Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

A methodology of orchard architecture design for an optimal harvesting robot

Victor Bloch [a,c], Amir Degani [a,b,*], Avital Bechar [c]

[a] Faculty of Civil and Environment Engineering, Technion, 32000 Haifa, Israel
[b] Technion Autonomous Systems Program, 32000 Haifa, Israel
[c] Institute of Agricultural Engineering, Agricultural Research Organization, Volcani Center, 50250 Bet-Dagan, Israel

ARTICLE INFO

Article history:
Received 20 January 2017
Received in revised form
6 November 2017
Accepted 17 November 2017

Keywords:
Robotic harvesting
Task based optimisation
Optimal tree architecture
Robot working environment
Optimal environment design

To improve robot performance for agricultural tasks, and decrease its cost, the robot can be optimally designed for a specific task in a specific working environment. However, since the environment defines the robot optimal kinematics, the environment itself should also be optimised for optimal robot performance. The objective of this paper is to present and demonstrate a methodology for simultaneous optimal design of robot kinematic and the working environment. This methodology was demonstrated by an example on a tree orchard design for an apple harvesting robot. First, an optimal robot structure for apple picking task was found for a number of tree architectures (shaped by different training systems): Central Leader, Y-trellis and Tall Spindle. Results indicate that for minimising the average apple picking time, the Tall Spindle architecture is preferable for the robotic harvesting of both a single tree and a tree row. Further, the influence of the robot platform motion time on the chosen robot kinematics and the tree training system was analysed. Results show that for fast platforms, the Tall spindle architecture is advantageous. If the platform movement between positions near the trees is slow, the Central Leader architecture is favourable. Additionally, the tilt angle of the Y-trellis training system was analysed using simulated models created by the L-systems simulations. The optimal tilt angle was found to be nearly horizontal (85°), allowing the robot designer to choose the optimal combination of the robot kinematics, number of robot harvesting positions around the tree and the tree training system.

1. Introduction

Despite decades of research on robotic applications in agriculture, commercial fruit harvesters are sparse (Bac, Van Henten, Hemming, & Edan, 2014; Bechar, 2010). Among others, the three main reasons are: 1) high cost of existing (industrial) robots and their maintenance, 2) insufficient speed of self-designed robots, making them unprofitable for farmers, and, 3) agricultural environment complexity, causing the motion, sensing and trajectory planning of the robot to be complicated, time consuming and therefore impractical (Bac et al., 2014).

* *Corresponding author*. Faculty of Civil and Environment Engineering, Technion, 32000 Haifa, Israel.
E-mail addresses: victorc@technion.ac.il (V. Bloch), adegani@technion.ac.il (A. Degani), avital@volcani.agri.gov.il (A. Bechar).
https://doi.org/10.1016/j.biosystemseng.2017.11.006
1537-5110/

Nomenclature

Symbols

α, θ	a and d parameters of the Denavit–Hartenberg notation
D_x	distance of the robot base from the plant stem, m
E_{fr}	energy for picking a single fruit, Joule
E_i	energy consumed by the ith actuator, Joule
E_s	static work against the load and the robot's weight, Joule
$\gamma_{trellis}$	trellis tilt angle in the Y-trellis training system
F	robot optimisation cost function, s
I	moment of inertia of the robot links and load, kg m^2
J	Jacobian
N_{DOF}	number of degrees of freedom of manipulator
N_{fruit}	number of fruit on a tree
N_{pos}	number of robot positions around a tree
N_{picked}	number of picked fruit
N_{tree}	number of trees in an orchard
q_{ini}	initial home robot configuration
q_{fin}	final robot configuration
τ	torques produced by robot actuators, Nm
$t_{fr,i}$	time spent picking a specific fruit, s
T_{mov}	movement time between robot base positions, s
T_{fruit}	average time for fruit picking, s
T_{tot}	total time for picking all the fruit in an orchard, s
W_i	power of the ith actuator, Watt
X	Cartesian coordinate system axis directed perpendicular to the row, m
Y	Cartesian coordinate system axis directed along the row, m
Z	Cartesian coordinate system axis directed along the plant height, m

Abbreviations

CL	tree shaped by Central Leader training system
DOF	degrees of freedom of manipulator
R	revolute joint
RRR	Revolute-Revolute-Revolute robot structure
RRRR	Revolute-Revolute-Revolute-Revolute robot structure
RRP	Revolute-Revolute-Prismatic robot structure
RRT	exploring random tree algorithm
P	prismatic (linear) joint
PPP	Prismatic-Prismatic-Prismatic robot structure
TS	tree shaped by Tall Spindle training system
YT	tree shaped by Y-trellis training system

Typically, to overcome the above three mentioned reasons in general, and particularly in fruit harvesting, self-designed simple robots fitted to their task and environment are built in a number of studies. One of the effective ways to design such robot is a task-based optimisation, enabling to find the robot kinematics with the best performance fitted to a geometric structure of the orchard architecture, representing the working environment of the robot. However, since both the robot kinematics and environment structure influence the optimal result, both the robot and the environment must be optimised simultaneously to achieve the optimal robot performance. In this paper, we present and demonstrate a methodology for simultaneous optimisation procedure for kinematic design of a robot as well as modification of the working environment. The methodology consists of the three following parts, each demonstrated by an example.

1. Comparison of the robotic harvesting effectiveness of existing tree shaping methods (training systems). This is shown by comparing the performance of a single robotic arm picking apples from single trees with different training systems: central leader, tall spindle, and y-trellis.
2. Influence of the robot platform moving time on the robotic harvesting effectiveness for trees shaped by different training systems. This is shown by comparing the performance of a single robotic arm carried by a platform moving along the row with different speeds.
3. Influence of the geometric tree parameters on the robotic harvesting effectiveness. This is shown by comparing the performance of a single robotic arm picking apples from single simulated y-trellis trees with different tilt angles.

The goals of this paper are to demonstrate the methodology for simultaneous optimisation for a robot and its working environment, and to discuss the preliminary results achieved by this methodology.

For agricultural applications, robotic arms are often tailor designed. They strive to be "light, simple and cheap" such as the arm for kiwi harvester of Scarfe, Flemmer, Bakker, and Flemmer (2009), with low number of degrees of freedom (DOF) such as a robot for harvesting lettuce of Cho, Chang, Kim, and An (2002), or low-cost robot for greenhouse applications developed by Belforte, Deboli, Gay, Piccarolo, and Aimonino (2006). Moreover, the robots are optimised for a specific task, such as an optimal robot for cucumber harvesting (Van Henten, Slot, Hol, & Van Willigenburg, 2009), eggplant harvesting (Han, Xueyan, Tiezhong, Bin, & Liming, 2007), melon harvester (Edan & Miles, 1993) or apple harvesting (Bloch, Bechar, & Degani, 2017; Silwal et al., 2017). However, up to now, the optimisation was focused mainly on the robot, assuming that the environment was given and unchangeable. A number of commercial developers (FFRobotics Ltd., Israel; Abundant Robotics Ltd., USA) use modern tree architectures as more convenient environment, nevertheless, systematic environment analysis was not reported.

In the manufacturing domain, the robot environment is defined as a robot cell. Design of robotic cells for throughput optimisation is well studied and helps to solve numerous industrial challenges (Dawande, Geismar, Sethi, & Sriskandarajah, 2007). The main methods of the cell design are effective scheduling, use of multiple gripper, and parallel working robots. Current agricultural environments were designed to meet the agro-technical constraints and to fit manual labour, and do not take into account the suitability for robotic processing.

