Fitness of Diverse Orchard Architectures on Optimal Robot Manipulator

Victor Bloch, Avital Bechar and Amir Degani

***Abstract*— Achieving a low cost robotic arm is crucial in agricultural applications. Task based optimization of the robot kinematics influence the robot simplicity and cost. Nevertheless, the environment of the robot also has a major influence on its simplicity. We have simulated a variety of orchard architectures and searched for an optimal robot design for each architecture. From the training systems which we considered, the Tall Spindle system provides the minimal average time for fruit picking and thus is preferable for robotic harvesting.**

I. Introduction

Despite decades of research on robotic applications in agriculture, commercial agricultural harvesters are sparse or even nonexistent [1], [2]. Among others, the two main reasons are high cost of existing (industrial) robots and serviceability, making them unprofitable for farmers; and agricultural environment complexity, causing the sensing and motion planning of the robot to be complicated, time consuming and therefore impractical.

For agricultural applications, robotic arms are often tailor designed. They strive to be "light, simple and cheap" such as the arm for kiwi harvester [3]. Moreover, the robots are in some cases optimized for a specific task, such as an optimal robot for cucumber harvesting [4], or an optimal robot for eggplant harvesting [5]. However, up until now, the optimization was focused mainly on the robot component of the robot-environment system.

In the manufacturing domain, the robot environment is defined as the robot cell. Design of the robotic cells intended for throughput optimization is well studied and helps to solve numerous industrial challenges [6]. The main methods of the cell design are effective scheduling, multiple gripper usage, and parallel working robot usage.

Simplification and structuration of the agricultural environment is noted [2]. Nevertheless, design or optimization of the environment has not been performed. Such optimization is difficult because of the large number of optimization parameters, the difficulty in reaching the desired design, the extensive work required to design a tree and the required knowledge of the plant behavior (e.g., parameters of the L-systems [7]). Evaluation of the effectiveness of the existing environment types is performed in this paper as a preliminary step to the environment optimization.

Victor Bloch and Amir Degani are with Faculty of Civil and Environment Engineering, Technion, 32000 Haifa, Israel (Tel: 972-48292632; e-mail: {victorc, adegani}@technion.ac.il).

Victor Bloch and Avital Bechar are with Agricultural Research Organization (ARO), Volcani Center, Agricultural Engineering, 50250 Bet-Dagan, (e-mail: avital@volcani.agri.gov.il).

Different training systems (orchard architectures) of the fruit trees represent clearly different robotic environments. Modern high plant density training systems, such as the Tall Spindle (Figure 1 b), and Y-trellis (Figure 1 c), were developed mainly for increasing the yields and quality of fruit [8], [9]. In addition, they save labor time during harvesting, providing a convenient environment for the human harvesters. This advantage can also be used to provide an environment suitable to robotic harvesters, turning them into a profitable harvesting solution. The goal of this research is to evaluate the fitness of these training systems to robotic harvesting.

II. Problem Description and Formulation

A. Environment Modelling

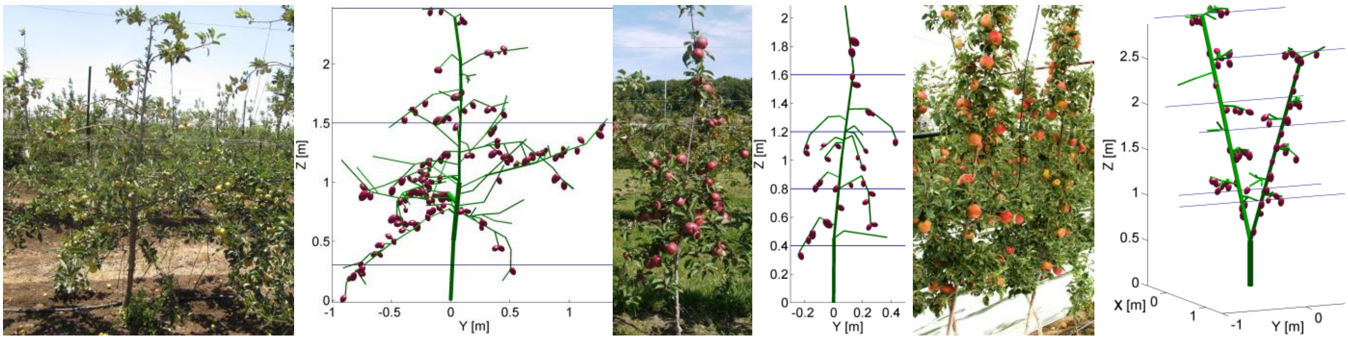
Three apple trees trained by Central Leader (CL), Tall Spindle (TS) and Y-trellis (YT) training systems were modeled as the robot environment (Figure 1). The tree model consists of cylinders modeling the segments of the branches and the trellis parts, ellipsoids modeling the fruits and a plane modeling the ground. The leaves were not modeled in this preliminary examination. The environment coordinate system has its origin in 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.

