POLITECNICO DI TORINO
Repository ISTITUZIONALE

Non-linear Model Predictive Control for Multi-task GPS-free Autonomous Navigation in Vineyards

Original
Non-linear Model Predictive Control for Multi-task GPS-free Autonomous Navigation in Vineyards / Sperti, M., Ambrosio, M., Martini, M., Navone, A., Ostuni, A., Chiaberge, M.. - (2024), pp. 96-101. (2024 IEEE 20th International Conference on Automation Science and Engineering (CASE) Bari (Italia) 28 August 2024 - 01 September 2024) [10.1109/CASE59546.2024.10711737].

Availability:
This version is available at: 11583/2994394 since: 2024-11-25T14:02:48Z

Publisher:
IEEE

Published
DOI:10.1109/CASE59546.2024.10711737

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

25 November 2026

Non-linear Model Predictive Control for Multi-task GPS-free Autonomous Navigation in Vineyards

Matteo Sperti, Marco Ambrosio[1], Mauro Martini[1], Alessandro Navone[1], Andrea Ostuni[1], and Marcello Chiaberge[1]

***Abstract*— Autonomous navigation is the foundation of agricultural robots. This paper focuses on developing an advanced autonomous navigation system for a rover operating within row-based crops. A position-agnostic system is proposed to address the challenging situation when standard localization methods, like GPS, fail due to unfavorable weather or obstructed signals. This breakthrough is especially vital in densely vegetated regions, including areas covered by thick tree canopies or pergola vineyards. This work proposed a novel system that leverages a single RGB-D camera and a Non-linear Model Predictive Control strategy to navigate through entire rows, adapting to various crop spacing. The presented solution demonstrates versatility in handling diverse crop densities, environmental factors, and multiple navigation tasks to support agricultural activities at an extremely cost-effective implementation. Experimental validation in simulated and real vineyards underscores the system's robustness and competitiveness in both standard row traversal and target objects approach.**

I. INTRODUCTION

In recent years, precision agriculture has advanced significantly, utilizing technology to optimize crop production and reduce waste [1]. Row-based crops, in particular, represent a pivotal scenario in precision agriculture applications. Research in this domain encompasses various aspects, such as plant health monitoring [2], harvesting [3], spraying [4], irrigation [5], and seeding [6].

This work contributes to the foundation problem of robust autonomous platforms in row-based crops [7], [8], to address all the aforementioned tasks. Standard localization technologies as the Global Navigation Satellite System (GNSS), can fail in this context due to adverse weather or dense vegetation [9]. Moreover, GPS-based solutions are often enhanced by the corrections carried out by multiple costly Real-Time Kinematics (RTK) receivers.

Alternative methods, such as Visual Odometry (VO), have been investigated to localize rovers using camera image streams [10]. However, challenges arise in row-crop fields due to the repetitiveness of environmental visual patterns. A versatile position-agnostic system is therefore proposed, excelling in scenarios where traditional methods fall short. The presented control system can be inserted in a framework to address multiple navigation tasks, such as traversing entire rows, avoiding obstacles effectively, and approaching target objects in varying row spacing. Position-agnostic sensorimotor agents directly map sensor data to rover velocity commands without relying on fixed Reference Frames (RFs).

[1]Department of Electronics and Telecommunications, Politecnico di Torino, 10129, Torino, Italy. {firstname.lastname}@polito.it



(a) Straight vineyard (b) Pergola vineyard

Fig. 1. Vineyards used for testing the proposed navigation system in Agliè, Turin, Italy.

For instance, [11], [12], [13] proposed to segment the input image to compute a set point in the camera frame and to use a proportional controller to align the rover towards the set point. These methods, however, fail in the case of pergola vineyards or high trees in which the sky is not visible [11], [13] or the crops are not uniform on both sides [12], see Fig. 1. Moreover, segmentation-based methods encounter difficulties in generalizing due to visual seasonal changes and in handling unexpected obstacles along the path.

Decision algorithms provide another avenue, with Deep Reinforcement Learning agents trained by [14] for decision-making or Convolutional Neural Network (CNN) used by [15] to output actions from a discrete set. Additionally, [16] introduced a path-following Non-linear Model Predictive Control (NMPC) approach, leveraging a Point Cloud Data (PCD) from four cameras to generate the reference path.

