoint, as long as no collisions occurred. PoultryBot stopped driving at the last waypoint if no more waypoints were available, or when manually halted. A flowchart of this behaviour is shown in the left half of Fig. 3, representing the daily operation of PoultryBot. In the process, the major loop is responsible for following all waypoints, and this can cover an operational period of 10 to 15 h.

For egg collection, the same driving behaviour was used, but with additional steps for egg collection integrated into the driving behaviour, as shown on the right in Fig. 3. As stated earlier, if an egg was detected at a new location, two new waypoints were inserted to the waypoint list: one before and one after the egg. When PoultryBot reached the waypoint before the egg, driving was halted and the collection device was lowered. Next, PoultryBot slowly drove towards the waypoint after the egg, while rotating the collection device and attempting to collect the egg. If the waypoint after the egg was reached, driving was halted again and the collection device was lifted. Navigation then continued as before, until another egg was found. If eggs were close together or new waypoints were near existing waypoints, the waypoints controlling their collection were fused to simplify driving.

This behaviour does not cover all situations in practice, such as autonomously stopping or reversing direction if a collision is imminent. In the experiments, these situations were handled by switching from autonomous to remote control (i.e. a human operator controls the robot). In the case of a collision, and PoultryBot was reversed a small distance, while a steering correction was applied if required. After that, control was switched back to autonomous mode, allowing PoultryBot to continue the planned path by itself. If necessary, this procedure was repeated several times until PoultryBot moved around the obstacle, or if the situation could not be resolved by PoultryBot it was moved away from the obstacle by remote control.

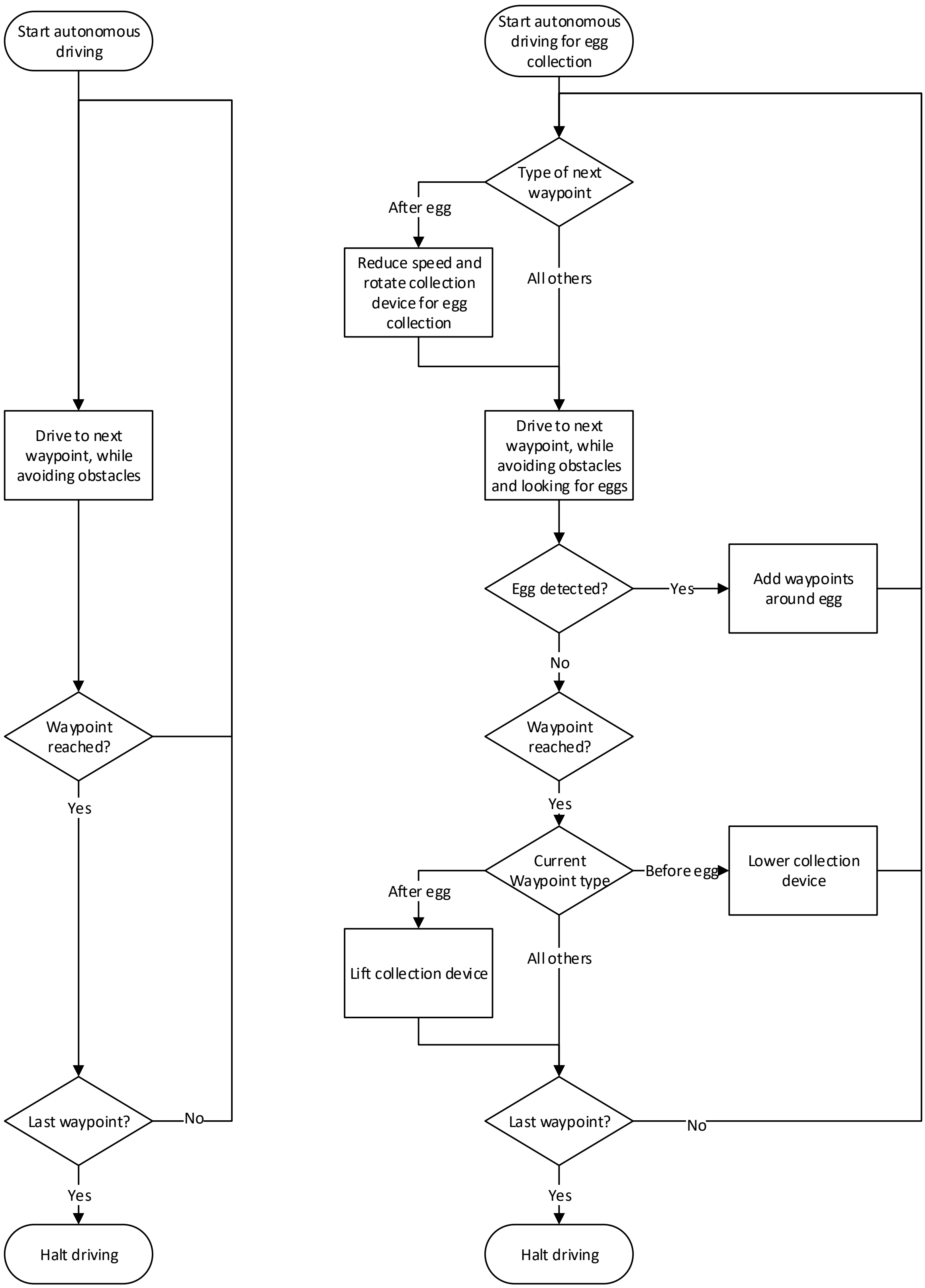

Fig. 3: Flowchart of PoultryBot's driving behaviour. On the left, general driving behaviour is shown. On the right half, driving behaviour for egg collection is displayed. Both diagrams cover the daily operation of PoultryBot, in which the major loop is continuously repeated over 10 to 15 h.

3 Experimental environment

PoultryBot's functional environment, a commercial aviary poultry house, has a number of specific characteristics relevant for correct functioning of a mobile robot. Firstly, it contains metal construction elements that provide facilities to the animals living there, and they act as a densely distributed, but fixed, set of obstacles, with elements sometimes no more than 1.2 m apart. Secondly, the housing interior is designed with the size of the animal in mind. Free space exists below interior elements and is used as part of the living area for the animals. This constrains the free height above the floor to less than 0.5 m. Thirdly, the uneven layer of loose litter on the floor hampers smooth driving. Fourthly, enrichment objects like roughage bins or pecking blocks are obstacles scattered around. Fifthly, the remaining free space is cluttered numerous animals that move around at will. All this clearly influences PoultryBot's sensing systems and navigation behaviour. Finally, the air contains high concentrations of dust, vapour and ammonia. All these influences can adversely affect the functioning of both robot hardware and sensing methods. The interior of a commercial poultry house is shown in the left part of Fig. 4.

Although PoultryBot has to work in a commercial poultry house, and several tests were carried out there, it was decided to perform the final tests and evaluation described here in a smaller and more open environment. This environment simplified the testing and experimental evaluation of performance as settings and conditions could be varied more easily, while allowing a better view of PoultryBot's behaviour and easier assessment of errors. In an area of 10 x 7 m, surrounded by 3 concrete walls and 1 wooden fence, 2 rows of housing interior were simulated with a wooden construction. In this area, 150 white laying hens (Dekalb White) were housed. Four feeder bins were placed, distributed over the area. Two drinker lines were placed on one of the interior rows, about 1 m away from the wall. Furthermore, below one end of this row, a laying nest was constructed. A picture of this environment is shown in the right of Fig. 4, while a schematic overview can be found in Fig. 5. Of the six challenging characteristics mentioned above, the first five were present:

- Main housing features such as construction poles and walls,
- Scattered objects such as feeder bins,
- Limited free height above the floor,
- Similar floor conditions with a layer of straw and litter on the floor,

- Animals (freely) occlude area, but at a lower density. However, as they were used to the robot and eager to approach it, they clearly affected PoultryBot's behaviour.

As a result, the conditions in this environment were representative for a commercial poultry house, especially from the point of view of operating a robot.

Fig. 4: Left: Interior of a commercial poultry house. Right: model of poultry house interior uses as experimental environment for testing and evaluating PoultryBot.

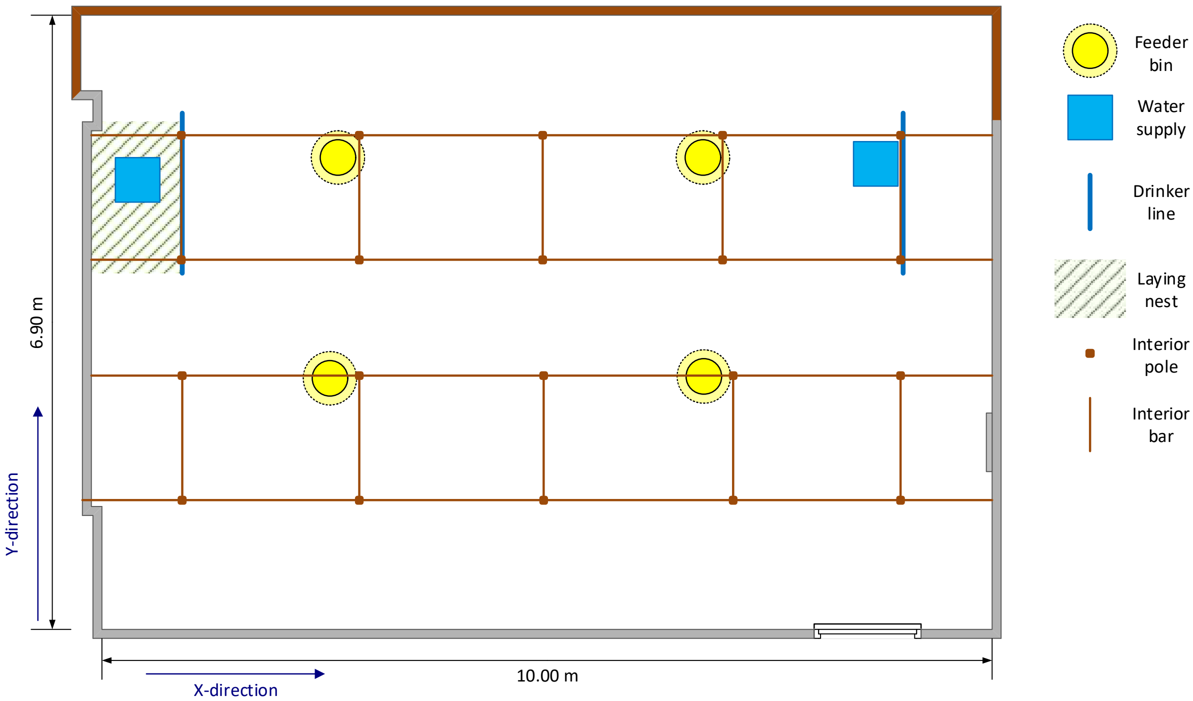

Fig. 5: Sketch of the test environment. Yellow circles indicate feeder bins. Blue blocks + lines indicate water supply and drinker lines, while the hatched area indicates the laying nest. The brown squares and lines indicate poles and housing bars. The test environment measured 10 m in X-direction and 6.90 m in Y-direction.

