Available online at www.eccomasproceedia.org
Eccomas Proceedia EUROGEN (2023) 125-131

ECCOMAS
Proceedia

EUROGEN 2023
15th ECCOMAS Thematic Conference on
Evolutionary and Deterministic Methods for Design, Optimization and Control
N. Gauger, K. Giannakoglou, M. Papadrakakis, J. Periaux (eds.)
Chania, Crete Greece, 1–3 June 2023

A METHODOLOGY TO CHARACTERIZE AN OPTIMAL ROBOTIC MANIPULATOR FOR SELECTIVE SPRAYING IN VINEYARDS

R. Azriel[1,2] **and A. Bechar**[1,2]

[1] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev
Beer-Sheva 8410501, Israel
e-mail: roniaz@post.bgu.ac.il

[2] Institute of Agricultural Engineering, Agricultural Research Organization, Volcani Institute
Rishon LeZion 7505101, Israel
avital@volcani.agri.gov.il

Abstract

In this paper a methodology is proposed for optimizing the design of a robotic manipulator specifically for selective spraying of nutrients in vineyards, a well-defined agricultural task. The optimization was based on maximizing the Manipulability Index. A simulator was developed using robotic software and tools, to assess the performance of each potential robotic configuration in a simulated vineyard. The optimization process utilized the Particle Swarm Optimization (PSO) algorithm within a constrained solution space. In addition, XGBoost models were designed to predict whether a robot configuration could reach all the desired grape cluster positions in order to reduce computational time. Results show that the learning models were able to correctly classify with average precision of 82.4% and 83.1% for the positive and negative classes respectively. The use of the XGBoost model was found to reduce running time by 42.7% and therefore greatly benefit the optimization process. Furthermore, several reasonable optimal configurations of manipulators were provided using the presented approach. Overall, this methodology presents a promising approach for optimizing the design of robotic manipulators for agricultural tasks, with potential for applications in other industries as well.

Keywords: Robotic manipulator, Optimal design, Simulation, Evolutionary algorithm, Machine learning.

Peer-review under responsibility of the organizing committee of EUROGEN 2023.
doi: 10.7712/140123.10194.19076

1 INTRODUCTION

Robotic technology has been developed to automate tasks that are repetitive, dull, strenuous, or hazardous. Robots could be highly advantageous in performing agricultural tasks like weeding, harvesting, and spraying, for these reasons. However, while robots have been successfully implemented in industry, their use in agriculture has been restricted. Industrial environments are more predictable and uniform, while agriculture faces a variety of challenges, such as weather, terrain, and crops that vary in color, size, and shape and can be easily damaged during handling [1-2]. In addition, despite decades of research, commercial crop robot applications remain constrained, mainly due to the high cost of existing robots, their maintenance requirements, and their limited performance. Nevertheless, there is an economic incentive to use automation in agriculture, particularly in countries with high labor costs. A possible solution to overcome these difficulties is to find a simple robotic arm that is best suited for the task and environment in which it will be operated.

In the previous years, several approaches have been proposed to optimize robot parameters based on different performance criteria. Zhou and Bai [3] presented in their study a method for optimizing a robotic arm to minimize the weight of the robot with constraints on kinematics performance, drive-train, and structural strength. The kinematic performance of the robot was indexed by the global conditioning index (GCI). The optimization method was implemented by using five modules, including a computer aided design (CAD), a kinematic and dynamic simulation, a finite element analysis system module, and Complex optimization method. Xiao et al. [4] focuses as well on optimizing the total mass but also the manipulability of a robotic arm based on non-dominated sorting genetic algorithm and Pareto fronts. The first objective was to find the lightest combination of motor and gearbox for all six degrees of freedom (DOF) of UR5 robot and the optimal length and thickness of several links that fulfill all constraints associated with the motors and gearboxes. The second objective defined as the manipulability of UR5 in a given trajectory.