The CL tree was modeled with the help of a mechanical digitizer developed for this purpose [10], the TS tree was reconstructed from pictures, and the YT tree was approximated by tuning the parameters of the L-systems and Markov chain. Future comprehensive studies will be based on tree models achieved by the digitizer.

B. Robot Performance Cost Function

Robot performance cost function is used to evaluate the effectiveness of the robot. The effectiveness depends on the customer's demands, thus, the cost function cannot be strictly defined for the general case. One of the most common requirements in robotic applications is minimization of the robot performance time, which we use in this research. We study the mechanical aspect of the robot performance, hence, we consider only the time of the robot motion, which is defined here as the robot performance cost function F. This function depends on the robot geometry and the power of the robot actuators.

Exact calculation of the performance time is also nearly impossible in the general case. The time depends on parameters such as actuator power and weight, construction material, etc., which are defined by the designer. In order to find the optimal robot for our case study, the performance



(a) (b) (c)

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

time is evaluated by making several assumptions on the robot actuators and structure.

- Robot arm kinematics is predefined as a 3-degrees of freedom (DOF) with revolute or prismatic joints and general Denavit–Hartenberg (DH) parameters table. The number of DOF is denoted as N_{DOF}.
- Actuators are considered massless, assuming that the actuators are mounted on the robot base and are transmitting forces through beams or cords similar to the actuating system presented in [3].
- The length density of the robot links is taken as 1.5kg/m (similar to the density of a 3mm thick aluminum tube with a diameter of 60mm).
- The mass of the load is taken as 0.1kg (assumed as the mass of an average apple).
- The power of the robot actuators is taken with specific values: 100W is the power of the actuator in 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 link moved by the actuator.
- Time spent moving the mobile platform is not included in the cost function (because of the large variety of platform types). Nevertheless, a designer working with a specific mobile robot can take the moving time into account while evaluating the total robot cost function.
- The robot places the picked fruit in a gathering bin adjusted to the robot platform. Thus, the robot task must consist 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 precise computation of the time and energy of motion is cumbersome for mechanical systems with three or more DOF. Therefore, the time is approximated with the help of basic physical expressions. Assuming the robot geometry, link masses and inertia and actuator power, the time spent for picking a specific fruit is

$$t_{fr,i} = \max(E_i/W_i)\,, \tag{1}$$

where E_i is the energy consumed by the i's actuator, and W_i is the power of the i's actuator.

The energy $E_{fr} = \sum_{\forall i} E_i$ needed for the picking of a single fruit is the energy of the robot's movement from its initial home configuration q_i to a final configuration q_f set for picking the fruit. Therefore, the energy is calculated as

$$E_{fr} = E_d + E_s + E_{damp}, \tag{2}$$

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 robot's weight, and E_{damp} is the damping work against the friction in the robot joints. Friction in the robot joints depends on the gearbox parameters chosen by the designer, hence, it is disregarded therein. The static work is evaluated as

$$E_s = \int_{q_i}^{q_f} \tau(q)\, dq. \tag{3}$$

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

$$\tau = J^T P, \tag{4}$$

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 by the link length and length density, with the weight applied at the middle of each link.