4 Evaluating autonomous navigation

Firstly, the autonomous driving capabilities of PoultryBot were evaluated in two experiments, to test navigation durability and navigation heuristic properties. To achieve this, robot performance was registered by logging robot data, such as position and speed, at a frequency of 10 Hz. Furthermore, an observer noted all relevant events, conditions and observations on driving behaviour. Each event or human intervention was given a reference number and its location, a description of the event (wall collision, hit pole) and the remotely controlled corrective action applied (continued driving, retracted and steered away) were noted. A human operated video camera (Sony DCR-SR78, Sony Corporation, Minato, Tokio Japan) was used to follow and record the behaviour of the robot, and it also registered all occurring events and comments made. Both the camera and observer were located on an elevated platform to provide a better overview of the scene, while the robot operator was present in the test environment. Based on this information, performance could be evaluated in detail and causes and possible solutions for current problems or bottlenecks could be identified.

4.1 Experiment 1: Navigation durability

In the first experiment, the navigation capabilities of PoultryBot were tested over an extended period of time and within the requirements set in section 1. Purpose of this experiment was to see how well the navigation performed under different conditions such as driving along a wall or traversing a large area. Also of interest were the kind of errors that occurred and if long-term operation would lead to a change of behaviour. To excite and evaluate the behaviour of PoultryBot, path segments with different shapes and structures were implemented. Furthermore, these path segments were repeatedly applied to see changes over time. It was expected that different path shapes and navigation conditions would show different robot performance with respect to the amount of control actions needed and the need for human interventions, but that the performance per path segment would remain constant over time.

4.1.1 Experimental outline

To identify changes due to long-term application, a closed tour was driven for a prolonged period of time, such that each path segment was traversed several times. The duration of the experiment was limited by available battery power. Since two sets of batteries were available, this experiment was executed twice, and the results will be referred to as the first and second test. Each test took about 1.5 h to complete and consisted of 12 full cycles of the given path. The given path was constructed by placing 30 waypoints and connecting these using straight lines. It contained 5 clearly different segments, each representing a specific type of condition encountered in practice: Segment 1; border navigation along the house wall (Blue). Segment 2; diagonally traversing the house (Red). Segment 3; sweeping the area, in lateral direction (Green). Segment 4; diagonally traversing the house (Purple). Segment 5. sweeping the area, in longitudinal direction (Yellow). The given path (or reference trajectory) with the individual segments are indicated in Fig. 6 with bold straight lines. Total length of the given path, measured as the Euclidian distance between the waypoints for a single cycle, was 94.2 m. So during each of tests one and two the robot was tested over a path of over 1100 m. For proper referencing in this experiment, PoultryBot's location was also tracked using a Trimble S6 Total Station (Trimble Navigation, Sunnyvale, California, USA), to assess the accuracy of the localisation method under these conditions. As using different path segments over extended timespans increases the chance of localisation failure, the accuracy of the localisation observed in this experiment also serves as upper limit for the localisation accuracy of PoultryBot in general.

4.1.2 Analysing performance

The navigation performance of PoultryBot was analysed using the robot’s log data and used to calculate several performance metrics, including:

- Path length, measured by the sum of the Euclidian distance between the robot's consecutive position estimates;
- Rotation, measured by the absolute sum of the differences in consecutive robot orientation estimates;
- Number of steering events, defined by the number of changes in the steering angle issued to the motor controller;
- Operational time, given by the amount of time PoultryBot was in autonomous mode or in remote control mode (e.g. controlled by a human operator).

Furthermore, also the observations of collisions and human interventions were analysed. All events were registered by the observer and divided into one of the following five categories to indicate the type of the intervention needed:

- Continued driving autonomously while touching an object;
- Human intervention using remote control, to reverse the PoultryBot after a collision;
- Human intervention using remote control, to steer PoultryBot away from the obstacle once, such that it was set free after a collision;
- Human intervention using remote control, to reverse and steer away once, to set PoultryBot free after a collision;
- Repeated human interventions using remote control to handle a collision.

In this experiment, PoultryBot's behaviour was evaluated at each path segment. To allow fair comparison between path segments, all performance metrics (except for distance) and interventions were divided by the autonomously driven distance in that segment averaged over all cycles. Statistical inference to investigate differences between path segments was done using an ANOVA, followed by a multiple comparison step using Fisher's protected LSD method in GenStat 18.1 (VSN International, Hemel Hempstead, UK).

4.1.3 Results and interpretation

Each cycle of the given path took between 6 to 8 min to complete, and covered a measured distance of around 100 m, which is longer than the Euclidian distance between the waypoints in the given path. In total, PoultryBot drove autonomously over 2400 m during this experiment. Figure 6 shows a selection of the position estimates generated during the first test. For readability, from this test every 10th estimated position is shown for 2 consecutive cycles to illustrate the driving behaviour of PoultryBot. The shortest path connecting the waypoints is displayed by straight lines. Video 1 (https://youtu.be/BSXoXR84cqg) shows PoultryBot behaviour in the same part of the experiment, and also contains an explanation on the events that occurred.

Figure 6 and Video 1 show that PoultryBot drove quite well from waypoint to waypoint, especially when sweeping in longitudinal direction (segment 3, yellow) and driving along the outside border (segment 1, blue). It can also be seen that occasionally, PoultryBot deviated clearly from the given path, such as in the lower-middle part of the area. Here, it correctly deviated from the given path to avoid a collision with the pole located there, and after negotiating this obstacle it returned to its given path. Also around segment 2 (red), similar behaviour was observed to avoid a collision with the feeder bin that was present on the first part of this segment. Also passing through narrow passages, such as between poles and walls at the left and right side of the area, did not present any difficulties for PoultryBot. This indicated the ability of PoultryBot to handle the presence of obstacles and variations in the environment. Furthermore, all path segments could be handled by PoultryBot, indicating its capability of dealing with the various types of conditions encountered in practice, as indicated in the introduction of this section. From the observations, the hens present in the environment appeared to have only a small effect on driving behaviour. They sometimes appeared in the laser data as obstacles, causing PoultryBot to avoid them by steering away from the hens, but this was without any changes in driving behaviour. In general, behaviour remained similar over time, suggesting that long-term application will not lead to an increase in navigation errors.

The localisation accuracy was evaluated over both tests (ground truth not shown). The Euclidian difference between positions estimated by PoultryBot and the reference measurement from the Total Station had a mean value of 0.127 m, with a 95 percentile of 0.319 m. The deviation in Euclidian distance was < 0.1 m for 63% of the time. This is an improvement on the results in Vroegindeweij et al. (2016), and this indicates that the desired accuracy indicated in that paper (<0.1 m for 95% of time) is achievable.

Logged human intervention and robot data are given in Table 1, as the mean value with standard deviation over all 24 cycles of the path. Values are separated into the 5 path segments. All metrics except for distance were corrected for the average distance driven to allow fair comparison. Also, having to drive longer distances than the waypoint distances is not necessarily negative, as this might indicate that PoultryBot deviated from its given path to avoid the obstacles present but it may also result from poor path tracking. Robot speed was similar for all segments (about 0.28 m s^{-1}) and is not included in Table 1.

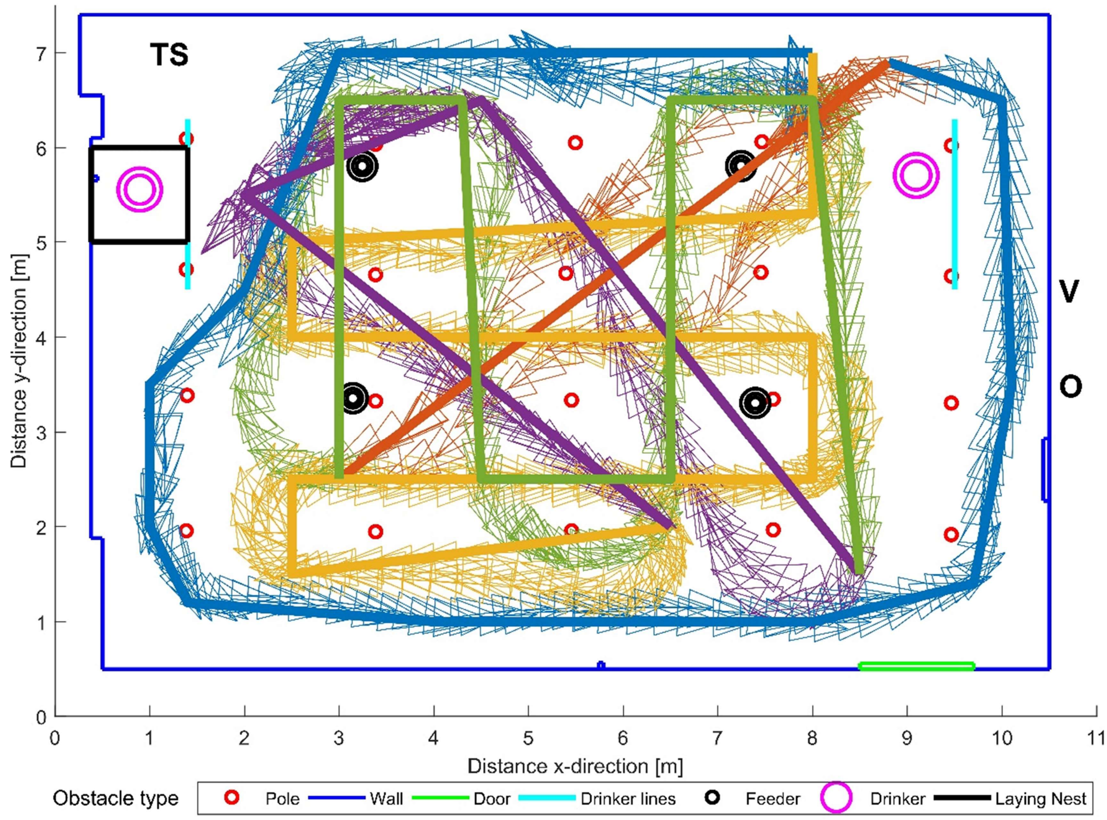

Figure 6: Result of two cycles from the first test of experiment 1. The triangles indicate each 10th estimated robot position, with the colour relating to the path segments. Blue is segment 1 with border navigation, orange and purple are diagonal traversals (segments 2 and 4), and green and yellow are longitudinal and lateral sweeping (segments 3 and 5). The straight lines indicate the shortest lines between the waypoints, with each colour representing a different path segment. TS indicates the position of the total station, V and O the positions of video camera and observer on the elevated platform.