Simplifying and structuring of the agricultural environment was investigated by Hua and Kang (2013) (optimising

L-systems parameters for best light interception), by Tanigakia, Fujiuraa, Akaseb, and Imagawa (2008) (comparison of robotic processing on two tree training methods), by Vougioukas, Arikapudi, and Munic (2016) (testing the suitability of the orchard trees for robotic processing) and by Bac et al. (2016) (widening sweet pepper stem spacing to improve robot performance). Nevertheless, design or optimisation of the environment for robotic harvesting has not yet been performed. Such optimisation is difficult because of the large number of optimisation parameters and required knowledge of the plant behaviour. To circumvent these problems, the models of actual fruit trees trained by different training systems which represent different types of robotic environments, were used. Evaluation of the effectiveness of the existing environment types is performed in this paper as a preliminary step to the optimisation of the environment, which is new to the agricultural robot optimisation.

Modern high plant density training systems, such as the Tall Spindle and Y-trellis were developed mainly for increasing the yields and quality of fruit (Bergerman, Sanjiv, & Hamner, 2012; Robinson, DeMarree, & Hoying, 2007). In addition, they save labour time during harvesting by providing a convenient environment for human harvesters. This advantage can also be used to provide an environment suitable to robotic harvesters, turning them into a profitable harvesting solution.

Constructing the environment model is a critical step in the task-based optimisation process. Several studies analysed agricultural environments using accurate models. Edan, Flash, Peiper, Shmulevich, and Sarig (1991) measured the location of orange fruit to achieve an efficient trajectory for a specific robot. Lee and Rosa (2006) measured location of orange fruits to develop a fruit picking technique. There are three commonly used methods for plant reconstruction and modelling: 1) based on visual images (e.g., Santos & Ueda, 2013), 2) 3D scanning (e.g., Méndez, Rosell-Polo, Sanz, Escolà, & Catalána, 2014; Preuksakarn et al., 2010), and 3) a mechanical sampling of points in space (e.g., passive robot devise used by Edan et al. (1991)). To achieve accurate 3D models of existing trees without developing and using complicated algorithms for 3D reconstruction, a simple mechanical device, similar to Edan et al. (1991), was used to digitise and model existing trees.

One of the methodology parts is testing the influence of the tree shape parameters on the robotic harvesting. To do this, models of trees with different shapes were simulated based on existing tree models varied in a certain range. To use tree models with required parameters, the tree models were described and simulated by the L-Systems (Prusinkiewicz & Lindenmayer, 1990), a method for the description of the geometric features, and Markov chain (Costes, Lauri, & Regnard, 2006), a method for the description of the statistic behaviour of the trees. The L-Systems method uses the fractal nature of the plants. The geometry of a plant can be defined by simple self-repetitive rules including geometric parameters of the plant. The Markov chain method uses a table of geometric transformations of the plant parts and their probability. In a number of studies, tree simulating software were developed, such as L-studio (Karwowski & Prusinkiewicz, 2004), PlantToon (Bonora, Stefanelli, & Costa, 2013), MAppleT (Costes et al., 2008), and AMAPstudio (Griffon & Coligny, 2013). Though, in these studies, mainly agronomic and biological aspects of the tree models were considered.

2. Materials and methods

2.1. Environment modelling

Three apple trees shaped by different training systems – the Central Leader (CL) (Fig. 1(a)), Tall Spindle (TS) (Fig. 1(b)) and Y-trellis (YT) (Fig. 1(a)) – were modelled and considered as the robot environment. The tree model consists of the branches and the trellis parts (wires) represented by cylinders, the fruit represented by ellipsoids, and the ground represented by a plane. Leaves were not modelled in this preliminary examination because they do not represent mechanical obstacles for a robotic arm and in order to simplify the modelling process, analysis and optimisation. The origin of the environment coordinate system is located at the point of intersection of the ground plane with the tree trunk. The *Z*-axis is directed upward perpendicular to the ground plane, the *Y*-axis is parallel to the tree row, and the *X*-axis complements the right handed system (Fig. 1(a)). The 3-D tree models used in this paper are included in the Supplementary material. For a full library of trees see https://github.com/CearLab/Tree-Models-Library.

To measure and model the environment of the robot, three methods were used: modelling by a mechanical measuring device used in the case of trees with complicated structures; modelling by manual reconstruction from images used in case of trees with few branches and visible structures; and by simulating the tree models using L-systems, i.e., creating a geometric tree model using mathematical rules, used in case when the influence of tree geometric parameters on the robot performance is investigated.

The mechanical measuring device, a 'digitiser', shown in Fig. 2, is a passive robotic arm with a kinematic structure of an RRRR robot (four revolute joints) with three links at lengths of 1.1 m, 1.1 m, and 0.43 m. The angles of the structure joints are measured by encoders with a resolution of 0.07°, which results in a total maximal error of 3.5 mm. The location of the tip of the device is calculated according to the forward kinematics of the robot. Full measurement of a cultivated apple tree takes approximately three hours using this device. The complicated structure of the CL tree did not allow to be modelled from pictures, hence the mechanical digitiser was used for more accurate modelling.

Unlike the CL, the TS and YT are modern architectures which are highly engineered and hence can be easily modelled using several images. Moreover, in section 3.3 different YT tree models with different geometries were simulated using the L-systems to show how these different tree geometries affect the robot performance.

2.2. Tree model simulation

Simulated models of the trees trained by the YT system were built using the L-systems method (Prusinkiewicz, Hammel,

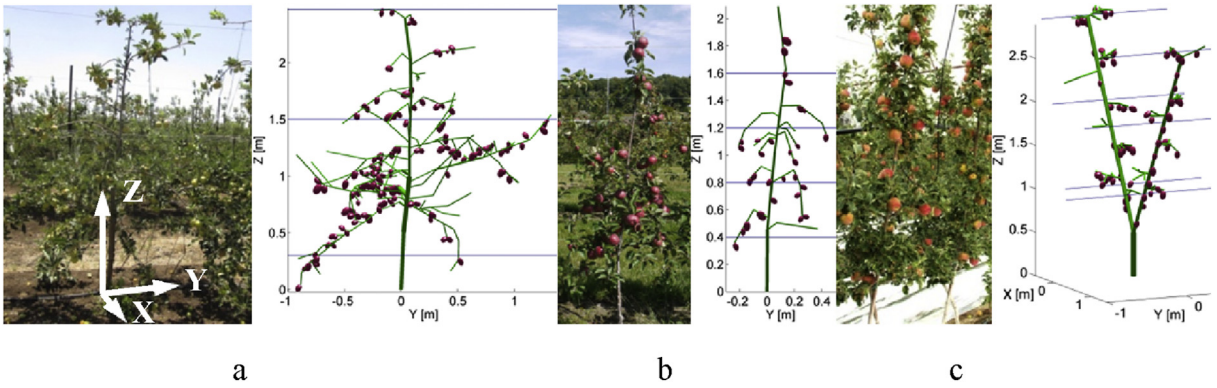


Fig. 1 – Three actual task environments and their models: apple trees trained to Central Leader (a), Tall Spindle (b) and Y-trellis (c) training systems.