The primary contribution of this research lies in developing a new robust controller tailored for row crop geometry, avoiding the need for precise and costly localization systems such as GPS receivers. Notably, a single RGB-D camera represents a cost-effective option compared to other sensors like 3D LIDARs.

Furthermore, the navigation system has been conceived to support task-oriented behavior. This enables the system not only to navigate the agricultural space efficiently but also to engage in auxiliary tasks. This flexibility enhances the system's overall utility and broadens its applicability

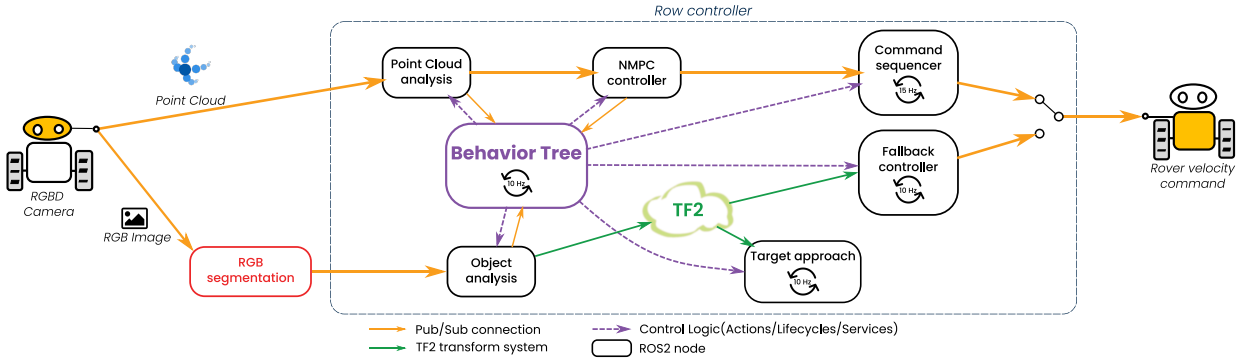

Fig. 2. Computation Graph of the ROS 2 overall application system. A Behavior Tree manages and coordinates the NMPC controller for row traversal, target object approach, and recovery behaviors for robust multi-task navigation.

for diverse agricultural tasks beyond navigation, from object transport to plant harvesting.

The next sections are organized as follows: Section II describes the proposed control system for multi-task position-agnostic autonomous navigation in row-based crops. Section III illustrates the experiments conducted both in simulated and real vineyards, discussing the obtained results. Finally, Section IV wraps up all the considerations on the study and suggests future directions.

II. METHODOLOGY

The proposed methodology adopts a position-agnostic controller approach to guide the robot in row-based crops without relying on a localization signal. Costly GPS sensorization of the platform may lead to unreliable performances in case of thick vegetation. Taking a PCD as input, the controller computes in real-time linear velocity v_x and angular velocity ω_z.

A. Navigation System Architecture

The computation graph, shown in Fig. 2, illustrates the system's structure. The overall system is orchestrated by a *Behavior Tree*, overseeing high-level logic, mission switches, start and stop commands, failure detection, and initiating fallback procedures.

The RGB-D camera data is analyzed by two parallel operation flows that carry out standard row traversal and check the presence of potential objects of interest in the mission to be approached. The *Point Cloud Analysis* process analyzes the environment starting from the PCD captured by the camera. Then the *NMPC controller* uses a Non-linear Model Predictive Control strategy to determine the optimal control sequence. This sequence is computed considering the geometrical constraints of the row and the obstacles identified by the *Point Cloud Analysis* process. Concurrently, a generic *Object Detection* visual algorithm could be adopted to estimate the position of potential target objects from the camera image. If any, the *Target approach* process is triggered to smoothly guide the robot to a desired position near the target. A *Fallback Controller* manages recovery from fault behaviors: a simple proportional controller is used to re-align the rover with the plants' row.

B. Point Cloud processing pipeline

The PCD of the camera is processed to perceive potential obstacles and the boundaries of the crop row. The output of this pipeline includes the obstacle points and two straight lines, which represent the geometrical sides of the row.