Table 1: Quantitative results of experiment 1, testing long-term navigation performance. Numbers are average values over 24 cycles of the given path, with standard deviation in brackets. Measured values are as given, rotation data and

interventions are expressed respectively per metre and per 1000 m autonomously driven path length. Different superscripts indicate statistical difference at $p<0.05$ using Fisher's protected LSD.

Path segment	Measured distance, as mean with SD: Waypoint distance (m)	Autonomously driven(m)	Remotely controlled driven(m)	Rotations per m, as mean with SD: Waypoint rotation (radians)	Rotation (radians)	Steer events
1 Border navigation	26.3	26.1 (0.3)	0.2 (0.3)	0.27	0.56 (0.04) ^{b}	12.5 (1.0) ^{c}
2 Diagonal transect	7.3	8.0 (0.6)	0.8 (0.8)	0.13	0.50 (0.08) ^{a}	11.7 (1.7) ^{bc}
3 Lateral sweep	21.8	23.7 (0.8)	0.6 (0.6)	0.52	0.72 (0.05) ^{d}	12.1 (1.0) ^{c}
4 Diagonal transect	14.8	16.1 (0.5)	0.6 (0.7)	0.40	0.66 (0.05) ^{c}	10.8 (1.3) ^{a}
5 Longitudinal sweep	25.2	27.0 (1.1)	0.3 (0.6)	0.53	0.71 (0.04) ^{d}	11.2 (0.8) ^{ab}

Path segment	# Interventions per 1000 m, as mean with SD: Continue Driving	Retract	Steer	Retract + Steer	Multiple Interventions	Total	Total except Continue
1 Border navigation	25.5 (37.8) ^{b}	11.2 (20.6) ^{ab}	4.8 (12.7) ^{a}	6.4 (18.1) ^{a}	0.0 (0.0) ^{a}	47.9 (48.5) ^{b}	22.3 (26.9) ^{a}
2 Diagonal transect	5.2 (25.0) ^{a}	31.3 (54.2) ^{bc}	0.0 (0.0) ^{a}	31.3 (65.2) ^{b}	47.0 (60.6) ^{b}	114.8 (108.0) ^{c}	109.6 (97.8) ^{c}
3 Lateral sweep	8.8 (21.0) ^{a}	33.3 (34.3) ^{c}	0.0 (0.0) ^{a}	8.8 (21.0) ^{a}	1.8 (8.4) ^{a}	52.6 (42.5) ^{b}	43.9 (41.2) ^{ab}
4 Diagonal transect	7.7 (27.2) ^{a}	28.4 (35.7) ^{bc}	0.0 (0.0) ^{a}	20.7 (34.3) ^{ab}	7.7 (20.5) ^{a}	64.6 (52.1) ^{b}	56.8 (47.1) ^{b}
5 Longitudinal sweep	1.5 (7.4) ^{a}	6.2 (17.4) ^{ab}	1.5 (7.4) ^{a}	3.1 (10.2) ^{a}	7.7 (23.8) ^{a}	20.0 (30.2) ^{a}	18.5 (30.2) ^{a}

The length of given path differed between segments, with the diagonal transects (segments 2 and 4) was the shortest (7.3 and 14.8 m) and the border navigation (segment 1) the longest. Compared to this, driven distance was less than the length of the given path for the border navigation (i.e. 26.1 vs 26.3 m), while for other segments it was up to 10% longer than the given path. This difference might relate to the structure of the border navigation segment, where turns are always in the same direction and with a limited need to avoid obstacles. Thus, a slightly shorter path was used here, in contrary to the other segments, which contained more turns and obstacle-avoidance manoeuvres. For the first diagonal transect (segment 2), the distance driven by remote control was highest (0.8 m) compared to both the segment length and the other paths. Also the second diagonal transect (segment 4) and the lateral sweep (segment 5) had higher remotely controlled distance (0.6 m), which indicates that stronger human interventions were needed on these segments. The largest platform rotations were observed while sweeping longitudinally (segment 5, 0.71 rad m^{-1}) and laterally (segment 3, 0.72 rad m^{-1}), followed by the second diagonal transect (segment 4, 0.66 rad m^{-1}). A clear relation with path structure can be seen, as the given path also contained the largest rotations (0.5 rad m^{-1}), and the resulting rotation is only 30 to 40% higher than the rotations required to fulfil the given segment. When considering measured platform rotations compared to the required rotation for the given path, segment 2 has most rotations, being almost four times greater than needed for the given path. This increase might be partly attributed to the presence of a feeder bin at the start of this segment. In the number of steering events, no relation with path segment type was visible, indicating that larger rotations were more likely the result of distinct turns than from frequent small steering corrections.

Most interventions took place in the diagonal traversal (segments 2 and 4 with averages of 114.8 and 64.6 interventions per 1000 m), while the longitudinal sweeping (segment 3) had fewest interventions (20.0 interventions per 1000 m). This was most likely the result of the waypoints in segment 3 placed in between the housing poles, such that the path was merely obstacle-free, while the diagonal traversals required the explicit avoidance of obstacles. When excluding the 'continue' events, the lateral sweep (segment 5, 43.9 interventions per 1000 m) performed similar to the second diagonal traversal (segment 4, 56.8 interventions per 1000 m), while the border navigation (segment 1, 22.35 interventions per 1000 m) showed similar behaviour as the longitudinal sweep (segment 3, 18.5 interventions per 1000 m). Also here, the larger need for intervention at segments 4 and 5 compared to segments 1 and 3 might be related to the number of obstacles that were encountered by PoultryBot on these segments. Minor interventions such as 'continue' (keep driving while hitting obstacles) or 'steering away' were most seen during the border navigation segment (segment 1, 25.5 and 4.8 interventions per 1000 m respectively), possibly as result of collisions with the wall. Stronger interventions, such as 'reverse and steer away', were mainly found at diagonal traversals (segments 2 and 4), while 'reverse' actions were also found more in the lateral sweeping (segment 5) but with substantial variation between cycles. The most serious cases, where 'multiple' interventions were required to solve collisions, were mainly seen along the first diagonal transect (segment 2, 47.0 interventions per 100 m, $p<0.000$), which also had by far the highest number of interventions (114.8 interventions per 1000 m).

This high number of interventions can be explained from the navigation algorithm, where reaching the goal had a higher weight and thus obtained more attention than avoiding obstacles. This frequently led to collisions, especially if an obstacle was close-by on the robot's path to a waypoint further away. Such conditions were indeed present at the start of the first diagonal segment, with an obstacle (feeder bin) being present in the most likely path of PoultryBot towards its next waypoint at the other end of the test environment. Alternatively, if obstacles were further away from the lines that connected the waypoints, navigation was fairly easy and both collision occurrence and the need for human intervention was lower. Also the field-of-view of the obstacle sensor played a role here, since in a number of cases obstacle collisions were observed just after the object has left the detection area. The presented results indicate that PoultryBot could handle various path types, but that path structure, especially the placement of waypoints with respect to obstacles, influenced the results.

4.2 Experiment 2: Obstacle Handling

In experiment 1, obstacles placed on the shortest line between PoultryBot's position and its target waypoint frequently led to collisions. The second experiment therefore tested whether this relationship between obstacle location and waypoint placement indeed existed, by testing the effect of changing obstacle positions on the driving behaviour of PoultryBot. Furthermore, it was investigated if changing the settings of the navigation heuristic (as explained in section 2.5), especially for the "heading to goal" behaviour, would lead to fewer collisions.

4.2.1 Experimental outline

A straight path A (green, see Fig. 7) was defined by placing 2 waypoints in the longitudinal direction of the test environment, with construction elements on either side of the path (at a lateral distance of about 0.7 m to the path). To observe if driving behaviour changed when the path was closer to, and thus conflicted more with, these construction elements (obstacles), two additional paths B (orange) and C (red) were defined, also shown in Fig. 7. To realise this, the waypoints were moved twice, in steps of 0.4 m, towards the poles on one side of the path. In these cases, it was expected that either the robot would steer more to avoid the poles, or that the number of collisions would increase. Additional waypoints were used to allow autonomous return of the robot from the end point to the starting position thus allowing repeated execution of the path. This path is indicated by the purple line in Fig. 7.

After moving the waypoints, the weight factor for "heading to goal" (as defined in Section 2.5) in the heuristic was also varied in this experiment. As all other weight factors ranged between 0.001 and 0.5, changing the “heading to goal” weight factor from its original value of 3 to 1 and 2 was expected to lead to more steering and better collision-avoidance by PoultryBot. For “heading to goal” weight factor values 2 and 3, each path was repeated 6 times. For a weight factor value of 1, each input path was repeated only 3 times, as during the experiment no clear reduction in the need for human interventions was seen with respect to results obtained when using a weight factor value of 3. Performance was evaluated only for the track between the first and last interior poles in longitudinal direction, which were 8.07m apart. Performance evaluation was done in a similar fashion as described for experiment 1, section 4.1.2.