The dynamic work, E_d, is calculated by

$$E_d = \int_{q_i}^{q_f} I(q)\, \ddot{q}\, dq\ , \tag{5}$$

where I is the inertia of the robot links and load, depending on the robot configuration. It is assumed that the acceleration has a maximal value (bang-bang control), hence, it is taken as a constant. Finally, the dynamic work can be approximated as

$$E_d = \ddot{q}_{max} \int_{q_i}^{q_f} I(q)\, dq\ . \tag{6}$$

The cost function F is the average fruit picking time for all picked fruits N_{picked}

$$F = \left(\sum_{i=1}^{N_{picked}} t_{fr,i}\right) / N_{picked}\ . \tag{7}$$

The cost function F is measured in units of time (seconds). The time calculation is approximated and simplified, and does not include important, but non-mechanical and difficult to define factors such as fruit recognition time, trajectory planning time, time for fruit detachment and placement in the gathering bin, etc. Therefore, the achieved time values are different from the values reported in previous researches, such as [4].

C. Optimization Parameters

The parameters of the optimization are the known DH convention parameters α, θ, a and d [12]. The total number of parameters defining the robot kinematics is 4xN_{DOF}. The parameters representing the robot's degrees of freedom (θ and d) are found by solving the inverse kinematics at a specific configuration of the robot. Thus, the total number of free unconstrained optimization parameters is 3xN_{DOF}.

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 orders: RRR, RRP and PPP.

The location of the robot base constitutes additional optimization parameters. Each location is defined by two parameters: X and Y coordinate on the ground plane. Searching the optimal base location is decoupled from the optimization of the robot kinematics, and is found for a given number of robot locations N_{loc} by the grid search method with the branch and bound algorithm [11].

We also define the limits of the optimization parameters fitted to the task as follows:

- α parameter is in the interval [-π, π],
- θ parameter is in the interval [-π, π],
- a and d parameters are taken in the interval [0,3], considering that the height and width of orchard trees do not exceed 3m,
- similarly, the X and Y coordinates of the robot base location are taken from the interval [-3, 3].
- The limits of the robot DOF depend on the mechanical design of the robot. Hence, in this paper the robot revolute parameters Θ are inside the interval [-π, π], and the robot prismatic parameter d is taken from the interval [0,3].

D. Optimization Constraint

The environment constraint is a set of geometric models of all obstacles and targets in the robot task environment. Interaction with these objects influences the robot motion: the robot must approach a target without collision with the obstacles.

To make the environment constraint more realistic, the allowed unpicked fruit percentage is defined. This percentage depends on the economic aspects of the fruit picking. In this paper, the percentage is taken 5%, meaning that in order to fulfill the task, the robot must be able to approach at least 95% of the targets.

E. Optimization and Navigation Algorithm

The optimization problem has a relatively large number of parameters, $3N_{DOF}$=3x3=9, and a long function evaluation time: 2 to 20 minutes, depending on the tree. On average, the inverse kinematics solution takes 10% of this period of time, and the robot navigation solution with collision check takes 90%. To solve this problem, a Genetic Algorithm is used with a population size of 200 and a mutation rate of 20%.

The rapidly exploring random tree (RRT) algorithm [13] is used as the planner of trajectories between the robot home position and the robot targets (fruit). The RRT uses 100 vertices and an incremental distance of 0.03m. The original version of RRT was implemented.

F. Environment Fitness Evaluation

The agronomical and economical aspects of the tree shaping, such as fruit quality and yields, are out of scope of this research. Hence, these aspects are not considered in the evaluation of the environment fitness to the robot performance (especially, considering the fact that originally the tree training was being performed for purposes not connected with robotic harvesting).

In order to find the environment fitness evaluation, we calculate a single fruit picking average time. The lower the average time is, the more fitted the environment is to the robotic harvester.

III. Results and Discussion

A. Optimal Robots for Different Environments

Three optimal robots for different environments are shown in Figure 2. For illustrations of the robot kinematics for each robot location, several fruits and robot homing configurations are presented in the upper part of Figure 2. The unpicked fruit are colored in black.

The trajectories of the end-effector from the homing configuration to the fruit picking configuration are shown in the lower part. We can observe that the shorter the trajectories are, the shorter the motion time is and, therefore, the robot is more optimized.

Table 1 shows the relation between the environment types, the robot cost function F, the order and type of the joints, and the number of robot locations around the tree N_{loc}.