The first part of the procedure consists of mapping the input PCD to the 2D horizontal plane. Hence, as a first thing, the PCD is transformed into the rover RF. A down-sampling, performing a voxelization operation with a resolution r_v, and filtering using a classical k-NN algorithm to exclude the noise points, are applied. Then, the PCD is cropped to eliminate outliers and misleading points in the sky and on the ground. Therefore, minimum and maximum height thresholds, $z_{th,min}$, $z_{th,max}$ on the z-*axis* are set to ensure the removal of ground points. This operation is necessary in cases where the rover is not perfectly parallel to the ground plane due to bumpy or rough terrain. If, after this preprocessing, the fraction of remaining points falls below a specified threshold, f_{points}, relative to the original point count, the field of view is considered empty, indicating no row detection. On the other hand, if the fraction exceeds the threshold, the points are projected onto a 2D plane by considering only their coordinates on the x-y plane generating a grid map. After the generation of the obstacle occupation map, the areas behind them are also considered occupied. This allows us to identify the inner edge of the plants in the row. Then, a heuristic approach is used to gather the occupied zones on the available borders.

Since the two internal row borders are considered two straight lines, a least square fit is applied to evaluate the angular and bias coefficients, $a_i, b_i \in \mathbb{R}$ of the equation $y = a_i x + b_i, i \in [l, r]$. Two lines are generated, one for the left side l and one for the right one r.

Finally, a safety distance margin R is added to the row's two borders to consider the robot's occupancy and account for possible errors. Moreover, suppose the rover is required to travel only in half of the available row space, for example in a scenario where multiple robots are expected to move in opposite directions. In that case, the middle line is computed and used to separate the two motion lanes. An error is raised if one of the two lines is (given a predefined maximum angle) perpendicular to the $x-$axis, i.e., the direction of

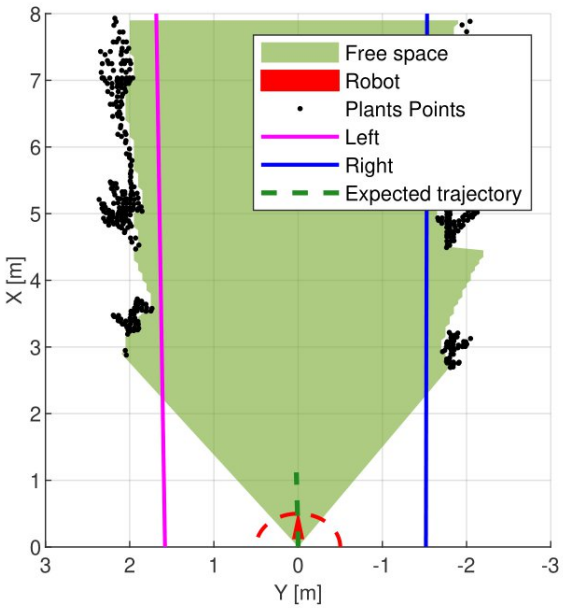

Fig. 3. The black points represent the input PCD, filtered and flattened on a 2D map (obstacles), the green area is the free space in front of the rover, while the two straight lanes represent the lane borders. Finally, the dotted green line is the expected trajectory as computed by the NMPC controller.

motion of the rover. Hence, the fallback recovery procedure is initiated to prevent the rover crash and realign it with the row direction.

C. NMPC formulation

A customized model and cost function were tailored to address the specific requirements and characteristics of the rover's navigation scenario. This involved carefully calibrating the model parameters and formulating the cost function terms, as well as the problem constraints. The inputs of the NMPC controller are the obstacle points and the two first-order polynomials representing the two straight lines delimiting the plants, expressed in the robot's RF, as described in Section II-B.

The NMPC approach requires a plant model to predict future states. For this purpose, a modified version of the Unicycle Model was selected. This decision is based on the differential drive mechanism of the testing rovers, which the unicycle model can sufficiently approximate. Summarizing, the kinematic model of the unicycle has been modified to:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} v(x_3^2 - x_4^2) \\ v(2x_3x_4) \\ -\omega\frac{x_4}{2} \\ \omega\frac{x_3}{2} \end{bmatrix} \tag{1}$$

where $x_1 = x$, $x_2 = y$, $x_3 = \cos\frac{\theta}{2}$, $x_4 = \sin\frac{\theta}{2}$.

Moreover, input saturation constraints were incorporated into the NMPC minimization problem, allowing for the specification of maximum linear and angular velocities, namely $v_{x,max}$ and $\omega_{z,max}$, as parameters before the system's initiation.