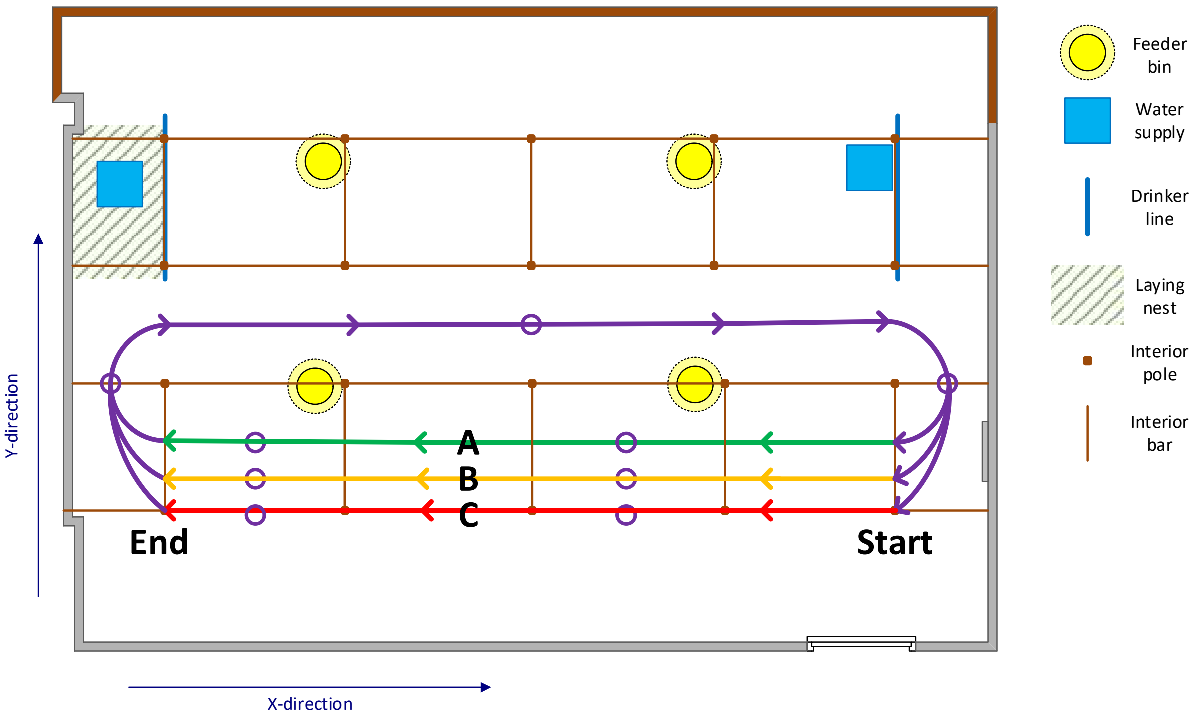

Fig. 7: Paths for experiment 2, to evaluate the effects of obstacle locations with respect to PoultryBot's path on the driving behaviour of PoultryBot. Waypoints (purple circles) were placed on varying positions A, B, and C with respect to the interior elements. The green, orange and red lines indicate the shortest path between Start and End along the waypoints for trajectories A, B and C. The purple line indicates the path used by PoultryBot for autonomous return from the end-point to the start-point.

4.2.2 Results and interpretation

In Video 2 (https://youtu.be/24g-XgALyqA), the first run of this experiment is shown on paths A, B and C with 3 repetitions each and using a weight factor value of 3. An overview of the results is given in Table 2, showing the mean and standard deviation of all data obtained from the robot's log file, grouped by weight factor and path. In Table 3, the type and number of interventions are given per combination of weight factor setting and given path.

Table 2: Autonomous driving results of experiment 2, evaluating the effect of changing the weight factor for "heading to goal"-behaviour and the obstacle placement with respect to PoultryBot's path on the navigation performance of PoultryBot. Data are presented per combination of given path and heuristic setting, and expressed as average with standard deviation over all repetitions.

	Weight factor	Path	Repetitions #	Distance *(m)* mean	SD	Rotation *(rad)* mean	SD	Steer events *(#)* mean	SD	Time *(seconds)* mean	SD
More goal-oriented	3	A	6	8.16	0.05	2.97	0.39	116.3	12.5	26.7	0.5
	3	B	6	8.31	0.10	3.62	0.61	110.8	11.0	28.0	0.9
	3	C	6	9.74	1.38	5.68	0.93	144.3	37.5	37.7	8.8
	2	A	5	8.74	0.60	4.80	1.50	122.6	6.0	32.0	6.0
	2	B	6	8.60	0.66	3.96	1.20	117.8	11.7	30.3	4.0
	2	C	6	9.28	0.44	5.29	0.83	128.5	19.8	35.3	5.2
Less goal-oriented	1	A	3	9.09	0.95	5.97	2.55	110.3	5.7	28.8	1.0
	1	B	3	9.99	1.72	7.21	1.02	160.7	46.6	35.9	8.9
	1	C	3	10.70	1.67	6.29	1.47	151.3	21.0	41.0	8.9

Table 3: Intervention results of experiment 2, evaluating the effect of changing the weight factor for "heading to goal"-behaviour and the placement of obstacles with respect to PoultryBot's path on the navigation performance of PoultryBot. Interventions are presented per combination of given path and heuristic setting, and given as total over all repetitions.

	Weight factor	Path	Repetitions #	Detour *# total*	Continued *# total*	Retract *# total*	Retract + Steer *# total*	Multiple *# total*	Total *# total*
More goal-oriented	3	A	6	0	0	0	0	1	1
	3	B	6	0	1	1	0	0	2
	3	C	6	2	1	2	2	4	11
	2	A	5	3	1	0	1	1	6
	2	B	6	3	0	2	0	0	5
	2	C	6	4	1	6	2	0	13
Less goal-oriented	1	A	3	3	0	1	0	0	4
	1	B	3	1	0	3	0	1	5
	1	C	3	3	0	3	1	0	7

Positioning the robot path such that it conflicted more with obstacles and thereby becoming more complex, increased path length and time required for path completion up to 20 percent (e.g. from 8.16 for path A to 9.74 m for path C when using weight factor 3, Table 2). Also the amount of steering increased, independent of the weight factor for the "heading to goal"-behaviour. This is visible in Table 2 for paths B and C showing more platform rotations and more steering events than path A for all values of the navigation weight factors. For the driven distance, this increase is also clearly significant, when comparing path A to B ($p = 0.002$) or C ($p = 0.009$), for all settings of the "heading to goal" weight factor. Furthermore, the number of interventions due to collisions increased clearly (Table 3), from 1 to 11 for weight factor 3 and from 6 to 13 for weight factor 2, if the obstacles were positioned more into the robot path. In the type of interventions no clear change can be observed, indicating the interventions to resolve collisions did not become more complex when obstacles were positioned more into the robot path. All of this matches with expectations and the results of the previous experiment, as obstacles on the way of PoultryBot will force it to steer around them, thus increasing time, distance and steering required, as well as the risk for collision. Furthermore, obstacles closer to the robot contour showed only a limited effect, whereas obstacles present in the middle of the robot path caused clear changes in results.

When modifying the weight factor for "heading towards goal" in the heuristic, a decrease of the weight factor seemed to produce longer robot paths (Table 2), although effects were smaller compared to that of changing the path. Already with no obstacles (except for hens) present in front of the robot (path A), path length increased with 10 to 15 % when changing the weight factor, e.g. from 8.16 to 8.74 m for a weight factor change from 3 to 2. The total rotation of PoultryBot (in radians) also showed a similar increase, whereas effects on the number of steer events and the required time were found to be less clear. Such behaviour seems logical, since a lower weight for "heading to goal" driving will be less target-oriented, and thus searches more for available free space between objects such as construction elements and hens present in PoultryBot's vicinity. If obstacles were present in the robot path (like path C), the effect of changing the weight factor was smaller and sometimes even opposite, as this path already required more steering. Still, a significant difference in path length was found between a weight factor of 3 and a weight factor of 1 (p=0.006) and 2 (p=0.016) for the "heading to goal"-behaviour. In terms of interventions (Table 3), it can be seen that lower weights lead to more detours, where PoultryBot instead of passing between the poles, drove around them. Also, lower weights seemed to require less interventions, especially for the 'multiple'-case. However, this trend was not consistent and sometimes an increase in the number of interventions was seen, so no firm conclusions can be drawn here.

These results showed that the position of obstacles on the path and the setting of weight factors for the heuristic had a clear influence on the driving behaviour of PoultryBot. Moving the robot path closer to obstacles increased both path length and steering behaviour and led to more collisions. Changing the heuristic settings to less goal-oriented behaviour led to longer paths with more steering and a tendency of having fewer collisions with obstacles. This indicates that further tuning of the navigation heuristic can be useful to improve the navigation performance of PoultryBot.

Furthermore, attention is needed for handling obstacle collisions. Currently, these are handled using remote control by the operator, but for autonomous operation these collisions either have to be avoided or solved autonomously. This requires for instance the implementation of automated collision detection and reverse driving behaviour, and possibly also adaptation of the navigation behaviour. Once such solutions are added, this will likely solve most or all of the cases that currently required human intervention, bringing PoultryBot closer to fully autonomous operation.

5 Evaluating egg collection performance

After testing PoultryBot's navigation, the egg collection performance was evaluated. In previous work (Vroegindeweij, Kortlever, et al., 2014) the collection device itself was evaluated in detail. This experiment assessed the egg collection capability of PoultryBot. Specific attention was given to the detection of the egg, the result of the collection operation and an analysis of the collection failures. As previous research indicated that the collection device had difficulty in collecting eggs located in corners, those locations were not considered in this experiment.

During egg collection, a video camera (Sony DCR-SR78) was positioned in line with the robot's path, to register PoultryBot's behaviour. Furthermore, a GoPro Hero 4 video camera (GoPro, San Mateo, California, USA) was mounted to the robot registered a close-up of the egg collection device. In the measurement notes, all egg detections and collection operations were registered. Subsequently, the location of the egg (if known) was registered, together with detection and collection results. Furthermore, relevant information like start time of a run, camera and algorithm settings, and specific behaviours and observations were noted as well. All this information was used to evaluate performance of detection and collection in more detail, but especially to indicate causes and possible solutions for current problems or bottlenecks.

5.1 Collection procedure

For the evaluation of floor egg collection, a repeatable procedure was used, based on a given path along one of the walls in the test environment. Each collection run was made along the wall in the longitudinal direction of the area, and consisted of 2 parts. In the first part, the wall was to the left of the robot, while in the second part the robot changed direction and had the wall on its right side. Between the two parts, the robot was turned around while driving using remote control by the operator. In each part of the collection run, 2 eggs were present, with the first one (longitudinal locations A and A') between the wall and the second pole, as seen from the start of this part, and the second one (longitudinal locations B and B') between the wall and the open space between the 3rd and 4th pole. To create a collection path that was similar for all eggs, the waypoints before and after each egg were placed in a straight line at about 0.7 m next to the wall and more than 1 m away from the egg.