In all the environments and robot types we can observe that the cost function value decreases as the number of locations around the tree $N_{loc,}$ increases. The reason for this dependence is that the robot working volume is divided into smaller parts as the number of the robot locations increases. As a consequence, the smaller the working volume, the shorter the lengths of the robot links and the smaller their

TABLE I. Average cost of the fruit picking by the optimal robot for different environments

		Central Leader	***Tall Spindle***	***Y-Trellis***
Joint type	N_{loc}	F [second]		
RRR	2	0.81	0.35	0.44
	4	0.64	0.3	0.42
	6	0.58	0.24	0.38
	8	0.55	0.23	0.37
RRP	2	0.33	0.17	0.27
	4	0.27	0.15	0.25
	6	0.26	0.12	0.21
	8	0.26	0.1	0.2
PPP	2	No solution	0.085	0.22
	4	0.4	0.079	0.17
	6	0.32	0.06	0.16
	8	0.32	0.06	0.16

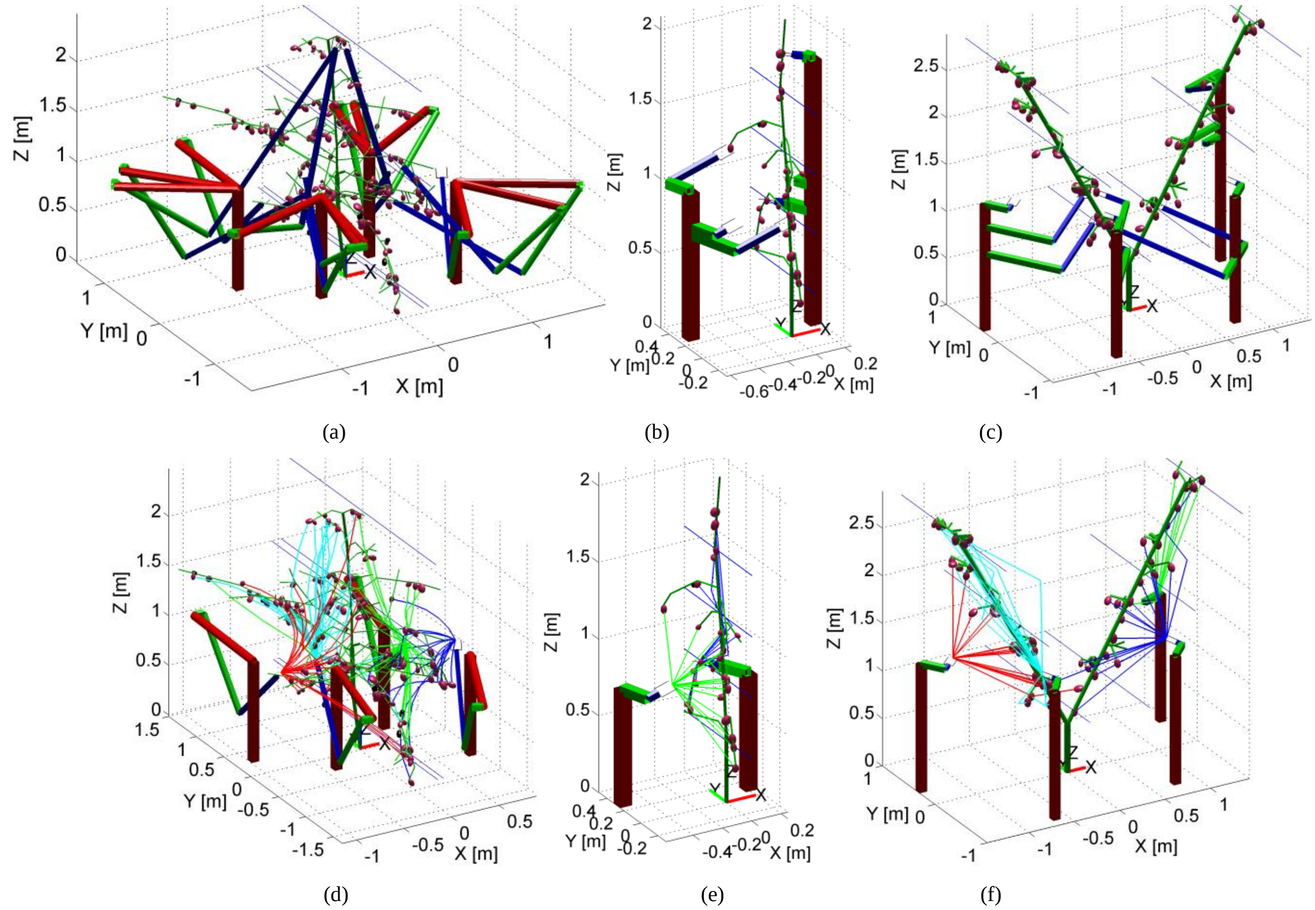


Figure 2. Optimal robots and their end-effector trajectories for Central Leader (a, d), Tall Spindle (b, e), and Y-trellis (c, f) apple trees. The robots are presented in the several configurations for fruit picking (upper row) and homing configurations (lower row).

masses are, resulting in a decrease in the energy and the time needed for their motion.

However, while the cost function decreases as the number of robot locations increases till six, for eight locations the cost function almost does not change. This can be observed in all robot types, and points at the fact that the larger number of robot locations has smaller efficiency.

Comparison of the robot types for each environment type shows that each environment has its most effective robot type, where the cost functions have minimal values: TS and YT environments have PPP, and CL environment has RRP. This is due to the geometrical features of the environments. Trees shaped by the TS and YT methods have relatively structured environments. As a result, most of the fruit is 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 location. This type of motion is typical for the PPP type robot. Trees shaped by the CL method have more fruit hidden by branches which constitute obstacles to the robot motion on a straight line. To approach them, revolute joints must be involved in the robot structure. The RRP type robot is suitable for this type of motion.

The fact that each environment type has a specific robot type best suited to operate in it enables an evaluation of the fitness of the environment to robotic harvesting. The simpler the optimal robot is, the better fitted its designated environment will be to robotic harvesting.