Hanan, & Mech, 1990). The tree geometry was defined by the following L-systems rules (symbols defined in Prusinkiewicz & Lindenmayer, 1990). The parameters of these rules were obtained by analysing a number of images of the trees growing in orchards, which can be used for the 3D reconstruction. The L-systems rules and parameters are

$$\begin{aligned} &n = 3 \\ &\#define \quad \alpha_1 30^\circ / * \gamma_{trellis} * / \\ &\#define \quad \alpha_2 45^\circ / * \text{branching of the generations 2 and 3} * / \\ &\omega : A[\&(\alpha_1)FFFFF][\backslash(180^\circ)\&(\alpha_1)FFFFF] \\ &p_1 : F \rightarrow A[+(\alpha_2)F][-(\alpha_2)F] \end{aligned} \tag{1}$$

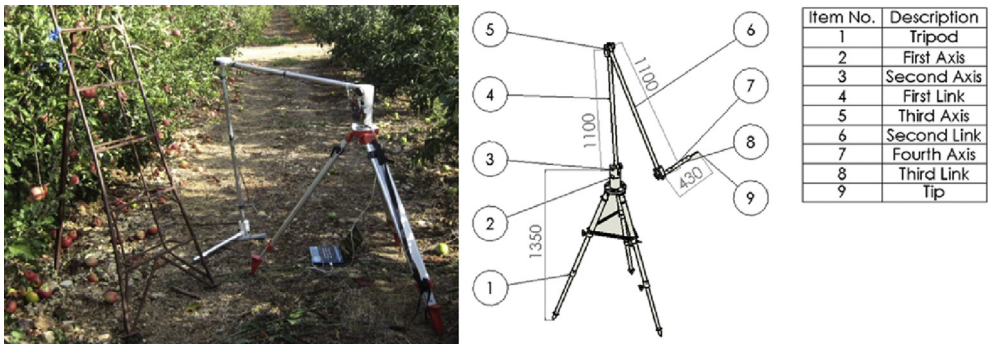
In the designed L-systems, the number of branch generations is three ($n = 3$). The first branching angle is defined as 30°, and the second as 45°. The axiom (ω) builds the following structure: build vertical branch A (the tree trunk), turn around the X axis by 30° $([\&(\alpha_1))]$, build five branch intervals (FFFFF), return to the previous position (][), turn around the Z axis by 180° (180°), turn around the X axis by 30° $(\&(\alpha_1))$, and build five branch intervals (FFFFF). The rule (p_1) replaces each branch interval F by the following structure: build branch interval A, turn around the up direction by 45° and build one branch interval $(+(\alpha_2)F)$, return to the previous position (][), turn around the up direction on −45° and build one branch interval $(-(\alpha_2)F)$.

The L-systems rules described in Eq. (1) were used to simulate a group of tree models with $\gamma_{trellis}$ with values in range of 5°–85° in steps of 5°, for the tree architecture design performed in Section 3.3. Three examples out of the total 17 simulated tree models, with trellis tilt angle, $\gamma_{trellis}$, of 5°, 30° and 85°, are shown in Fig. 3. The tree model with $\gamma_{trellis} = 5^\circ$ resembles the TS training system but with two main branches. The model with $\gamma_{trellis} = 30^\circ$ is close to the common YT trees. The model with $\gamma_{trellis} = 85^\circ$ resembles the trellises of kiwi orchard. The wires of the supporting structures were added to the tree models similar to the wires in actual orchards.

2.3. *Robot performance cost function*

The effectiveness of the robot was evaluated by the robot performance cost function which depends on the customer's needs. Thus, the cost function could not be strictly defined for the general case. In this research, the robot average time to pick a fruit was defined as the cost function. Since the mechanical aspect of the robot's performance is studied in this paper, only the robot manipulator motion time, which is defined here as the cost function F, is considered. This function depends on the robot's kinematics and the power of its actuators.

The average picking time depends on parameters such as actuator power and weight, construction material, etc., which are defined by the designer; an exact calculation of the average picking time is therefore nearly impossible for the general case. To find the optimal robot for our case study,



Item No.	Description
1	Tripod
2	First Axis
3	Second Axis
4	First Link
5	Third Axis
6	Second Link
7	Fourth Axis
8	Third Link
9	Tip

Fig. 2 – Picture (left) and sketch (right) of the digitiser, a passive, robotic arm with an RRRR structure used as a measuring device.

average picking time was evaluated by making several assumptions about the robot actuators and structure:

- The robot was a manipulator with revolute or prismatic joints with number of DOF denoted as N_{DOF}. In this paper, N_{DOF} is predefined to be $N_{DOF} = 3$. General Denavit–Hartenberg (DH) parameters table (Craig, 2005) is used.
- Actuators were considered massless, assuming that the actuators are mounted on the robot base and are transmitting forces through four bar mechanisms or cables, similar to the actuating system presented in Scarfe et al. (2009).
- The length density of the robot links were taken as 1.5 kg/m (similar to the length density of a 3-mm thick aluminium tube with a 60 mm diameter).
- The mass of the load were taken as 0.15 kg (assumed as the mass of an average apple). The mass of a fruit picking end-effector, as well as the mass of a camera, strongly depends on its design, e.g., in Bac et al. (2017), hence it was not taken into consideration in this study.
- The power of the robot actuators was given specific values: 10 W for the first robot joint actuating the weight of the entire robot. The power of the rest of the actuators decreases proportionally to the weight of the links moved by the actuator.
- The picked fruit were placed in a gathering bin attached to the robot platform in the robot home configuration. The considered robot motion consists of the following stages: moving the end-effector from the robot home configuration to a fruit, approaching the lower hemisphere of the fruit, and retracting back to the home configuration. The fruit detachment is performed by the end-effector.

To approximate the time and energy of motion we used basic physical expressions. Assuming that the robot geometry, link masses and moments of inertia, and actuator power are known, the time spent picking a specific fruit, $t_{fr,i}$, is

$$t_{fr,i} = \max(E_j/W_j), \tag{2}$$

where E_j is the energy consumed by the jth actuator, and W_j is the power of the jth actuator. According to the above assumption, the power of the first actuator is $W_1 = 10W$ and the power of the jth actuator is

$$W_j = W_1 \frac{\sum_{k=j}^{N_{DOF}} a_k + d_k}{\sum_{k=1}^{N_{DOF}} a_k + d_k}, \tag{3}$$

where a_k and d_k, are the robot link lengths according to the DH notation if the joint k is revolute, and the maximal link stroke if the joint k is prismatic.

The energy $E_{fr} = \sum_i E_i$ needed to pick a single fruit is the energy of the robot's movement from its initial home configuration, q_{ini}, to a final configuration, q_{fin}, set for picking the fruit. Therefore, the energy is calculated as

$$E_{fr} = E_d + E_s, \tag{4}$$

where E_d is the dynamic work against the inertia of the robot and the load, E_s is the static work against the load and the

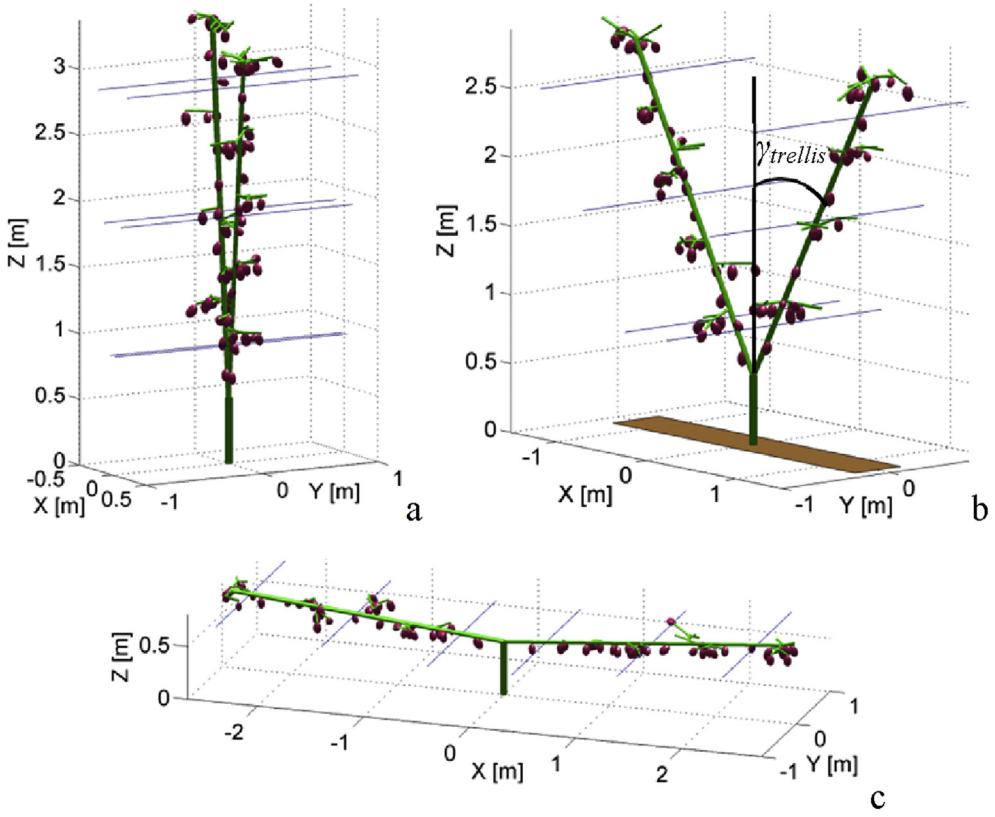


Fig. 3 – Three simulated trees shaped by the Y-trellis method with different trellis tilt angle: (a) 5°, (b) 30° and (c) 85°. The branches, trellis wires, and fruit are modelled by cylinders and ellipsoids.