In addition, non-linear constraints were integrated to ensure obstacle avoidance, according to the following formula:

$$-(x_1 - o_1^i)^2 - (x_2 - o_2^i)^2 + R^2 \leq 0 \tag{2}$$

where the two negative terms represent the square of the Euclidean distance between the rover pose $\mathbf{x}$ and the i-th obstacle $\mathbf{o}^i$, and the parameter R represents a predetermined safe distance between the rover and an obstacle point. This constraint must hold for each time step $t_k = 1, \ldots, T_H$ and for every obstacle point, providing a robust mechanism for obstacle avoidance throughout the prediction horizon.

The core of the NMPC formulation lies in defining an objective function, which needs to be optimized, represented as follows:

$$C = \sum_{k=0}^{n-1} (\underbrace{l(\mathbf{x}_k, \mathbf{u}_k, p)}_{\text{Lagrange term}} + \underbrace{\Delta\mathbf{u}_k^T \mathbf{R} \Delta\mathbf{u}_k}_{\text{r-term}}) + \underbrace{m(\mathbf{x}_n)}_{\text{meyer term}} \tag{3}$$

In this equation, three contributions can be identified, respectively the *Lagrange* term, the *meyer* term, and the *r*-term.

The *Lagrange* term, $l(\mathbf{x}_k, \mathbf{u}_k, p)$, evaluated and summed at each time step until the prediction horizon, is composed of two contributions as in the following equation:

$$l(\mathbf{x}_k, \mathbf{u}_k, p) = K_{lane} C_{lane}(\mathbf{x}_k, \mathbf{u}_k, p) + K_{orient} C_{align}(\mathbf{x}_k, \mathbf{u}_k, p) \tag{4}$$

The first term, $C_{lane}(\mathbf{x}_k, \mathbf{u}_k, p)$, aims at maintaining a central trajectory with respect to the lane while, the second term, $C_{align}(\mathbf{x}_k, \mathbf{u}_k, p)$, aims at minimizing misalignment from the row direction. The constants K_{lane} and K_{orient} are the weights of the corresponding contributions.

Given a position $\mathbf{x} = [x_1, x_2, x_3, x_4]$, and the two lines delimiting the row $y_l = a_l x_1 + b_l$ (on the left), and $y_r = a_r x_1 + b_r$ (on the right), the cost term regarding the lane centrality is described by the following equation:

$$C_{lane} = \frac{4}{(y_l - y_r)^2} x_2^2 - 4\frac{(y_l + y_r)}{(y_l - y_r)^2} x_2 + \frac{(y_l + y_r)^2}{(y_l - y_r)^2} \tag{5}$$

Essentially, it consists of a paraboloid with its minimum coinciding with the middle of the row. For each depth value x_1 a convex-upward parabola is constructed along the axis x_2 with a minimum in the middle of the lane. Therefore, the minimum cost trajectory ideally aligns perfectly with it.

The cost term for the alignment is computed considering the difference between the angular coefficient of the middle line $a_{avg} = (a_l + a_r)/2$ and the angular coefficient of a straight line oriented as the rover a_{rover} as in the following equation:

$$a_{rover} = \tan\theta = \frac{\sin\theta}{\cos\theta} = \frac{2x_3x_4}{x_3^2 - x_4^2} \tag{6}$$

$$C_{align} = (a_{avg} - a_{rover})^2 \tag{7}$$

The *r*-term is the quadratic penalty on changes for control inputs, which can be utilized to smooth the obtained optimal solution and serve as a crucial tuning parameter.

The terminal (or *meyer*) term of the objective function is designed to maximize the distance traveled by the rover in the

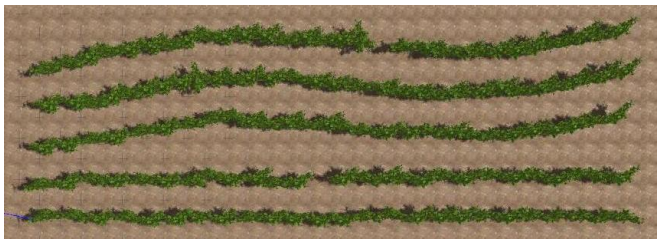

Fig. 4. Aerial view of vineyards in Gazebo used for testing in simulation.

prediction horizon time interval. So, recalling that $\max f = \min -f$, the terminal (or *meyer*) term is set as follows:

$$m(\mathbf{x}) = -K_{travel} \frac{x_1 + a_{avg} \cdot x_2}{\sqrt{1 + a_{avg}^2}} \tag{8}$$

here K_{travel} represents the parameter for weighting this term, $a_{avg} = (a_l + a_r)/2$ is the angular coefficient of the line in the middle of the row, and x_1, x_2 are the coordinates of the rover in plane at the horizon $t_k = T_H$. The distance traveled by the rover is projected onto the middle line to weigh only the distance traveled in the direction of the row.