Eggs were placed on a line perpendicular to this path, and five egg locations in lateral direction were individually tested: in front of the robot, on the robot's edges or outside the robot contour and close to the wall or poles. These lateral locations were numbered 1 to 5 when going from the wall to the pole. In each part of the run, both longitudinal locations (A and B) and a single lateral location were evaluated. Each combination of longitudinal and lateral location (indicated by a combination of the letter A or B and a number between 1 and 5) was repeated for at least 20 correctly detected eggs. In Fig. 8, an overview of the experiment is given, showing the waypoints, the egg's longitudinal (A or B) and lateral (1 to 5) locations, and the expected driving pattern (reference trajectory) of PoultryBot. Video 3 (https://youtu.be/QJxZXIyGEIc) shows 2 runs of the experiment on lateral location 3, i.e. in the middle of the path.

In preliminary research, egg orientation prior to collection showed limited effects on collection success, as the egg rotates during collection under influence of floor structure and the collection device. Therefore, egg orientation was fixed with the egg's major axis aligned with the direction of PoultryBot's given path and their minor end towards the robot. As the hens present during the experiment exhibited egg-eating behaviour, hard-boiled eggs were used to reduce egg eating in case of egg breakage.

Weather conditions outside the building influenced the light intensity in the experimental area, making the use of a single fixed setting of the camera and detection method impossible. Thus, before each measurement series, the camera gain and detection threshold were adapted to the amount of light present at that moment. In this way, the effect of ambient light on the detection and collection results was prevented as far as possible.

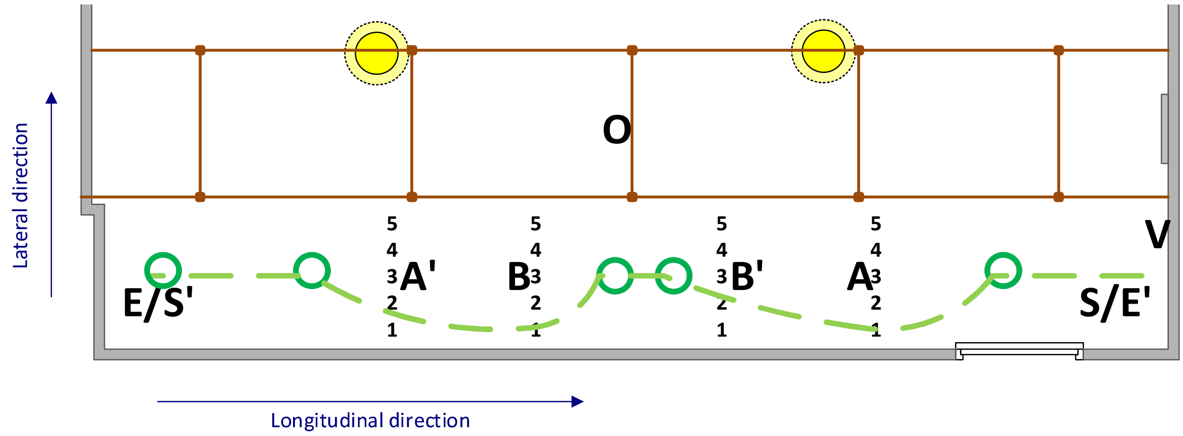

Fig. 8: Egg locations and waypoints used in the egg collection evaluation. The numbers 1 to 5 indicate the egg locations in lateral direction, while capitals A and B indicate the location of an egg in longitudinal direction. O indicates the location of the observer, V indicates the location of the video camera used for observation of the experiments. Green circles indicate waypoints in the first part of the run, going from the starting point at the right (S) to the end point on the left (E), with eggs located at A and B. The green dashed line indicates the expected robot behaviour for collection at lateral location 1. The second part of the run (not indicated) goes from S' to E', with eggs located on A' and B'.

5.2 Data registration and processing

During the egg collection experiments, results were registered separately for egg detection, collection operation, and collection failure. Egg detection potentially yielded one of the following results:

- False negative (FN), i.e. the egg was not detected, and therefore no collection operation was performed;
- False positive (FP), i.e. detected something else than an egg. A collection operation was started, but failed due to the absence of an egg;
- True positive (TP), i.e. egg correctly detected, and egg collection was started.

True negatives (TN) were not registered, as this would include every non-egg object seen by the robot. For an egg collection operation performed on correctly detected eggs (TP), the following options were considered as collection result:

- egg collected correctly;
- collection failure: collection started correctly, but the egg was not collected;

- wrong location of collection (robot was clearly off);
- no start of collection.

In case egg collection failed, one of the following causes for failure was assigned:

A. collection device ran over the egg without collecting it;
B. egg was broken by the collection device;
C. egg left the collection device after collection;
D. collection device was located just next to the egg;
E. collection device was lifted before actually reaching the egg;
F. collection device was lowered after passing the egg;
G. robot switched to remote control by the operator, as result of a collision with an obstacle.

If the collection operation was ended manually, an asterisk was added to the collection result, independent of the collection result itself. All results were registered by the observer during the experiment, and analysed afterwards as described below.

After the measurements, data from runs that produced unreliable responses due to an unsuitable combination of detection algorithm settings and varying ambient light conditions (see the last paragraph of Section 5.1) were excluded from further analysis, and results were clustered per longitudinal location (A or B) and lateral location (1 to 5). As false positives (FP) in detection could not be related to a specific egg or location, they were only assessed in relation to the number of eggs present. Based on this data, the following performance indices were calculated, which are similar to those used in fruit harvesting robots (Bac, 2015; Bac et al., 2017):

a. Egg detection success (%): the occurrence of a correct detection (TP) as % of the total number of eggs present (TP+ FN);
b. False discovery rate (FDR) for egg detection (%): the number of objects falsely detected as eggs (FP), as percentage of the total number of eggs present (TP + FN);
c. Collection success rate (%): the occurrence of each collection result as % of the number of correctly detected eggs (TP);
d. Collection failure rate (%): the occurrence of each failure type as % of total collection failures.

Statistical inference on the results (both raw data and performance indices) was carried out using an ANOVA, followed by a multiple comparison using Fisher's protected Least Significant Difference test in GenStat 18.1 to investigate differences in performance between locations.

Cycle time was not investigated, as it was determined by the fixed driving speed of the robot and the time required for lowering and raising the collection device. It hardly varied in the experiment and/or as result of actual conditions. For each egg, a single collection operation was done. The number of eggs tested varied between 25 and 40 per combination of longitudinal and lateral location, and is indicated in the results in Fig. 8. Egg damage rate was already investigated in earlier research (Vroegindeweij, Kortlever, et al., 2014), and not investigated in detail here as it was not significantly dependent on operation of the device.

5.3 Detection performance

Collection performance was evaluated using 313 eggs in total. Results for egg detection, given in Fig. 9, show that the majority of the eggs (86%) were properly detected by PoultryBot, although results were dependent on the lateral location of the egg. For clustered longitudinal locations A and B, Fig. 9 suggests that B-locations (even bars, 91% detected) have slightly more ($p=0.21$) eggs detected correctly compared to A locations (odd bars, 80% detected). As B-locations were more in the middle of the area and light intensity was slightly higher than at the A-locations, this might have positively affected detection rate.

Clear differences in performance can be observed between the lateral locations 1 to 5. In front of the robot (location 3) more than 90% of the eggs were detected correctly, whereas only 65 to 85% of the eggs further to the sides of the robot (locations 1 and 5) were detected correctly. Correct detection at location 5 indeed proved different from locations 2 to 4 (p-values between 0.000 and 0.039), and A1 shows similar results (p-values ~ 0.05), while difference for B1 could not be proven. When combining data for A and B locations, locations 2 and 3 had significantly more correct detections than locations 1 ($p = 0.04$ and 5 ($p <0.000$), while results for location 4 seem closer to results for location 1 ($p = 0.13$) but they still differed from results obtained at location 5 ($p <0.000$).

These results indicate that PoultryBot had more difficulty detecting eggs correctly if they were towards the side of the robot, such as on lateral locations 1 and 5. The lower detection rate seen for these locations might be explained by the egg being present more towards the boundaries of the camera view. As result of optical properties of the imaging setup, the images contained a radial intensity fall-off, such that eggs further from the image centre appeared darker and therefore were not always detected correctly. For more details on this matter, see Vroegindeweij et al. (2018). Using vignette correction morphological processing reduced the effects of the setup on the image, but they could not be removed completely. Also, if eggs were located further away from PoultryBot, there was a greater chance that a hen blocks the view towards the egg. The effect of lateral location on detection performance is relevant for future application, as this affects the scanning range of PoultryBot. From the current results, it can be concluded that at least 75% of the eggs within 0.5 m from the PoultryBot can be correctly detected. However, as locations in a poultry house will be visited multiple times a day and this allows for detection at a later moment, the effect of occasionally not detecting an egg is reduced. In these cases, however, eggs will remain longer in the poultry house, which might induce further undesired effects such as additional floor laying and egg eating.

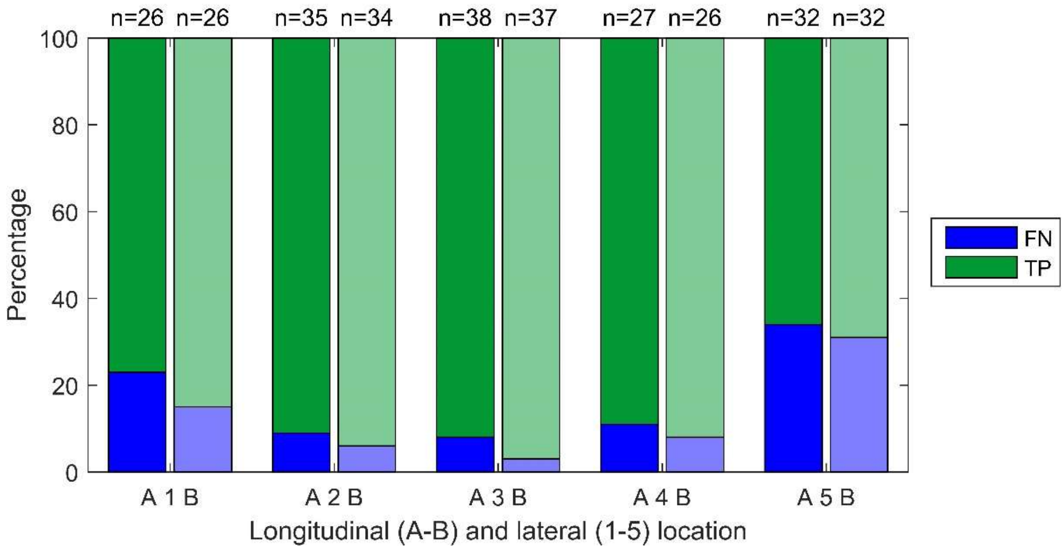

Fig. 9: Detection results as percentage of the number of the eggs present, expressed per location. Letters A and B indicate the first and second longitudinal location, while numbers 1 to 5 indicate the lateral location, which was perpendicular to robot path. Total number of eggs considered is given above each bar. False positives are not indicated, as they could not be related to a specific position.