robot's weight. The damping work is disregarded since it highly depends on the specific actuator design, such as the gearbox design and ratio. The static work is evaluated as

$$E_s = \int_{q_{ini}}^{q_{fin}} \tau(q)\mathrm{d}q. \quad (5)$$

The torques, τ, produced by the actuators are calculated with the help of the transposed Jacobian.

$$\tau = J^T P \quad (6)$$

where P is the force acting on the robot, consisting of the weight of the load and of the robot links, which are all directed in the negative Z-axis direction. The mass of each link is calculated using the link length and length density, with the weight applied at the middle of each link.

The dynamic work, E_d, is calculated by

$$E_d = \int_{q_{ini}}^{q_{fin}} I(q)\ddot{q}\mathrm{d}q, \quad (7)$$

where I is the moment of inertia of the robot links and load, which depends on the robot configuration, and $\ddot{q}$ is the acceleration in the robot joints. It is assumed that the acceleration has a maximal value, $\ddot{q}_{max}$, (bang-bang control), hence, it is taken as a constant. Finally, the dynamic work can be approximated as

$$E_d = \ddot{q}_{max} \int_{q_{ini}}^{q_{fin}} I(q)\mathrm{d}q. \quad (8)$$

The cost function F is therefore the average fruit-picking time, t, for all picked fruit N_{picked}.

$$F = \frac{\left(\sum_{i=1}^{N_{picked}} t_{fr,i}\right)}{N_{picked}}. \quad (9)$$

The cost function F is measured in time units (s). The time calculation does not include fruit-recognition time, trajectory-planning time, time for fruit detachment and placement in the gathering bin, etc. These are important, but has negligible influence on the kinematics of the robot, and hence can be ignored. Therefore, the time values in our results are different from the values reported in previous experimental studies, such as Van Henten et al. (2009) and Bac et al. (2017).

2.4. Robot structures

Type and order of the robot joints can strongly affect the applicability of the robot structure to different environments. For a 3-DOF robot, we checked the following structures: RRR, RRP and PPP, where R and P represent revolute and prismatic (linear) joints, respectively, as shown in Fig. 4. These structures were chosen because they represent typical industrial robotic arms: RRR represents an articulated robot, RRP represents a telescopic robot, and PPP represents a Cartesian robot.

2.5. Optimisation variables

The optimisation variables defining the robot kinematics are the known DH convention parameters α, θ, a and d (Craig, 2005). The total number of variables is $4 \times N_{DOF}$. The variables representing the robot's DOF (either θ or d) are found by solving the inverse kinematics for a specific robot configuration. Thus, the total number of free unconstrained optimisation variables is $3 \times N_{DOF}$. To decrease the number of the optimisation variables, the structures of the typical industrial robots were used to predefine the angles (α_i) to be either zero or $\pi/2$ and the link offset to be zero or $d_2 = 0.1$ m as shown in Fig. 4. Thus, the optimisation variables for each robot construction are: d_1, a_2, a_3 for RRR, d_1, a_2 for RRP, and θ_1, θ_2 for PPP. Moreover, the position of the robot base constitutes two additional optimisation variables. Each position was defined by two parameters: X and Y coordinates on the ground plane.

The limits of the optimisation variables were defined as, α in the interval $[-\pi, \pi]$, θ in the interval $[-\pi, \pi]$, a and d in the interval [0, 3], assuming that the height and width of the orchard trees does not exceed 3 m. Similarly, the X and Y coordinates of the robot base position are in the interval [−3, 3].

2.6. Optimisation constraint

To perform a task in a given environment, a robot has to be able to pick some minimal amount of fruit. In the robot optimisation process, this ability is equivalent to an optimisation constraint, which fulfils conditions depending on the robot working environment. To make the environment constraint more realistic, an allowed unpicked fruit percentage is defined. This percentage depends on the economic aspects of harvesting. In this paper, the percentage was taken as 5%, meaning that in order to fulfil the task, the robot must be able to reach at least 95% of the targets. This value is typical for the apple growing farms, though, it can be changed according to a specific case.

2.7. Optimisation and motion planning algorithms

To solve the optimisation problem, a grid search method was used. The values of the optimisation variables were taken from a set of 7 uniformly distributed values between the limits ($[-\pi, \pi]$ for α and θ, and [0, 3] m for a and d). The cost function was evaluated for all the possible combinations of optimisation variables. The search for the optimal base position was found for a given number of robot positions, N_{pos}, using the same grid-search method.

The inverse kinematics of the RRR, RRP, and PPP robots was found using the inverse Jacobian Newton's numerical method (Siciliano, Sciavicco, Villani, & Oriolo, 2010). To simulate the robot and its scene, the arm and the tree were modelled by cylinders and planes. The collisions between the robot links and the tree branches, trellises and the ground were checked using the Quad Trees method (Finkel & Bentley, 1974).

The rapidly exploring random tree (RRT) algorithm (LaValle, 2006) was used as the motion planner of trajectories between the robot home position and the robot targets (fruit) similar to Bac et al. (2016). In our implementation, the RRT

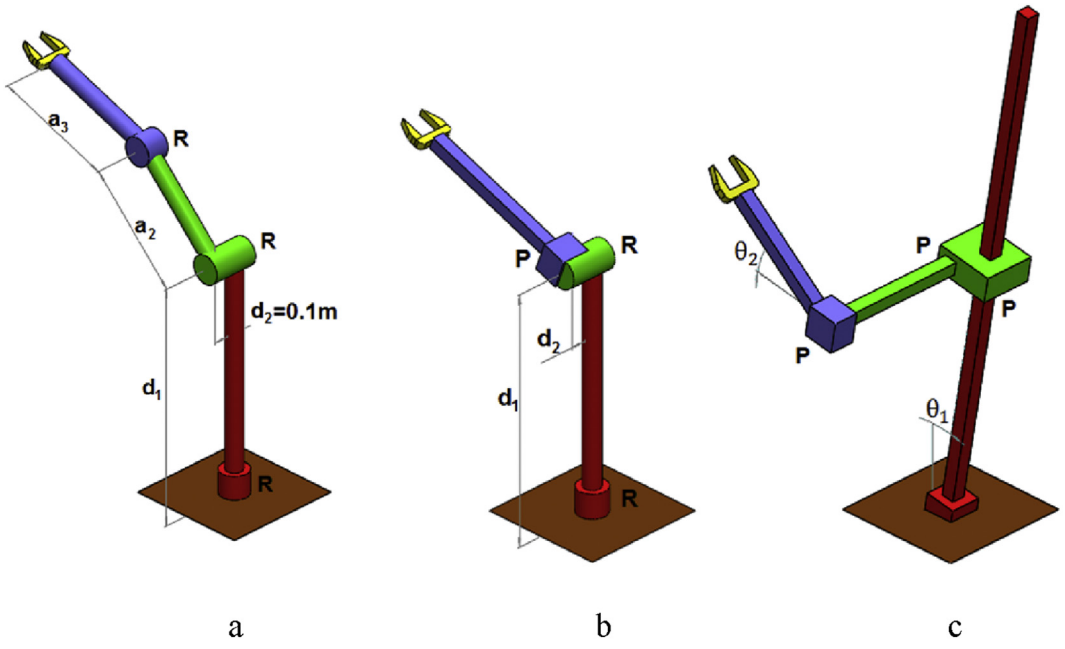


Fig. 4 – Robotic arms with 3 degrees of freedom of the robotic arm and different kinematic structures: (a) RRR, (b) RRP and (c) PPP. The optimisation variables are shown for each robot structure. The arms are mounted on a mobile platform which is not considered to be part of the arm's kinematics but were included in the optimisation of the position of the robot with respect to the tree in terms of X and Y coordinate.