In this study, the optimal robot is task oriented and aimed to perform the task of nutrients spraying table grapes in vineyards. Table grapes have a total production of 23.7 million tons per year in Europe only. Gibberellin, a plant hormone, is widely used today in grape production at different concentrations during berry development and growth to increase yield, and improve quality characteristics of berries and clusters [5]. It is hand sprayed selectively, targeted only to the grapefruit, or delivered by hand dipping the grape clusters. This both applications are time-consuming, labor intensive, inaccurate, and results in major overdoses [6]. The utilization of a robotic system would enhance this task by enabling quick and precise delivery.

This study proposes a methodology for optimizing the structure of task-oriented robotic manipulator based on XGBoost architecture and Particle Swarm Optimization algorithm (PSO), for the task of selectively spraying nutrients in vineyards. The study considers the manipulability index as the performance measure, which was constrained by the reachability capacity. To collect the necessary data, the Gazebo simulator was employed as the data acquisition tool.

2 PROBLEM FORMULATION

2.1 Model description

The end effector payload is lightweight, so the structure of the robot could be simple in design, compact and dexterous. Taking into consideration the characteristics of the robot and the work requirements, a serial manipulator structure, a series of links connected by motor-actuated joints, was adopted for the robot. Upon setting the robot's structure, DOF must be determined. Having fewer DOF may simplify the structure, but the robot may not be able to

perform its work adequately. Alternatively, too many DOF would complicate the calculation and make controlling the robot more difficult. Therefore, the solution space was limited to robots with 6 DOF, similarly to industrial robots exists nowadays. As a derivative, three categories of independent variables emerge: link lengths, denoted as $L = [l_1, l_2 \dots l_6]$, joint motion types, denoted as $J_1 = [j_{1,1}, j_{1,2} \dots j_{1,6}]$, and joint movement axes, denoted as $J_2 = [j_{2,1}, j_{2,2} \dots j_{2,6}]$. The link lengths considered are [0.1, 0.3, 0.5, 0.7] meters and the joint type can either be prismatic (allows linear motion in one direction), roll (revolute joint about the Z-axis in the element coordinate system) or pitch (revolute joint about the Y-axis in the element coordinate system). The movement axis can be X, Y, or Z. Manipulator configuration is therefore denoted as $x = [L, J_1, J_2]$.

2.2 The optimization problem

The kinematics performance of a robotic manipulator is crucial in its design, and various performance indices are available to evaluate the robot's abilities. Designers can use these indices to assess different designs and select the most suitable manipulator for a given application. In the case of agricultural tasks, such as spraying grape clusters, the primary requirement for the robot is its manipulation abilities. Since grape clusters are often located in irregular and hard-to-reach areas, manipulability is necessary to evaluate the performance of the robot. The Manipulability Index presented by Yoshikawa [7], created a mathematical measures for the manipulability of any serial robot. The concept of this manipulability index is centered around the ability of a robotic arm to position and reorient its end-effector. It also gives information regarding the proximity of singular positions, in which the robot's end-effector becomes blocked in certain directions. The manipulability index is based on the Jacobian matrix (J) which provides the relation between joint velocities and the end-effector velocities, in a certain joints angles. The manipulability index for specific position, is calculated as follows:

$$\mu = \sqrt{\det\left(JJ^T\right)} \tag{1}$$

The simulation measures the manipulability index for several desired end-effector positions and orientations. Optimizing the minimum obtained will guarantee the performance. Hence for n desired positions, the objective function is as follows:

$$f_1(x) = \min\left(\mu_1(x), \mu_2(x) \dots \mu_n(x)\right) \tag{2}$$

Once environmental and conceptual assumptions have been used to condense the solution space for this optimization, the remain constraint is maximum reachability (R), meaning the robot's ability to reach all desired positions. Reaching single position is denoted as r_i, so the constraint:

$$R = \sum_{i=1}^{n} r_i \tag{3}$$

Finally, the optimization problem is defined as:

$$\max_x f_1(x) \tag{4}$$

$$\text{s.t } R = n \tag{5}$$

3 METHODS

3.1 Simulation

To evaluate the performance of each robot configuration, Gazebo and RVIZ simulators were used with Robot Operating System (ROS), melodic version, and MOVEIT, a motion planning

framework of ROS in Linux environment [8]. Those tools enable motion planning, inverse and forward kinematics, control, and visualization. The simulation aimed to realistically demonstrate the execution of a spraying task, using data collected from the experimental vineyard.