False positives could not be related to specific locations and are thus not shown in Fig. 9. The calculated that the False Discovery Rate (FDR) per lateral location (1 to 5) ranged from 21% to 43%. As the false discovery rate varied between 0% and 57% for individual runs, numbers serve as indication only with no clear trends visible. False positive occurrence seemed to be dependent on ambient light levels and camera settings (especially the camera gain and the intensity threshold), as there appeared to be a correlation between brighter light conditions and animals close to robot being mistaken for eggs (see Vroegindeweij et al. (2018) for more details). As fixed intensity thresholds were used for egg detection, and ambient light conditions varied between runs due to variation in outdoor weather, this might explain a large part of the false positives. Furthermore, spots from sunlight, white paint on the wall or feathers on the floor were frequently mistaken as egg by PoultryBot. In commercial poultry houses, these effects are expected to be smaller, since under poultry house conditions both the amount of ambient light and light intensity are much lower.

5.4 Collection performance

In general, if eggs were detected, the collection operation started in the neighbourhood of the egg, and more than half of the eggs (54%) were immediately collected successfully. Some form of collection failure (i.e. starting the collection operation correctly, but failing to collect the egg) occurred in most other cases (at 43% of the eggs), and seems almost complementary to successful collection. Other collection results (clearly wrong location of collection or collection not starting at all) occurred only a few times for correct detections. This is as expected, since eggs do not disappear or move away easily. Cases that did occur, might have related to either a detection error or PoultryBot acting incorrectly during the collection operation. For example, in several cases the robot passed both waypoints around the egg faster than the control system could respond, indicating that the control system is responsible for these errors.

For each correct detection (TP), also collection performance was assessed. Results are given in Fig. 10, as percentage of the number of correctly detected eggs. For false positive detections, a collection operation was made resulting in 'no egg present'. As no location was known for these cases, they were excluded from the results shown. As shown in Fig. 10, between 40 to 70% of the correctly detected eggs were collected at once, but clear variation in collection performance can be observed. For statistical comparison of lateral locations 1 through 5, data from longitudinal locations A and B was combined. Eggs in front of the robot (lateral location 3) appeared to be collected correctly more often than eggs found near robot edges (locations 2 and 4, $p = 0.03$). The number of correct collections at location 1 was also lower than at location 3 ($p = 0.08$), but not significantly different from locations 2 and 4 (p

>0.7). Location 5, at the other side of PoultryBot, showed a further decrease in performance, especially compared to location 3 ($p < 0.001$).

These results might be explained by the behaviour of the robot, as eggs located further away from the initial path required more steering over a short distance to correctly approach the egg. If this was not accomplished in time, the risk of collection failure increased, mainly from an incorrect approach to the egg. The most likely explanation for the results at lateral location 1 being different from those at location 5 is that at location 1 the robot collided with the wall and ending up in front of the egg, such that chances for correct collection increased. This was already observed during the experiments, and can be confirmed when looking at the causes for failure, as shown in Fig. 11. Here, location 1 had a high amount of human intervention as the result of wall collisions, which were not seen at the other locations.

Also between longitudinal locations A and B variation was observed. On locations aside from the robot contour (lateral locations 1 and 5), collection results for the longitudinal locations B (more free space) were clearly worse compared to the A-locations (close to a pole). On the other hand, for lateral locations 2, 3, and 4 (more in front of the robot) the results for longitudinal locations B were slightly better compared with A-locations. This was most explicit for location 4 with $p = 0.08$. Although no clear explanations could be identified for this effect, it might be that the driving behaviour of PoultryBot for collecting the egg at the A-location influenced the collection performance on the B-location. Also, in a number of cases at location A5, the collection operation was ended manually due to collisions with the pole of the interior construction. Based on these results, the location of the egg with respect to obstacles such as a construction pole seems to have limited effect on the collection results. However, waypoint placement and driving behaviour does need improvement to make sure no collisions with construction elements occur during or after egg collection.

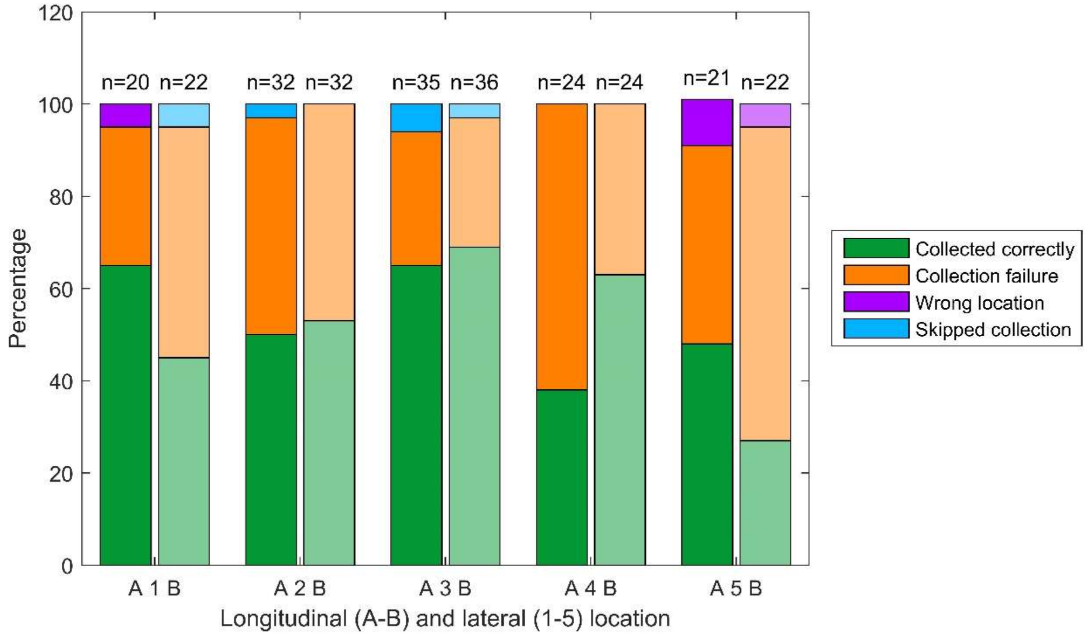

Fig. 10: Results of the collection operations as percentage of correctly detected (TP) eggs, expressed per location. Letters A and B indicate the first and second longitudinal location, while numbers 1 to 5 indicate the lateral location, which was perpendicular to robot path. Total number of eggs considered is given above each bar.

5.5 Collection failures

As 43% of the collection operations showed some collection failure (with some locations reaching 60%), also failure causes were investigated based on observations of the collection operation. Results as percentage of the total number of failures for each location are shown in Fig. 11. Statistical inference did not show any difference in failure between locations (p-values >0.25), except for the expected difference in obstacle collisions at lateral location 1.

The most frequent cause of failure for lateral locations 2 to 5 (56% of the cases) was the collection device being placed just next to the egg at the start of collection. This occurred less often at location 1, as wall collisions ensured that the robot was directed towards the egg. Frequently, these collisions also resulted in human intervention during the collection operation. These causes are likely due to a combination of reasons. Firstly, in the placement of waypoints for egg collection PoultryBot's steering behaviour or obstacle presence were not accounted for. Next, although the navigation method did account for the collection device's position on the robot, steering effects from the heuristic close to the waypoint might still have led to a wrong orientation of the collection device when reaching the egg. As a result, PoultryBot's front wheel might be oriented correctly towards the egg, but the collection device may still have been just next to the egg. Finally, steering corrections applied during collection operation may not always have resulted in the desired move of the collection device, as it had some freedom of movement and frictional forces limited the required lateral shift. Thus, the collection performance of PoultryBot was clearly influenced by the path planning for floor egg collection and the navigation algorithm. Improving the navigation method by better waypoint placement, changing the behaviour of the navigation heuristic and including not only the next waypoint, but also the one after that in the navigation control, are therefore all considered to be desirable. This is likely to reduce these problems and thus improve overall collection performance.

Other failures that occurred frequently, were the egg leaving the collection device after collection (as result of collection device shape), or the collection device being lowered after passing the egg. Lifting the device before actually reaching the egg, moving over an egg without collecting it or breaking the egg during collection also occurred, but they were not seen frequently. As loosing eggs mainly occurred towards the rear of the collection device, improving the design of the collection device is expected to resolve this cause of failure. Reducing failures such as moving over or breaking eggs during collection was more difficult, but their lower occurrence made them less important. The cases of lowering the collection device after passing the egg, lifting it before actually reaching the egg, or missing the waypoint after the egg and not stopping the collection operation at all, might all have to do with the processing speed of the control method. Improving and speeding up this method may allow PoultryBot to respond faster to new observations and changes in position, and make navigation and collection control more accurate.

Finally, in some cases the collection operation was not ended automatically and human intervention was required. Egg collection usually occurred properly in most of these cases, which explains why these cases exceeded 100% of collection failures. This particularly happened at location A5 due to collisions with a construction pole, and at lateral location 1 due to collisions with the wall. For the cases that occurred on locations 2 and 3, no direct explanation other than a control error could be identified. To avoid these situations, the navigation method needs improvements in handling obstacles. Furthermore, adding a strict time or distance limit on the collection operation appears useful to assure the collection operation stops in time.