used 100 vertices and an incremental distance of 0.03 m. To overcome the stochastic nature of the RRT method and achieve a path closest to the shortest, multiple reruns (up to 10) for the same scene were performed and the shortest path was used.

The evaluation time of the optimisation cost function was relatively long, between 2 and 20 min, depending on the tree type and shape. The calculation was performed using the MATLAB parallel computing tool on a computer with four cores (Intel® Core™ i5-2310 CPU 2.9 GHz, 4 GB RAM). On average, the inverse kinematics solution took 10% of this period of time, and the robot navigation solution with collision checking took 90%.

2.8. *Environment fitness evaluation*

The agronomic and economic advantages of these tree training systems, i.e., fruit quality, fruit yield and suitability for human harvesters, have been proven in previous agronomic research (e.g., Robinson et al., 2007). Hence, these aspects were not considered in the evaluation of the environment's fitness for robot performance.

To evaluate the environment's suitability, the cost-function F (Eq. (9)), representing the average fruit-picking time, was calculated. Lower F values correspond to better suitability of the environment to the robotic harvester.

2.9. *Preliminary tree optimisation*

Despite the complexity of the tree geometry, there are parameters that can be controlled by growers and changed in the tree models. One of them is the angle of the trellis tilt depicted by $\gamma_{trellis}$. Different $\gamma_{trellis}$ angles were examined to achieve an optimal YT tree model fitted to robotic harvesting.

This tree optimisation was performed for $\gamma_{trellis}$ values between 5° and 85°, with steps of 5°, assuming that the other tree geometry parameters of the L-systems rules are independent of $\gamma_{trellis}$.

3. Results

3.1. *Optimal robots for different environments*

In this section, the effectiveness of the robotic harvesting is compared on three existing tree training systems to find the optimal training system. Three optimal robots for the CL, TS and YT training systems are shown in Fig. 5. To illustrate the robot kinematics for each robot position, several robot configurations for the fruit picking are presented in the top row of Fig. 5. The optimal robot structures which were found are RRP for the CL and PPP for the TS and YT. In each image, the robot is located in N_{pos} optimal positions found during the robot base position optimisation.

The trajectories of the end-effector from the home configuration to the fruit picking configuration are shown in the lower row of Fig. 5.

A comparison of the different robot structures performing tasks in different environments, presented in Table 1, shows that each environment has its most effective robot structure, where F has minimal values (marked in bold): CL and YT environments require an RRR structure, and TS environment requires an RRP structure. The fact that not all environments have the same optimal robot structure can be explained by the

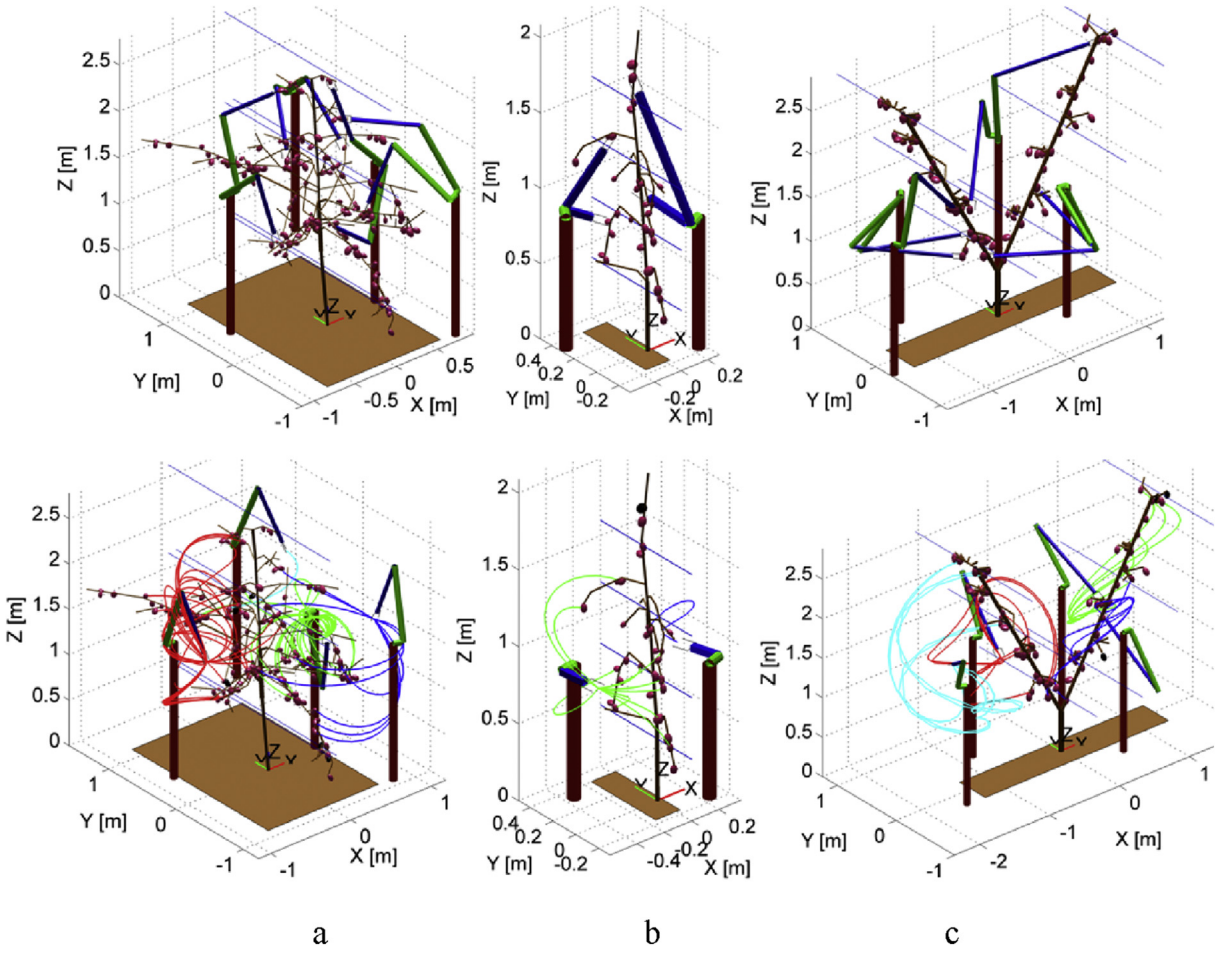


Fig. 5 – Optimal robots for (a) Central Leader, (b) Tall Spindle, and (c) Y-trellis apple trees located in optimal positions. The robots are presented for the several configurations for fruit picking (upper row) and home configurations and end-effector trajectories (lower row).

differences in the geometrical features of the environments. Most of the fruit on trees shaped by the TS system are surrounded by open space without obstacles, which enables the robot to approach the fruit by a straight line in the workspace from any robot base position. This type of motion is typical for the RRP (telescopic) structure robot. Trees shaped by the CL and YT systems have more fruit hidden behind branches which constitute obstacles for the robot motion on a straight line. To approach them, revolute joints must be involved in the robot structure. The RRR structure robot is suitable for this type of motion.