Thus, according to the defined cost function, a prismatic joint is more effective than a revolute joint, since during the motion it changes only the position of its link and end-effector, while the revolute joint changes orientation as well as position for the same motion. Hence, the more prismatic joints a robot has, the more efficient it is. Consequently, TS and YT training systems are more effective than CL training systems, and, according to the cost function values, the TS method is more effective than the YT method.

B. *Total Robot Motion Time Evaluation*

The movements between the robot base locations must also be considered in the evaluation of the cost function for a row or entire orchard. The time of the movement depends on the platform carrying the robotic arm and is not evaluated here for the general case. To understand its influence, we propose to define the average movement time as 3 seconds (denoted as T_{mov}) and evaluate the time needed for picking 7920 fruit in orchards shaped by the considered methods.

We assume that the orchard consists of the modeled trees duplicated and located along the rows. The number of fruit on each tree, N_{fruit}, is given in Table 2. Hence, to model 7920 fruit, the row must include the following number of trees N_{tree}=7920/N_{fruit} (given in Table 2). The total time for the

TABLE II. AVERAGE COST OF THE FRUIT PICKING BY THE OPTIMAL ROBOT FOR DIFFERENT ENVIRONMENTS

	Environment and Tree Type		
	CL, RRP	***TS, PPP***	***YT, PPP***
N_{fruit}	144	30	66
N_{tree}	55	264	120
T_{tot}, N_{loc}=2	2944 sec	**2257** sec	**2462** sec
T_{tot}, N_{loc}=4	**2798** sec	3794 sec	2786 sec
T_{tot}, N_{loc}=6	3049 sec	5227 sec	3427 sec
T_{tot}, N_{loc}=8	3379 sec	6811 sec	4147 sec

fruit picking is calculated according to

$$T_{tot} = T_{fruit}N_{fruit} + T_{mov}N_{tree}N_{loc}, \qquad (8)$$

where T_{fruit} is the average time for the fruit picking equal to the cost function value F. The total time for different N_{loc} and specific robot types is presented in Table 2.

The results presented in Table 2 show that the TS training system is the most effective. The total fruit picking time depends on the number of trees and the number of locations around a single tree, and has an optimal value (shown in bold), which is the tradeoff between the average fruit picking time and the number of movements.