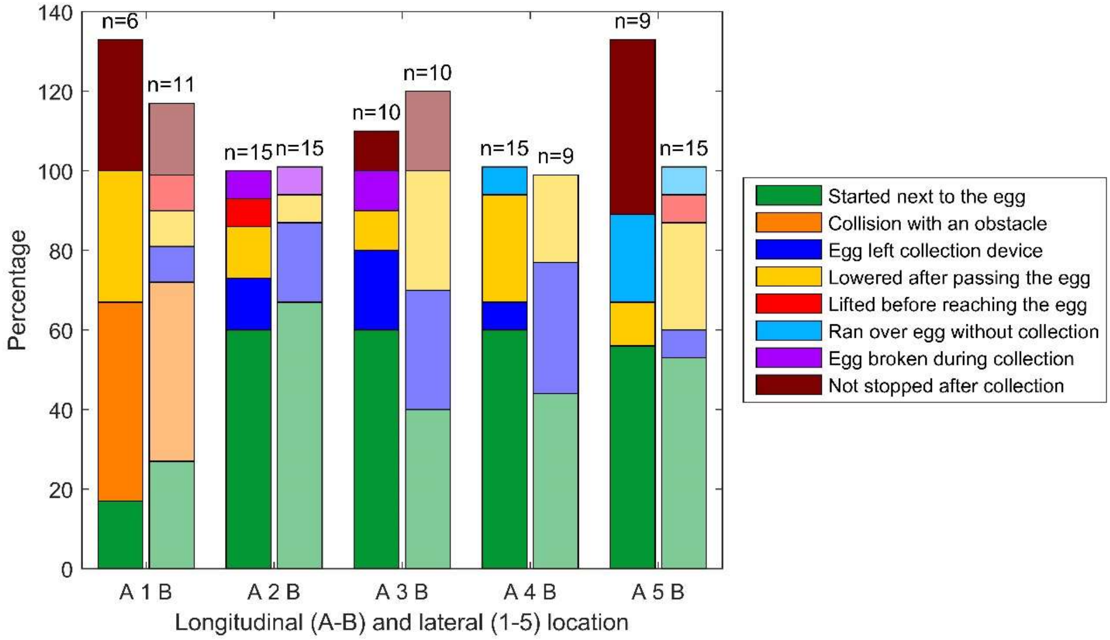

Fig. 11: Causes for collection failure as percentage of the number of failures, expressed per location. Letters A and B indicate the first and second longitudinal location, while numbers 1 to 5 indicate the lateral location, which was perpendicular to robot path. Total number of eggs considered is given above each bar. Results for not stopping the collection operation in time could exceed 100% score, as an egg could be successfully collected but the collection operation was not ended automatically.

6 Combining navigation and egg collection performance

In the previous sections, PoultryBot's capabilities for navigation and egg collection were evaluated in separate experiments. This section provides an integrated reflection on PoultryBot's performance by summarising the main findings and comparing them with the performance requirements stated in the introduction. Furthermore, it indicates limitations of the current system for application in commercial poultry houses, as well as suitable directions for further development.

6.1 Navigation performance overview

The first experiment in section 4 evaluated the long-term navigation capabilities of PoultryBot over 2400 m of autonomous driving. Here, PoultryBot proved capable of handling the various path types tested, ranging from border surveying via area sweeping to traversing large areas. Furthermore, it showed its ability to pass through narrow spaces by closely approaching obstacles, but also to deviate from the specified path if this was required to avoid obstacles. As the results of experiment 1 indicated that the occurrence of collisions might relate to the presence of obstacles on PoultryBot's path and the settings of the heuristic used, this was investigated in more detail in experiment 2. Obstacle position with respect to PoultryBot's desired path indeed influenced driving behaviour, with more steering and an increased prevalence of collisions. Also, an indication was found that changing the settings of the navigation heuristic could improve navigation results. During navigation, PoultryBot was able to localise itself with mean accuracy of 0.13 m. For 63% of the time, deviations also remained below 0.1 m. This approaches the desired accuracy of less than 0.1 m for 95% of the time. During the experiments, no clear effects of localisation failures on the behaviour of PoultryBot were observed. Some improvements are still desired in obstacle mapping and handling reference measurements. Although these results sound promising in terms of the requirements stated in the introduction, full autonomous operation of PoultryBot for navigation tasks in commercial poultry houses is not yet possible. The main reasons for this are the number of obstacle collisions observed and PoultryBot's inability to resolve these collisions autonomously.

6.2 Obstacle detection and awareness

For achieving the desired operational autonomy of PoultryBot, the first step is to improve its obstacle awareness. In the experiments, several cases were observed were PoultryBot collided with an obstacle just after it had left PoultryBot's field-of-view and was therefore no longer considered in the navigation. Adding or replacing sensors, such that the total field-of-view increases, might solve this problem as obstacles will then remain in sight even after PoultryBot passed them. Also for back-up manoeuvres, additional sensing on the rear side of PoultryBot might be required. Alternatively, a short-term obstacle history can be kept for navigation purposes, which still considers detected obstacles for a certain time after they have left the view of PoultryBot. This approach also allows for a different treatment of so-called hard and soft obstacles (Bac, 2015), as for example their retention time or relative importance can now be varied. In that case, PoultryBot should avoid hard obstacles such as construction elements, while soft obstacles such as hens still allow for a certain amount of interaction. Such an approach seems useful, as it was observed that the presence of hens in the environment had a minor influence on the navigation behaviour of PoultryBot. Thus, improving awareness of obstacles in the vicinity of PoultryBot, as well as taking obstacle properties into account, is desirable for further development of PoultryBot. Furthermore, a collision detection method is not incorporated and this still has to be added to PoultryBot.

6.3 Navigation components

Next to improving obstacle awareness, also some navigation components have to be added or improved before PoultryBot can function fully autonomous. For example, PoultryBot's inability to autonomously stop or reverse direction directly influenced current performance, as such manoeuvres are required for autonomous collision resolving. Although the method of Schlegel (1998) does allow for such actions, this was not yet implemented properly in the navigation system of PoultryBot, and needs therefore to be added. In case collisions occur, not only additional navigation behaviours such as reversing direction of motion are required, but also more high-level reasoning that considers adding or moving waypoints to resolve such situations. For example, when during egg collection PoultryBot reaches a dead end or has to collect an egg in a corner, this requires several additional waypoints for a back-up manoeuvre and the indication of a suitable follow-up path. In path planning for car-like robots, methods for defining such manoeuvres already exist (Csorvási, Á, & Kiss, 2015; Kiss & Tevesz, 2014), which might also be suitable for PoultryBot. If these missing components are implemented as well, PoultryBot is likely able to autonomously handle (potential) collisions, as well as entering corners and dead ends for egg collection.

If these navigation components are added, also the navigation heuristic requires an update, as already suggested from the results of experiment 2. This can be an improved static tuning of weight factors, but it might also be that different conditions require different settings. For example, conditions with an increased risk on collisions might need more obstacle-avoidance behaviour, whereas for traversing an open area more goal-oriented behaviour is desired. Also, handling collisions or reversing might require different settings of the weight factors. Thus, a system where weight factors in the heuristic are made dependent on the desired behaviour under specific driving conditions, might be a suitable improvement for proper functioning of PoultryBot.

6.4 Egg collection performance

Next to PoultryBot's navigation capabilities, also its performance in egg detection and collection was determined on over 300 eggs, and showed a dependency on the egg's location with respect to the robot. In front of PoultryBot, about 90% of the eggs were detected, while more towards the side this decreased to about 65%. On average, some 75% of the eggs within 0.5 m of PoultryBot were detected. Regarding false positive detections, a range between 0 and 57% was observed, with results being dependent on the combination of ambient light conditions and fixed settings for camera gain and detection threshold. As the images also contained some radial intensity fall-off, improving the optical setup is likely to increase detection performance, as also indicated in Vroegindeweij et al. (2018). Having more constant ambient light, which is expected to be the case in commercial poultry houses, will also benefit detection performance. With that, the performance comes close to the desired level of detecting 95% of the eggs present, although reaching the maximum 5% false positive detections might still be challenging. If the current detection method does not provide sufficient room for performance improvement, it might also be worthwhile to consider more advanced methods like Conditional Random Fields (He, Zemel, & Carreira-Perpiñán, 2004) for detecting eggs and other objects in images.

In terms of egg collection performance, about 40 to 70% of the eggs could be collected at once. If collection failed, this was mainly due to incorrect positioning of the collection device. The improvements for the navigation method proposed above can already solve part of this, but for egg collection some more improvements of the collection operation are recommended. The first is the path planning for the collection operation. When placing the waypoints for egg collection, more attention should be given to the robot's current pose, how to approach the egg and the presence of nearby obstacles. Instead of taking the shortest straight line from the current pose towards the egg, a smoother path is desired that can also be followed accurately by PoultryBot, while at the same time avoiding obstacle collisions. Next, the vehicle navigation strategy should be improved further, so that the orientation of the collection device at the start of collection is included. Finally, the speed of the control loops should be higher, such that the steering actions applied are also executed in time. Next to collection control, also design of collection device needs attention, as part of the eggs escaped after collection. Placing a barrier can easily solve this problem, while adapting the settings of the collection device might also reduce the occurrence of breaking or moving over eggs. With these improvements in collection control and the collection device, it is likely that almost all eggs will be collected properly, and the requirement on collection performance can be reached as well.

However, indicating a dependency between navigation behaviour and collection performance remains complex, for various reasons. Firstly, the robot path contained 2 waypoints between the eggs, to ensure the first and second eggs were approached from a similar direction, but driving behaviour was also subject to animal presence. Thus, paths were never exactly the same and effects of driving behaviour on collection performance varied between experiments. Secondly, the detection method proved sensitive to the variations in ambient light, leading to more false positives in these cases. As these also lead to collection operations, this affected driving behaviour and egg collection as well. Finally, floor conditions and egg properties do influence collection results but were subject to changes from natural variation, even between subsequent runs. Despite these difficulties, the presented results still provide a good indication of the future possibilities for applying autonomous robots for the collection of floor eggs in commercial poultry houses.

6.5 Wrap-up

By improving PoultryBot's obstacle handling and navigation behaviour as indicated above, it should be possible to cover all accessible areas of a poultry house. Furthermore, PoultryBot already has large flexibility in its search path, which can contain both long-distance movements, local search actions and other movements, in any combination. These features make PoultryBot capable of handling a wide range of physical environments, path characteristic and navigation behaviours. Furthermore, the obtained localisation accuracy is sufficient to map climate conditions or to register the location of the eggs found, thereby allowing to use this information to inform the farmer on house conditions or for planning PoultryBot's next day's collection path.

By also improving the egg collection operation, PoultryBot will be able to collect almost each egg that is detected and physically reachable. In that case, the exact position of the egg within the poultry house and the position of the egg with respect to PoultryBot's pose will be of limited influence on performance. Given the results from the presented experiments, in the current configuration already more than 40% of the eggs was collected successfully at once, with improper control being the major cause for failure. Using an improved control method likely leads to more than 80% of the eggs properly collected at first encounter, thus approaching the performance requirement on egg collection as stated in the introduction.

All these capabilities make the presented concept a suitable candidate for automating tasks in poultry houses, such as monitoring the animal environment or collecting floor eggs. However, problems or tasks with similar characteristics and requirements can also be found in many other applications, such as cleaning buildings, weed removal or security patrolling. Also, the flexibility and robustness present in PoultryBot for functioning in dense environments can be a great advantage when creating autonomous applications.