The robot's workspace was affected by the maximum and minimum heights of the grape clusters in the vineyard's trees and by the "Y" shaped trellising (Figure 3). The manipulator was based on a platform moving orthogonally to trees, with the Y axis being a DOF for the system. To simulate the movement of the platform, a beam was added to create a free axis of movement (Figure 1). The simulation involved reaching various positions that represented both typical and extreme positions of actual grape clusters.

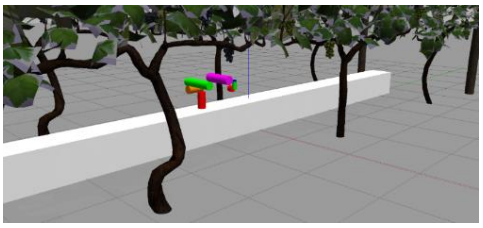

Figure 1: Illustration of the simulation.

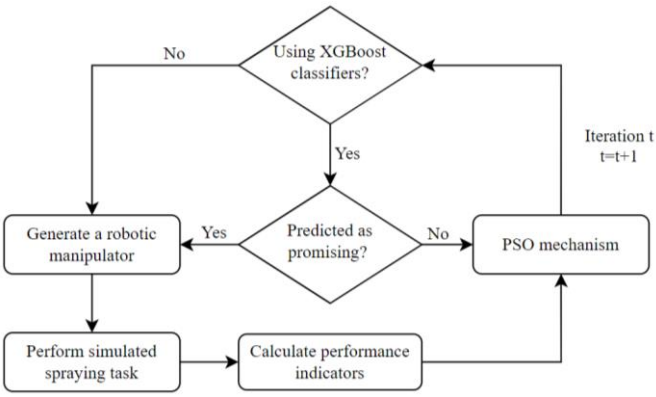

Figure 2: The experimental optimization framework.

Figure 3: The experimental vineyard and the prototype system.

3.2 Optimization framework

The optimization process is embedded as a design optimization platform containing three components: a kinematic simulation to calculate the objective function (as described in section 3.1), an optimization algorithm that manages the search for the optimal solution and a learning model as a threshold condition for testing a potential solution in the simulator.

This optimization problem is solved by an evolutionary algorithm, Particle swarm optimization (PSO), a method suitable for nonlinear problems with complex search space. PSO is a stochastic algorithm based on swarm intelligence, initially proposed by Kennedy and Eberhart [9]. This approach draws inspiration from the social behavior of animals such as fish schooling and bird flocking. The algorithm treats each potential solution to a given problem as a particle with a unique velocity, traveling through the problem space. These particles then utilize a combination of their own historical best location and current location, along with those of other swarm agents, to determine their next movement through the search space, with some random

disturbances added in. This process is repeated for all particles before the next iteration occurs. Over time, the swarm as a whole, much like a flock of birds searching for food, is expected to gradually approach the optimum of the objective function. This evolutionary algorithm is robust and suitable for variety of optimization problems and can converge quickly [10].

In order to further optimize the performance of the optimization process, a machine learning (ML) models, XGBoost architecture, was trained in order to predict by the manipulator configuration the success in reaching the desired grape clusters positions. XGBoost, is a gradient boosting tree algorithm that achieves state-of-the-art results on numerous tabular datasets. XGBoost creates new models from previous models' residuals and then combining them to make the final prediction, and uses gradient descent to minimize loss when adding a new model [11]. Each XGBoost classifier was trained independently to classify different grape cluster (GC) as a target, while reaching it is denoted as the positive class.