The result of this optimization problem is the sequence of velocities $[v_x, \omega_z]_k$ for $t_k = 1, \ldots, T_H$, which minimizes the cost function. The first pair is used to move the rover immediately, while the entire sequence is used to initialize the optimization at the next time step, thus enhancing the startup efficiency of the optimization process.

III. TESTS AND RESULTS

Extensive experiments were conducted on both simulated and real vineyards to illustrate the proposed solution's competitive advantages for testing and validation.

A. Experimental Setting

All the code was developed in a ROS 2 framework and has been tested on Ubuntu 22.04 LTS using the ROS 2 Humble distro. This research employed two distinct mobile robots: the Clearpath Robotics Jackal and Husky[1]. For simulated tests, the Gazebo platform, the Jackal model and description, and the PIC4rl_gym [17] evaluation tool were utilized. The world chosen, shown in Fig. 4, contains a straight and curved vineyard, with an intra-row space of around 1.5 m.

Instead, tests in a real vineyard utilized the Husky to evaluate the path metrics, an Intel Realsense D455 RGB-D camera, and a Velodyne VLP16 3D LIDAR for comparison. The tests were conducted on a straight vineyard with an intra-row space of around 2.5 m and on a pergola vineyard with an intra-row space of around 4 m, both shown in Fig. 1.

An accurate robot localization in the row was necessary for comparing its position to a ground truth path. However, the odometry system of the IMU of the rover failed to localize the rover due to significant drifts; SLAM techniques based on scan matching algorithms such as KISS-ICP [18], also failed to correctly localize the system due to the repetitiveness of the environment. So, the GPS position provided by the SwiftNav Duro GNSS receiver was used as a reference to

[1]https://clearpathrobotics.com/

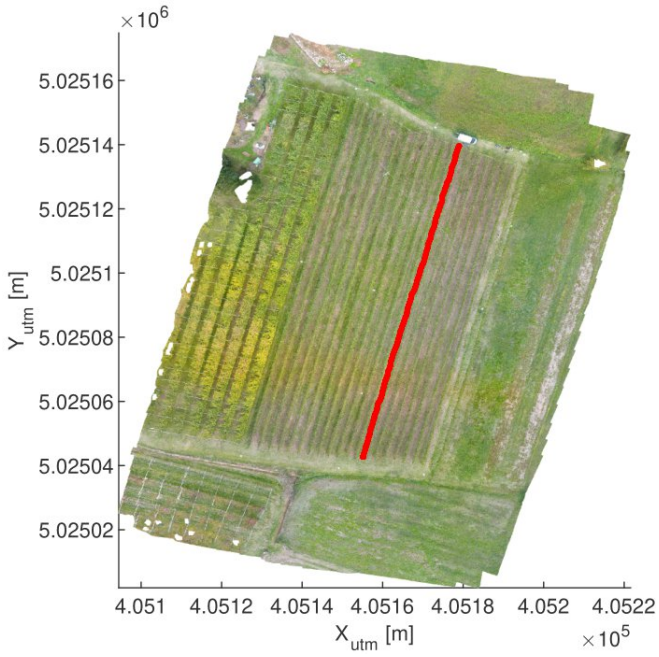

Fig. 5. Satellite view of the vineyard. In red the trajectory followed by the Husky rover during a test session.

compute the metrics, along with a precise geo-localization of the row in the vineyards (Fig. 5). However, GPS positioning is prone to errors in environments where leaves obstruct GPS visibility, leading to signal failures and inaccuracies in position tracking. Moreover, costs must also be considered: the GNSS receiver chosen to obtain a sufficiently precise localization is much more expensive than an RGB-D camera. These facts highlight the difficulties in localizing a ground rover in this environment and suggest the advantages of adopting a position-agnostic controller such as the one developed in this project. RGB-D cameras are also a cost-effective choice to get a limited FOV PCD, compared to a multi-range 3D LIDAR.