In all training systems and with all robot structures, the cost-function value decreases as the number of positions around the tree, N_{pos}, increases. The reason for this dependence is that the robot working volume is divided into smaller parts as the number of robot positions increases. As a consequence, the smaller the working volume, the shorter the lengths and mass of the robot links. This results in a decrease in the energy and time required for their motion. However, the **rate** of change **decreases** as N_{pos} increases which is further discussed in Section 4.

Table 1 enables to evaluate the fitness of the environment to robotic harvesting. The faster the optimal robot is, the better the training system is fitted to robotic harvesting. In this example, choosing the TS system provides an average picking time 4.2 times lower relative to the CL system for the robots located in two positions, and 2.4 times for the robots located in four positions.

Table 1 – Comparison of the robot performance cost function for optimal robots for different tree types and robot structures. (Dash symbols represent cases when no robot fulfilling the optimisation constraint was found.)

Robot structure	N_{pos}	Central Leader	Tall Spindle	Y-Trellis
		F [s]		
RRR	2	**1.35**	0.41	**1.24**
	4	**0.87**	0.28	**0.84**
	6	**0.71**	0.28	**0.81**
	8	**0.61**	0.28	**0.81**
RRP	2	–	**0.32**	2.4
	4	–	**0.26**	1.6
	6	1.13	**0.25**	1.45
	8	1.12	**0.25**	1.45
PPP	2	–	0.48	1.9
	4	–	0.43	1.1
	6	1.97	0.40	0.95
	8	1.87	0.40	0.90

3.2. Total robot performance time evaluation

Up to now, we considered harvesting of a single tree without movements between the robot base positions. In this section, the different training systems are compared in a more realistic case, in which the total harvesting time is calculated for a row of trees. The harvesting of multiple trees was considered including the motion time while harvesting a single tree and also moving from tree to tree. To understand the influence of movement time, T_{mov}, we considered the values of T_{mov} from the range [0,1,...,40] s, and evaluated the total time, T_{tot}, needed to pick 7920 fruit in orchards shaped by CL, TS and YT training systems.

We assumed that the trees in the orchard were multiple copies of the modelled trees considered in the previous example located along the rows. The number of fruit in each of the modelled trees was 144, 30 and 66 on the CL, TS and YT trees, respectively. Therefore, to model 7920 fruit (the least common multiple of 144, 30 and 66), the row must include the following number of trees $N_{tree} = 7920/N_{fruit}$, which resulted in 55, 264 and 120 trees for the CL, TS and YT orchards, respectively. The total time for the fruit picking was calculated as

$$T_{tot} = 7920 \cdot T_{fruit} + T_{mov} \cdot N_{pos} \cdot N_{tree},$$

where, T_{fruit} was the average time for the fruit picking which was in fact the cost function F, taken from Table 1. The total time for different N_{pos} and robot structures is presented in Fig. 6.

The optimal training system was defined by the minimal total time T_{tot}, which is the trade-off between the average fruit picking time and the number of movements. For T_{mov} close to zero (corresponding to fast passage between the harvesting positions), the optimal training system was found to be the TS with a PPP robot located at $N_{pos} = 6$ positions (corresponding to the minimal value in Table 1). Increasing the movement time but keeping T_{mov} below 14 s, the optimal combination is the TS training system with a PPP robot with $N_{pos} = 2$. For T_{mov} >14 s, the high density of TS becomes a disadvantage, making the number of trees and movements too high. In these cases, the CL with an RRP robot with a lower tree density becomes optimal.

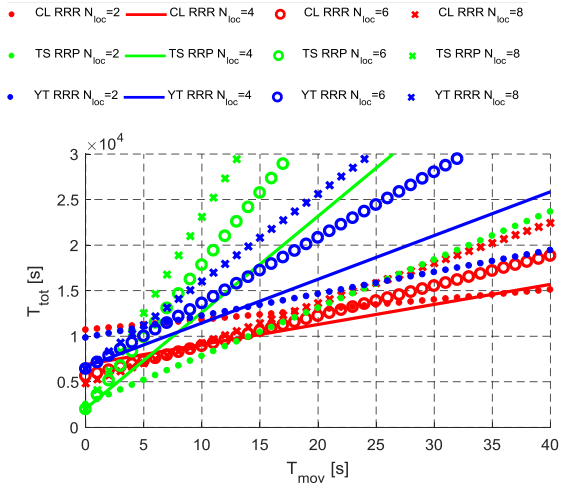


Fig. 6 – Total fruit picking time for different orchard types, robot structures and number of positions depending on the movement time T_{mov}.

3.3. Y-trellis tree structure optimisation

In this section, the performance of a single robotic arm is evaluated on single YT trees with different values of the tilt angle. The cost function F, depending on the Y-trellis tilt angle $\gamma_{trellis}$, is presented in Fig. 7. In section 3.1, the RRR and RRP robot structures were found as the most effective. Thus, in this section, these structures were taken as an example for the optimisation of the tree architecture. For each tree architecture, the optimal robots were calculated for the number of positions around the tree $N_{pos} = \{2, 4, 6, 8\}$. Similar to Sec 3.1, the time to move the robot base was not taken into account.

The graphs in Fig. 7 describe the relationship between the environment defined by the training system and its optimal robot. For the RRR robot structure (Fig. 7(a)), as the trellis tilt angle increases, the robot performance cost function decreases, meaning that the large angled Y-trellis system is better fitted for robotic harvesting. If agronomic and economic considerations do not permit such high tilt-angle values, 15° is optimal for the low angle values, and 35° (close to the tree model presented in Fig. 3b) is optimal for the middle angle values. In addition, locating the robot in two positions around the tree ($N_{pos} = 2$, one on each side of the tree row) causes relatively high values of the cost function, hence poor performance. The unordered behaviour of the graph for $N_{pos} = 2$ is the result of an insufficient number of robot positions. When a robot has only one position on each side of the row, it can only effectively pick fruit on the nearest side of the branch, whereas the fruit from the other side of the branch are picked much longer, in contrast to cases where N_{pos} is greater than 2.

The RRP robot structure (Fig. 7(b)) with $N_{pos} = 2$ is efficient only with low values of $\gamma_{trellis}$, where the Y-trellis is similar to a "fruiting wall" (presented in Fig. 3a). For $\gamma_{trellis} > 40°$, there are no solutions (none of the robots fulfil the constraint). The reason for this is the same as the reason for the unordered behaviour of the graph for $N_{pos} = 2$ for RRR, the inability of a robot in a single position on each side of a row to pick fruit from both sides of a branch. The fit of the environment decreases for $N_{pos} = 4$, and remains nearly constant for $N_{pos} = 6$ and $N_{pos} = 8$, when the tilt angle grows. Thus, for the RRP robot structure with $N_{pos} = 2$ and $N_{pos} = 4$, the most effective tilt angle for all values of N_{pos} is close to zero, producing a "fruiting wall". For a greater number of positions, the tilt angle has little effect on the cost function.

Assuming that any $\gamma_{trellis}$ angle can be realised, the optimal combination is the Y-trellis with $\gamma_{trellis} = 85°$ (presented in Fig. 3c) and a (RRR) robotic arm located in at least six positions around a tree.

4. Discussion

4.1. Assumptions

In this paper, a methodology for the simultaneous design of agricultural robot and its environment was demonstrated.