To assess the proposed methodology and the benefits of ML classifiers in terms of runtime and objective function score, an initial experiment was performed. This experiment aimed to determine the feasibility of utilizing the simulator and PSO to identify an optimal robotic configuration. Additionally, it tested the performance of the same methodology in conjunction with the XGBOOST models as a threshold for evaluating potential solutions in the simulator (Figure 2).The simulation task included reaching ten grape clusters and the objective defined as the minimum manipulability index for the ten reaches, for the manipulators succeed reaching them all, and zero otherwise. Priorly, the PSO hyperparameters were adjusted using a grid search and based on the characteristics of the problem and running time limitations, the number of generations (iterations) was set to 200.

Position predicted	NPV	PPV
GC1	0.884	0.862
GC2	0.865	0.856
GC3	0.843	0.82
GC4	0.835	0.827
GC5	0.85	0.833
GC6	0.835	0.822
GC7	0.815	0.804
GC8	0.819	0.803
GC9	0.825	0.815
GC10	0.794	0.795
Average	0.831	0.824

Table 1: XGBoost classifiers results.

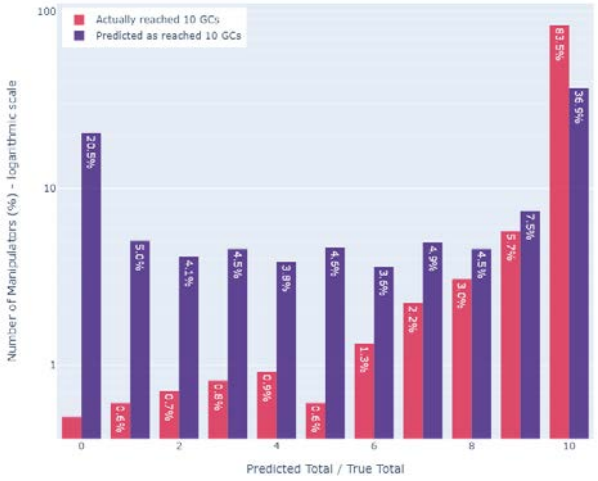

Figure 4: Manipulators reaching 10 GCs (or classified as such).

4 RESULTS

Table 1 displays the average cross-validate performance of the XGBoost trained models. The false positive (FP) prediction error involved with missing promising manipulators and can lead to missing the optimum. The false negative (FN) prediction error results in runtime costs. Furthermore, the datasets utilized to train and evaluate all classifiers was imbalanced. Therefore, the most suitable metrics for evaluating the performance of the machine learning models were Precision or Positive Predictive Value (PPV) and Negative Predictive Value (NPV). PPV measures the ratio of true positive predictions considering all positive predictions. NPV measures the ratio of true negative predictions considering all negative predictions. The results showed that all models achieved an average PPV of 82.4% and an average NPV of 83.1%.

Another analysis was conducted regarding the most critical group of manipulators to identify, those that could reach all grape clusters and among them the optimum is found. As shown in Figure 4, 83.5% of these promising manipulators were correctly identified, where 5.7%, 3%, and 2.2% of them identified with one, two, or three grape clusters errors, respectively. Therefore, a threshold was set such that all manipulators predicted to reach at least seven clusters were sent to simulation, while the rest received a score of zero for the objective. On the other hand, Figure 4 also shows the distribution of the manipulators classified as promising. Only 36.9% of them actually reached all the positions, while 20.5% failed to reach any desired location. These predictions errors result in runtime expended on unsuccessful manipulators.

Optimized designs of the robotic manipulator are listed at Table 2, obtained by the experiment described in section 3.2. By combining PSO and XGBoost modules, the objective function results obtained was 52.67% higher than using only PSO. The two methodologies yielded different optimal configurations. Incorporating the XGBoost classifier had a significant advantage in terms of runtime, reducing processing time by 42.7%.

	PSO			PSO and XGBoost		
Kinematic serial index	Joint Type	Joint Axis	Link Length [m]	Joint Type	Joint Axis	Link Length [m]
1	Roll	Z	0.1	Roll	Z	0.1
2	Pitch	Y	0.1	Roll	Y	0.1
3	Roll	Z	0.7	Roll	Y	0.5
4	Prismatic	Z	0.7	Prismatic	Z	0.7
5	Pitch	Y	0.1	Pitch	Y	0.1
6	Roll	Z	0.1	Pitch	X	0.1

Table 2: Optimal design obtained by PSO with and without XGBoost classifiers.