7 Conclusion

An autonomous mobile robot platform for use in a modern aviary poultry house has been introduced. PoultryBot was tested under real-life conditions, and has proved capable of moving autonomously through this environment. For this, various path types were used, while PoultryBot handled both fixed and moving obstacles during more than 3000 m of autonomous driving. Egg collection was tested on more than 300 eggs, of which about 46% was successfully collected, while for about 37% of the eggs present some collection failure occurred and only 16% of the eggs was completely missed. The most observed failures were caused by the collection device being placed just next to the egg, which can be solved by improving the control algorithms used for navigation and egg collection. These results show the validity of the PoultryBot concept and indicate that application of smart autonomous vehicles in dense animal environments is possible. Improvements in obstacle handling and navigation and the collection and reliability of components are required before commercial application of this idea becomes feasible.

8 Acknowledgements

Fonds Pluimveebelangen is acknowledged for their financial support in the development of PoultryBot. Furthermore, several other people also contributed to this work, so the authors would like to thank: The Wageningen University students participating in the Field Robot Event 2015, and Ragnar Groot Koerkamp, Gerco Schopman and Maaike Vollering for their assistance in preparation of PoultryBot and the test environment; Wicher Aantjes, Yeb Andela and Dini Ras for their assistance in executing the experiments; Andries Siepel and other Unifarm staff for taking care of the animals in the test environment.

References

Aelfers, R., van Esbroeck, E., van Hell, S., Raedts, R., & Russchen, B. (2015). Report Field Robot Event 2015 - Team Steketee SmartTrike. Wageningen: Wageningen University.

Aertsen, T., Bauweleers, K., Bessemans, N., de Geest, P., Dinc, M., Donders, L., Fivez, R., Geyssents, R., Nuytsm, K., Oorts, L., Smets, E., van Aggelen, A.M.,Vandevoorde, K. (2012). *The Farmer Assistant*. Leuven, Belgium.

Bac, C. W. (2015). *Improving obstacle awareness for robotic harvesting of sweet-pepper.* (PhD thesis Wageningen University), Wageningen University, Wageningen. Retrieved from http://edepot.wur.nl/327202

Bac, C. W., Hemming, J., & van Henten, E. J. (2013). Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and Electronics in Agriculture, 96*, 148-162. doi: https://doi.org/10.1016/j.compag.2013.05.004

Bac, C. W., Hemming, J., van Tuijl, B. A. J., Barth, R., Wais, E., & van Henten, E. J. (2017). Performance Evaluation of a Harvesting Robot for Sweet Pepper. *Journal of Field Robotics 34(6)*, 1123-1139 doi: 10.1002/rob.21709

Bac, C. W., van Henten, E. J., Hemming, J., & 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. doi: Doi 10.1002/Rob.21525

Bakker, T. (2009). *An autonomous robot for weed control : design, navigation and control.* (Proefschrift Wageningen Universiteit), Wageningen Universiteit, Wageningen. Retrieved from http://edepot.wur.nl/1099

Bayar, G., Bergerman, M., Koku, A. B., & Konukseven, E. i. (2015). Localization and control of an autonomous orchard vehicle. *Computers and Electronics in Agriculture, 115*, 118-128. doi: http://dx.doi.org/10.1016/j.compag.2015.05.015

Blokhuis, H. J., & Metz, J. H. M. (1995). *Aviary housing for laying hens*. Wageningen.

Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S. (1999). Experiences with an interactive museum tour-guide robot. *Artificial Intelligence, 114*(1–2), 3-55. doi: http://dx.doi.org/10.1016/S0004-3702(99)00070-3

Claeys, D. (2007). *Socio-economische gevolgen van verschillende huisvestingssystemen in de leghennenhouderij*. Mededeling ILVO nr. 20 Merelbeke-Lemberge: Instituut voor Landbouw- en Visserijonderzoek, Eenheid Landbouw & Maatschappij.

Csorvási, G., Á, N., & Kiss, D. (2015, 27-30 May 2015). *RTR+C*CS: An effective geometric planner for car-like robots.* Paper presented at the Proceedings of the 2015 16th International Carpathian Control Conference (ICCC).

Deepfield Robotics. (2016). BoniRob Retrieved 1-4-2016, 2016, from http://www.deepfield-robotics.com/en/BoniRob.html

Dubrofsky, E. (2009). *Homography estimation. Optical Engineering*, *15* (March), 977. https://doi.org/10.1117/1.3364071

Froehlich, E. K. F., & Oester, H. (2001). *From battery cages to aviaries: 20 years of Swiss experience*. Paper presented at the 6th European Poultry Conference, Zollikofen, Switzerland.

He, X., Zemel, R. S., & Carreira-Perpiñán, M. Á. (2004). *Multiscale conditional random fields for image labeling.* Paper presented at the IEEE computer society conference on Computer vision and pattern recognition, 2004. .

Hiremath, S., Evert, F. K. van, Heijden, G. W. A. M. van der, ter Braak, C. J. F. , & Stein, A. (2012). *Image-Based Particle Filtering For Robot Navigation In A Maize Field.* Paper presented at the Workshop on Agricultural Robotics: Enabling Safe, Efficient, Affordable Robotics for Food Production, Vilamoura, Portugal, 11-10-2012.

Kiss, D., & Tevesz, G. (2014, 28-30 May 2014). *A steering method for the kinematic car using C*CS paths.* Paper presented at the Proceedings of the 2014 15th International Carpathian Control Conference (ICCC).

Lely. (2015). Lely Discovery - mobile barn cleaner. Retrieved 28-11-2015, 2015, from http://www.lely.com/en/housing/mobile-barn-cleaner/discovery_0

Nof, S. Y. (2009). *Springer Handbook of Automation*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Qi, H., Brookshaw, I. J., Low, T., & Banhazi, T. M. (2013). *Development of an autonomouos welfare robot to be used in poultry buildings*. Paper presented at the 2013 Society for Engineering in Agriculture Conference, Mandurah, Australia.

Qi, H., Zhou, H., Low, T., Mehdizadeh, S., Tscharke, M., & Banhazi, T. (2013). *A hybrid WSN system for environment monitoring at poultry buildings.* Paper presented at the Proceedings of the 2013 Conference of the Australian Society for Engineering in Agriculture (SEAg 2013).

Schlegel, C. (1998, 13-17 Oct 1998). *Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot.* Paper presented at the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998. Proceedings.,.

Shalal, N., Low, T., McCarthy, C., & Hancock, N. (2015a). Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion – Part A: Tree detection. *Computers and Electronics in Agriculture, 119*, 254-266. doi: http://dx.doi.org/10.1016/j.compag.2015.09.025

Shalal, N., Low, T., McCarthy, C., & Hancock, N. (2015b). Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion – Part B: Mapping and localisation. *Computers and Electronics in Agriculture, 119*, 267-278. doi: http://dx.doi.org/10.1016/j.compag.2015.09.026

Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. *Intelligent robotics and autonomous agents*. Cambridge, MA [etc.]: MIT.

SmartTrike. (2015). SmartTrike. *Field Robot Event 2015 - Competitors* Retrieved 26-7-2016, 2016, from http://fre2015.um.si/index.php/2-fre2015/17-smarttrike

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Haehnel, D., Rosenberg, C., Roy, N. (2000). Probabilistic algorithms and the interactive museum tour-guide robot minerva. *The International Journal of Robotics Research, 19*(11), 972-999.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. Cambridge, Massachusets: The MIT Press.

Triebel, R., Arras, K., Alami, R., Beyer, L., Breuers, S., Chatila, R., Chetouani, M., Cremers, D., Evers, V., Fiore, M., Hung, H., Islas Ramírez, O.A., Joosse, M., Khambhaita, H., Kucner, T., Leibe, B., Lilienthal, A.J., Linder, T., Lohse, M., Magnusson, M., Okal, B., Palmieri, L., Rafi, U. van Rooij, M. Zhang, L. (2015). *SPENCER: a socially aware service robot for passenger guidance and help in busy airports*. Paper presented at the 10th Conference on Field and Service Robotics, FSR 2015, Toronto, Canada. http://doc.utwente.nl/98273/

van Henten, E. J., Hemming, J., van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., & van Os, E. A. (2002). An Autonomous Robot for Harvesting Cucumbers in Greenhouses. *Autonomous Robots, 13*(3), 241-258. doi: 10.1023/a:1020568125418

van Niekerk, T. G. C. M., & Reuvekamp, B. F. J. (1997). *Alternatieve huisvesting leghennen : verslag derde ronde + eindverslag = Alternative housing systems for laying hens : report third trial and final report*. Beekbergen: Praktijkonderzoek Pluimveehouderij "Het Spelderholt".

Vroegindeweij, B. A., Ijsselmuiden, J., & van Henten, E. J. (2016). Probabilistic localisation in repetitive environments: Estimating a robot's position in an aviary poultry house. *Computers and Electronics in Agriculture, 124*, 303-317. doi: http://dx.doi.org/10.1016/j.compag.2016.04.019

Vroegindeweij, B. A., Kortlever, J. W., Wais, E., & Henten, E. v. (2014, 6-10 July 2014). *Development and test of an egg collecting device for floor eggs in loose housing systems for laying hens.* Paper presented at the International Conference of Agricultural Engineering AgEng 2014, Zurich.

Vroegindeweij, B. A., van Hell, S., IJsselmuiden, J., & van Henten, E. J. (2018). Object discrimination in poultry housings using spectral reflectivity. *Biosystems Engineering, 167, 99-113*.

Vroegindeweij, B. A., van Willigenburg, L. G., Groot Koerkamp, P. W. G., & van Henten, E. J. (2014). Path planning for the autonomous collection of eggs on floors. *Biosystems Engineering, 121*(0), 186-199. doi: http://dx.doi.org/10.1016/j.biosystemseng.2014.03.005

Wageningen University. (2009). *Proceedings of the 7th Field Robot Event 2009 : Wageningen, July 6 & 7, 2009*. Wageningen: Wageningen University, Farm Technology Group.

Wang, G.-H., Hu, Z.-Y., & Wu, F.-C. (2004). Single view based measurement on space planes. *Journal of Computer Science and Technology, 19*(3), 374-382.

Zheng, Y., Yu, J., Kang, S. B., Lin, S., & Kambhamettu, C. (2008, 23-28 June 2008). *Single-image vignetting correction using radial gradient symmetry.* Paper presented at the Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.