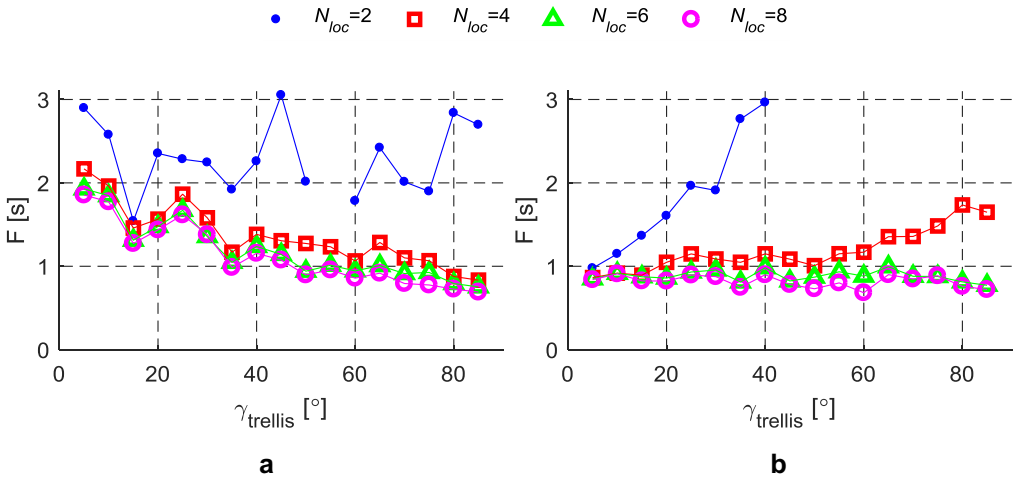


Fig. 7 – Performance cost function of the robot optimal for Y-trellis tree models. The trellis tilt angles are from the range [5°,10°,…85°]. The robot structures are (a) RRR and (b) RRP. The number of the robot positions around the tree N_{pos} is 2,4,6 and 8. Starting from 45° for the RRP robot $N_{pos} = 2$ there is no solutions for the optimal robots.

Nevertheless, the methodology was based on a number of assumptions required to simplify the definition and the solution of the design problem. Assumptions, such as the number of robot DOFs, power of the robot's actuators, mass of the apples and end-effector, mass of robot links, influence the optimal results mainly quantitatively and can be chosen appropriately for any specific case. However, the influence, of qualitative assumptions, such as definition of task, the cost function, robot locating, and robot universality should be further analysed.

The design of the robot environment is based on the optimality of the robot performance. However, the definition of the task, which has to be performed by the robot, is not strict. In this paper, the robots were optimised to perform apple harvesting on a single tree for a number of types of training systems. This task definition permits to demonstrate the design method with minimal effort for tree modelling and computation time (both aspects are further discussed). For more realistic cases, the task definition must be generalised. For example, the robot must be optimised for apple harvesting on an entire orchard trained by a specific system, apple harvesting on a number of orchards trained by a different system, harvesting of different fruits, branch pruning, disease monitoring, etc.

Increasing the universality of the robot, i.e., enlarging the spectrum of tasks for which the robot is designed for, is not always favourable. On the one hand, universal robots decrease the number of robots required to perform the farm's tasks. On the other hand, increasing the universality of the robot increases its complexity and also the average fruit picking time. The exact definition of the task depends on the goals of the robot designer and the requirements of the farm.

The definition of the robot performance cost function, i.e., measuring the efficiency of the robotic harvesting system, is a critical assumption. The most universal cost function is the farmer's income, however, it strongly depends on economic and agronomic factors, which were chosen not to be taken into account in this paper. The income was therefore simplified to the robot cost, initial (buying robots and creating the infrastructure) and maintenance (expenses for energy supply, upkeep, etc.), and was defined in terms of the robot kinematics and dynamics. The average fruit picking time, which was used in this paper, simultaneously measures the amount of picked fruit (maintenance cost) and the number of robots needed to perform the harvesting task in a specific time (initial cost).

The cost function used in this paper is the average time needed for a robotic arm to approach a fruit. This time depends only of the kinematic and dynamic specifications of the robotic arm. Actual fruit picking time also includes additional actions depending on the hardware, e.g., fruit detaching end-effector, environment sensing device, computation unit, etc., and software, e.g., fruit recognition, path planning, etc. The performance and development of the above sub-systems is typically separated from the robotic arm, hence, in this paper it is assumed that they have a negligible influence on the kinematics design.

4.2. *Results*

Values of fruit picking time were reported in other studies: robotic arm motion to a fruit is about 10 s in Bac et al. (2017), 6 s in Silwal et al. (2017), total harvesting cycle (without specifying the time of the robotic arm motion to a fruit) reviewed by Bac et al. (2014) is distributed "in a large range of 1–227 s with an average of 33 s (N = 28)". Relative to these values, the robot motion times achieved in this paper are low since this time depends on the assumed actuator power and includes only the robot motion time. To translate the simulated time to the actual robot, all the factors discussed above (hardware and software) must be taken in the consideration in the simulation. Nevertheless, the simulated values achieved in this paper are useful for comparisons between different robots in order to find the optimal robot structure.

The number of robot positions around a tree N_{pos}, considered in this paper, was chosen according to the

dimensions of the considered trees. Its minimal value, 2 (one from each row side), is sufficient for harvesting simple and narrow TS trees, while its maximal value, 8, is a required number for harvesting the larger CL trees. As was shown in Table 1, the efficiency of a robot increases when N_{pos} increases, since from each position the robot is closer to the fruit hence its path is shorter. Nevertheless, even for large N_{pos} the average distance between the home robot configuration and the fruit cannot be zero, and approaches some finite value when N_{pos} increases.

The high density training systems are based on growing thin trees in high density and reducing the number of branches. This provides two main advantages for the robot's operation. Concentration of the fruit in a compact volume (near the row plane at TS and along the trellis at YT) shortens the robot's link lengths and enables the robot to shorten the time of movement between the positions around the tree. In addition, the smaller number of branches provides a working volume with minimum obstacles. An additional non-mechanical advantage of the high density training systems is reducing the number of branches and leafs occluding the fruit, which can simplify fruit recognition and decrease the robot trajectory planning time. In this paper, the optimal robot for the TS tree is a telescopic robot RRP, which is able to approach the most of the fruit by a straight trajectory. This result corresponds with the conclusions achieved by Vougioukas et al. (2016), that the majority of fruit in high density tree architectures can be approached by a linear trajectory.

5. Conclusions

The methodology for simultaneous optimisation for the robot and its working environment is demonstrated in this paper. Three approaches of the methodology were shown: checking the existing tree training systems, considering the motion time of the robot platform, and tree simulation. All of these approaches should be used during a simultaneous design of the optimal robot and its environment to achieve the minimal cost of fruit harvesting. Ideally, the full simultaneous design, including all important agronomic and economic factors, must be performed.

The following conclusions were made based on the methodology developed. The comparison between different tree training systems shows that the harvesting time needed for high density training systems such as Tall Spindle (using the RRP robotic arm) and Y-trellis (using the RRR robotic arm) is less than the time needed for the CL tree training systems (Section 3.1). Therefore, these training systems are better fitted for robotic harvesting. In addition to the agronomical advantages of high density training systems (Robinson et al., 2007), they provide automation advantages.

The time required for the robot to move between different positions around the tree is crucial and can influence the optimality of the robot–environment combination. The analysis of the influence of the movement time should be performed by the system designer in order to choose the platform, an optimal robot structure or the number of the robot positions around a tree (Section 3.2).

The analysis of the Y-trellis tilt angle is an example for the optimisation of the tree shape with a single optimisation variable. Including the agronomic and economic considerations, the robot designer can achieve the most profitable robot–environment combination (Section 3.3). More effective optimisation should include additional parameters defining the tree shape by the L-systems method and limitations in agronomic ability to grow a tree with an arbitrary structure.