5 CONCLUSIONS

In this study, an integrated approach for finding the optimal design of a task-oriented robot manipulator was presented. Selections of the manipulator configuration parameters, joints, and links characteristics were formulated as a discrete optimization problem, which was solved by non-linear optimization method. Preliminary results show that the methodology proposed can achieve an optimal design, while satisfying the constraints of reaching all the desired positions. The benefits of the learning model for the optimization process are twofold: 1) The use of XGBoost as a threshold for using the simulation accelerate the convergence of the PSO algorithm; 2) By reducing the computational time, at the same number of generations, the swarm potentially observes better possible solutions and therefore is expected to find better solutions. This methodology can be applicable for finding optimal robotic manipulator for other application with similar characteristics.

REFERENCES

[1] A. A. W. Ahmed, J. Markendahl, and A. Ghanbari, “Development of An Autonomous Kiwifruit Picking Robot,” vol. 1. pp. 1–5, 2013, doi: 10.4236/jssm.2010.

[2] O. Spykman, A. Gabriel, M. Ptacek, and M. Gandorfer, “Farmers’ perspectives on field crop mpag.2021.106176. robots – Evidence from Bavaria, Germany,” *Comput. Electron. Agric.*, vol. 186, p. 106176, 2021, doi: 10.1016/j.co

[3] L. Zhou and S. Bai, “A new approach to design of a lightweight anthropomorphic arm for service applications,” *J. Mech. Robot.*, vol. 7, no. 3, pp. 1–12, 2015, doi: 10.1115/1.4028292.

[4] Y. Xiao, Z. Fan, W. Li, S. Chen, L. Zhao, and H. Xie, “A Manipulator Design Optimization Based on Constrained Multi-objective Evolutionary Algorithms,” *Proc. - 2016 Int. Conf. Ind. Informatics - Comput. Technol. Intell. Technol. Ind. Inf. Integr. ICIICII 2016*, pp. 199–205, 2017, doi: 10.1109/ICIICII.2016.0056.

[5] A. M. A. Alrashdi, A. D. Al-Qurashi, M. A. Awad, S. A. Mohamed, and A. A. Al-rashdi, “Quality, antioxidant compounds, antioxidant capacity and enzymes activity of ‘El-Bayadi’ table grapes at harvest as affected by preharvest salicylic acid and gibberellic acid spray,” *Sci. Hortic. (Amsterdam).*, vol. 220, no. December 2016, pp. 243–249, 2017, doi: 10.1016/j.scienta.2017.04.005.

[6] R. Berenstein, O. Ben Shahar, A. Shapiro, and Y. Edan, “Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer,” *Intell. Serv. Robot.*, vol. 3, no. 4, pp. 233–243, 2010, doi: 10.1007/s11370-010-0078-z.

[7] T. Yoshikawa, “Manipulability of Robotic Mechanisms.,” pp. 439–446, 1985.

[8] Z. Huang, F. Li, and L. Xu, “Modeling and simulation of 6 DOF robotic arm based on gazebo,” *2020 6th Int. Conf. Control. Autom. Robot. ICCAR 2020*, vol. 2020-April, pp. 319–323, 2020, doi: 10.1109/ICCAR49639.2020.9107989.

[9] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *Micro Machine and Human Science, 1995. MHS ’95., Proceedings of the Sixth International Symposium on*. pp. 39–43, 1995.

[10] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” *Soft Computing*, vol. 22, no. 2. pp. 387–408, 2018, doi: 10.1007/s00500-016-2474-6.

[11] R. Shwartz-Ziv and A. Armon, “Tabular data: Deep learning is not all you need,” *Inf. Fusion*, vol. 81, no. September 2021, pp. 84–90, 2022, doi: 10.1016/j.inffus.2021.11.011.