To implement the NMPC controller, the DO-MPC library [19] was chosen for its versatility. The hyper-parameters of the NMPC controller have been set by a trial and error procedure. For the tests, $v_{x,max} = 0.4\ m/s$ or $v_{x,max} = 0.5\ m/s$ and $\omega_{z,max} = 0.5\ rad/s$ has been set. The control period has been fixed to 0.7 s. In the PCD processing pipeline, the resolution of the voxel has been set to $r_v = 0.05\ m$, the minimum and maximum height threshold have been set to $z_{th,min} = 0.15\ m$ and $z_{th,max} = 2\ m$ and the minimum point threshold has been set to $f_{points} = 0.2$. The safety margins for the Jackal and the Husky robots were respectively set to $R_{Jackal} = 0.3\ m$ and $R_{Husky} = 0.4\ m$.

B. Evaluation Metrics

The metrics used to evaluate the performances of the control system include:

- *Clearance Time* [s] and *Mean linear velocity* v_{avg} [m/s]: gauging the effectiveness of the proposed solution.
- *Cumulative heading average Cum.*γ_{avg} or *standard deviation of the heading* γ_{std} [rad], and the *standard*

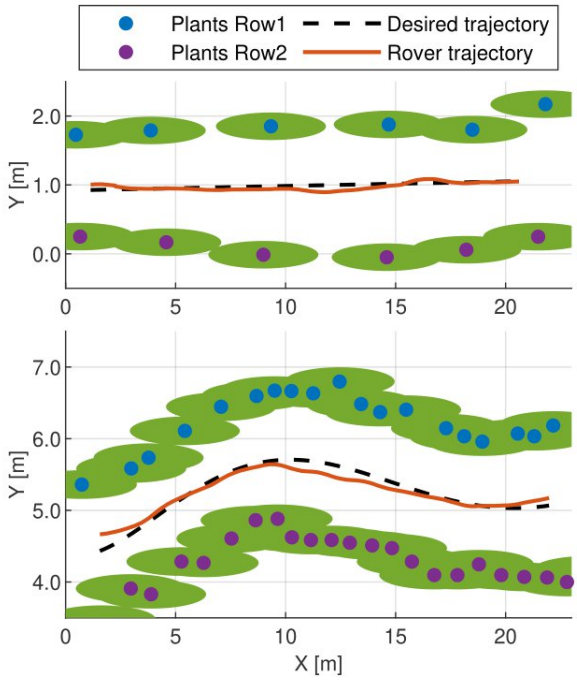

Fig. 6. Tests in a simulated vineyard using the PCD of the camera as input in two different scenarios.

deviation of the angular velocity ω_{std} [rad/s]: measuring the oscillation around the trajectory.

- *Trajectory Mean Absolute Error (MAE)* [m] and *trajectory Mean Squared Error (MSE)* [m^2]: measuring the error of the rover trajectory concerning a predefined ground truth.

C. Tests in simulated environment

The extensive simulations conducted in simulated vineyard environments have demonstrated the reliability and robustness of the proposed navigation system. As illustrated in Fig. 6, the rover's trajectory closely aligns with the desired central path, exhibiting minimal oscillations in both straight and curved vineyards.

Detailed results are provided in Tab. I, revealing several key performance indicators. In both straight and curved vineyards, the rover consistently achieves speeds close to the maximum limit ($v_{avg} \simeq 0.39$ m/s for $v_{x,max} = 0.4$ m/s), resulting in effective clearance times. The rover's trajectory shows minimal oscillations, as indicated by a small standard deviation of angular velocity ($\omega_{std} \simeq 0.05$ rad/s), reflecting stable and smooth behavior. Path metrics, including MAE and MSE, are minimal, on the order of centimeters. This demonstrates the rover's precise adherence to the center of its lane. In the curved vineyard, a slightly larger path error is observed (MAE up to 0.2 m in the worst case), attributed to the controller's inclination to cut curves. This behavior can be mitigated through parameter tuning. The algorithm's consistent performance across input sensors, including RGB-D cameras, highlights its reliability and versatility. This robustness, even compared to more expensive technologies such as LIDAR, underscores the algorithm's adaptability to various sensor configurations. The ability to achieve comparable results with RGB-D cameras suggests a cost-effective alternative for applications where LIDAR may be cost-prohibitive. Overall, these findings underscore the effectiveness and versatility of the proposed navigation system across diverse vineyard scenarios.