IV. CONCLUSION

We propose to evaluate the fitness of the orchard tree architecture to robotic harvesting. The comparison of the robot performance cost function for different tree training systems shows that high density training systems have a structure that demands a simpler optimal robot to perform the fruit picking in less time. Therefore, these systems are better fitted to robotic harvesting than the conventional systems. In addition to the agronomical advantages of high density training systems [8], they provide automation advantages.

Tree training according to high density training systems is performed by growing high trees in high density and eliminating small branches. This provides three main advantages. 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 locations around the tree. In addition, a plane-shaped tree (so called "fruit wall") enables the positioning of the robot close to its targets and orients its joints along the plane, making the end-effector motion close to two-dimensional and allowing the use of prismatic joints. Finally, the decreased number of branches provides a working volume with minimum obstacles. An additional non-mechanical advantage of tree training according to high density training systems is the decreased number of branches and leafs occluding the fruit, which can simplify fruit recognition.

The stochastic nature of the agricultural environment, which includes a large number of unordered objects, makes the designing process extremely time consuming. To complete a design in an acceptable amount of time, more effective methods, such as environment characterization [10], should be applied.

Future work must consider the following generalizations. Agricultural environment has a large deviation in its geometrical features even for trees taken from the same row in an orchard [10]. Hence, a single plant cannot represent the entire orchard. To achieve an optimal and robust robot and a reliable evaluation of the training system fitness, we must collect enough sampled data (tree models) characterizing the tree training system. 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 optimization time, our method described in [10] can be used. The method is based on building an average, characteristic tree model with the help of the minimal amount of data sufficient for the optimization.

Trellis cables and supports represent obstacles for the robot. We will try to offer engineering recommendations defining the positioning of these supports in a way which minimizes their interference with the operation of the robot.

Methods of describing the environment, such as L-systems and Markov chains, can be used as parameters for an optimization of the environment best suitable to robotic harvesting.

REFERENCES

[1] A. Bechar, "Automation and robotic in horticultural field production," in *Stewart Postharvest Review*, 6(3), 1-11, 2010.

[2] C. W. Bac, E. J. Van Henten, J. Hemming, Y. Edan, "Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead," in *Journal of Field Robotics*, 31(6), 888-911, 2014.

[3] A. J. Scarfe, R. C. Flemmer, H. H. Bakker, C. L. Flemmer, "Development of an autonomous kiwifruit picking robot," in *Autonomous Robots and Agents, 4th International Conference*, 380–384, 2009.

[4] E. J. Van Henten, D. A. Van Slot, C. W. J. Hol, and L.G. Van Willigenburg, "Optimal manipulator design for a cucumber harvesting robot," in *Computers and Electronics in Agriculture*, 65(2), 247–257, 2009.

[5] S. Han, S. Xueyan, Z. Tiezhong, Z. Bin, and X. Liming, "Design optimisation and simulation of structure parameters of an eggplant picking robot," in *New Zealand Journal of Agricultural Research*, 50(5), 959–964, 2007.

[6] M. W. Dawande, H. N. Geismar, S. P. Sethi, C. Sriskandarajah, "Throughput Optimization in Robotic Cells," in *International Series in Operations Research & Management Science*, vol. 101, 2007, 420 p.

[7] J. Hua, M. Kang, "Optimize Tree Shape: Targeting for Best Light Interception," in *Proceedings of the 7th International Conference on FSPM*, 27-29. 2013.

[8] T. L. Robinson, A. N. Lakso, Z. Ren, "Modifying Apple Tree Canopies for Improved Production Efficiency," in *HortScience*, vol. 26, no. 8., 1991, pp. 1005-1012.

[9] M. Bergerman, S. Sanjiv and B. Hamner, "Results with autonomous vehicles operating in specialty crops," in *Robotics and Automation International Conference on. IEEE*, 2012.

[10] V. Bloch, A. Bechar, A. Degani, "Development of an Environment Characterization Methodology for Optimal Design of an Agricultural Robot," in preparation.

[11] P. T. Kolesar. "A Branch and Bound Algorithm for the Knapsack Problem", in *Management Science*, vol. 13, No. 9, May 1967.

[12] J. J. Craig. "Introduction to Robotics: Mechanics and Control," Pearson/Prentice Hall, 400 pages, 2005.

[13] S. M. LaValle. "Planning algorithms," Cambridge University Press, 842 pages, 2006.