Future work must consider a number of generalisations. Agricultural environment has a large deviation in its geometrical features even for trees taken from the same row in an orchard. Hence, a single plant cannot represent the entire orchard. To achieve an optimal and robust robot and a reliable evaluation of the training system fit, sufficient sampled data (tree models) must be collected to characterise the tree training system. This will permit to evaluate the precision of the optimisation results, for example, figuring out if the local minimum in Fig. 7(a) is true for the entire orchard or just a local optimum for a specific tree. Nevertheless, we have to take into account that a large dataset leads to a growth in the computation time. To address this problem and to shorten the optimisation time, we propose to apply the method described in Bloch et al. (2017), which is based on building an average, characteristic tree model with the help of the minimal amount of data sufficient for the optimisation based on a group of trees. In the case of multiple trees, the robot positions are not related to the tree, as was assumed in this paper, but are related to the tree row.

As an initial proof-of-concept, to demonstrate our proposed methodology, only a subset of robot kinematic structures was chosen (three common kinematics out of eight possible structures for the 3-DOF structure). In future work, we intend to fully test all possible structures to see if they increase the efficiency of robot manipulation for these harvesting tasks. Similar to the trellis wires, additional supporting structures should be included in the tree models. Moreover, their location and shape must be included as optimisation variables. The methods of describing the environment, such as L-systems, can be used for an optimisation of the environment best suitable for robotic harvesting. Exact description of the tree geometry, topology and stochastic behaviour is complex and demands finding numerous parameters of the L-systems from field experiments. Future comprehensive studies will be based on tree models achieved by the digitiser or simulated according to the found parameters.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.biosystemseng.2017.11.006.

REFERENCES

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*. https://doi.org/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.
Bac, C. W., Roorda, T., Reshef, R., Berman, S., Hemming, J., & van Henten, E. J. (2016). Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. *Biosystems Engineering, 146*, 85–97.
Bechar, A. (2010). Automation and robotic in horticultural field production. *Stewart Postharvest Review, 6*(3), 1–11.
Belforte, G., Deboli, R., Gay, P., Piccarolo, P., & Aimonino, R. D. (2006). Robot design and testing for greenhouse applications. *Biosystems Engineering, 95*(3), 309–321.
Bergerman, M., Sanjiv, S., & Hamner, B. (2012). Results with autonomous vehicles operating in specialty crops. *Robotics and Automation International Conference on IEEE*.
Bloch, V., Bechar, A., & Degani, A. (2017). Development of an environment characterization methodology for optimal design of an agricultural robot. *Industrial Robot, 44*(1).
Bonora, E., Stefanelli, D., & Costa, G. (2013). Nectarine fruit ripening and quality assessed using the index of absorbance difference (IDA). *International Journal of Agronomy*.
Cho, S. I., Chang, S. J., Kim, Y. Y., & An, K. J. (2002). Development of a three-degrees-of-freedom robot for harvesting lettuce using machine vision and fuzzy logic control. *Biosystems Engineering, 82*(2), 143–149.
Costes, E., Lauri, P.-E., & Regnard, J. L. (2006). Analyzing fruit tree architecture: Implications for tree management and fruit production. *Horticultural Reviews, 32*, 1–61.
Costes, E., Smith, C., Renton, M., Guédon, Y., Prusinkiewicz, P., & Godin, C. (2008). MAppleT: Simulation of apple tree development using mixed stochastic and biomechanical models. *Functional Plant Biology, 35*(10), 936–950.
Craig, J. J. (2005). *Introduction to robotics: Mechanics and control*. Pearson/Prentice Hall.
Dawande, M. W., Geismar, H. N., Sethi, S. P., & Sriskandarajah, C. (2007). Throughput optimization in robotic cells. *International Series in Operations Research & Management Science, 101*, 420.
Edan, Y., Flash, T., Peiper, U. M., Shmulevich, I., & Sarig, Y. (1991). Near-minimum-time task planning for fruit-picking robots. *IEEE Transactions on Robotics and Automation, 7*(1), 48–56.
Edan, Y., & Miles, G. E. (1993). Design of an agricultural robot for harvesting melons. *Transactions of the ASAE, 36*(2), 593–603.
Finkel, R., & Bentley, J. L. (1974). Quad trees: A data structure for retrieval on composite keys. *Acta Informatica, 4*(1), 1–9. https://doi.org/10.1007/BF00288933.
Griffon, S., & Coligny, F. (2013). AMAPstudio: An editing and simulation software suite for plants architecture modelling. *Ecological Modelling, 290*, 3–10.
Han, S., Xueyan, S., Tiezhong, Z., Bin, Z., & Liming, X. (2007). Design optimisation and simulation of structure parameters of an eggplant picking robot. *New Zealand Journal of Agricultural Research, 50*(5), 959–964.
Hua, J., & Kang, M. (2013). Optimize tree shape: Targeting for best light interception. In *Proceedings of the 7th international conference on FSPM* (pp. 27–29).
Karwowski, R., & Prusinkiewicz, P. (2004). The L-system-based plant-modeling environment L-studion 4.0. In *Proceedings of the 4th international workshop on functional-structural plant models* (pp. 403–405).
LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press.
Lee, B. S., & Rosa, U. A. (2006). Development of a canopy volume reduction technique for easy assessment and harvesting of Valencia citrus fruits. *Transactions of the ASABE, 49*(6), 1695–1703.
Méndez, V., Rosell-Polo, R. J., Sanz, R., Escolà, A., & Catalána, H. (2014). Deciduous tree reconstruction algorithm based on cylinder fitting from mobile terrestrial laser scanned point clouds. *Biosystems Engineering, 124*, 78–88.
Preuksakarn, C., Boudon, F., Ferraro, P., Durand, J-B., Nikinmaa, E., & Godin, C. (2010). Reconstructing plant architecture from 3D laser scanner data. *Proc. of the 6th Intl. Workshop on Functional-Structural Plant Models*, 14–16.
Prusinkiewicz, P., Hammel, M., Hanan, J., & Mech, R. (1990). L-systems: From the theory to visual models of plants. In *Proceedings of the 2nd CSIRO symposium on computational challenges in life sciences* (Vol. 3).
Prusinkiewicz, P., & Lindenmayer, A. (1990). *The algorithmic beauty of plants*. New York: Springer.
Robinson, T. L., DeMarree, A. M., & Hoying, S. A. (2007). An economic comparison of five high density apple planting systems. *Acta Horticulturae*, 732-481-490.
Santos, T., & Ueda, J. (2013). Automatic 3D plant reconstruction from photographies, segmentation and classification of leaves and internodes using clustering. *Proceedings of the 7th international conference on functional-structural plant models*, 95–97.
Scarfe, A. J., Flemmer, R. C., Bakker, H. H., & Flemmer, C. L. (2009). Development of an autonomous kiwifruit picking robot. In *Autonomous robots and agents, 4th international conference* (pp. 380–384).
Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, planning and control*. Springer Science & Business Media.
Silwal, A., Davidson, J. R., Karkee, M., Mo, C., Zhang, Q., & Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics, 34*, 1140–1159.
Tanigakia, K., Fujiuraa, T., Akaseb, A., & Imagawa, J. (2008). Cherry-harvesting robot. *Computers and Electronics in Agriculture, 63*(1), 65–72.
Van Henten, E. J., Van Slot, D. A., Hol, C. W. J., & Van Willigenburg, L. G. (2009). Optimal manipulator design for a cucumber harvesting robot. *Computers and Electronics in Agriculture, 65*(2), 247–257.
Vougioukas, S., Arikapudi, R., & Munic, J. (2016). A study of fruit reachability in orchard trees by linear-only motion. *IFAC-PapersOnLine, 49*(16), 277–280.
http://www.ffrobotics.com/FFRobotics Ltd., Israel.
https://www.abundantrobotics.com/Abundant Robotics Ltd., USA.