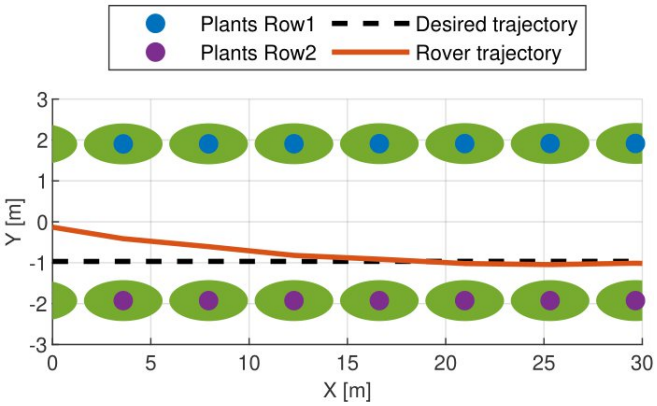

Fig. 7. Test in a real pergola vineyard using the PCD of the camera as input. The desired position is in the middle of the right lane (so at 3/4 of the entire intra-row space). The rover starts in the middle of the row and then converges smoothly to the desired position.

D. Tests in real scenario

The real-world tests conducted in vineyards have validated the results obtained in the simulated environment. Detailed results are presented in Tab. II, highlighting the robust performance of the controller in real scenarios. As in simulation, the rover consistently achieves speeds close to the maximum limit ($v_{avg} \simeq 0.399\ m/s$ for $v_{x,max} = 0.4\ m/s$ and $v_{avg} \simeq 0.49\ m/s$ for $v_{x,max} = 0.5\ m/s$). The achieved trajectory shows minimal oscillations, as indicated by a small standard deviation of angular velocity ($\omega_{std} \simeq 0.05\ rad/s$), reflecting stable and smooth behavior. The exception is the narrow straight vineyard in the right lane configuration, where this metric is slightly larger ($\omega_{std} \simeq 0.18\ rad/s$): the rover displays a more oscillatory behavior, likely due to the proximity of the right lane to the crops. This behavior is less prominent in the pergola vineyard test (Fig. 7) with a larger intra-row distance (4 m), where the rover shows a smooth convergence to the right lane without significant oscillations. Path metrics, including MAE and MSE, are minimal, on the order of centimeters (up to 20 cm for the narrow vineyard and up to 30 cm for the larger pergola vineyard). However, it's important to consider the error in the reference trajectory, as well as the error of localization, affected by the intrinsic accuracy of the sensor used when interpreting these results.

To prove that our solution provides a valid alternative to localization-based navigation algorithms, KISS-ICP [18], a state-of-the-art SLAM technique based on scan matching, has been tested on a set of trajectories collected in the real-world vineyard. It was demonstrated that the localization provided by KISS-ICP is not reliable for a standard control algorithm. In fact, in two tests of straight vineyards, it failed after 13.6 m in the first case and after 2.4 m in the second. In the pergola vineyard, it fails after 20.7 m in one case but succeeds in localizing the robot until the end of the row.

TABLE I

RESULTS OF CONDUCTED EXPERIMENTS IN SIMULATED STRAIGHT AND CURVED VINEYARDS.

Field	Sensor	Clearance time [s]	Cum. γ_{avg} [rad]	v_{avg} [m/s]	ω_{std} [rad/s]	MAE [m]	MSE [m^2]
Straight	LIDAR	49.528±0.167	0.036±0.001	**0.395±0.002**	**0.034±0.001**	**0.034±0.001**	**0.001±0.000**
	PCD cam	52.586±4.130	0.045±0.001	0.377±0.019	0.038±0.001	0.048±0.005	0.003±0.001
	RGB-D cam	**49.321±0.356**	**0.011±0.005**	0.395±0.001	0.046±0.004	0.104±0.011	0.018±0.004
Curved	LIDAR	52.080±0.220	-0.024±0.001	**0.397±0.001**	**0.036±0.001**	0.102±0.001	0.015±0.000
	PCD cam	52.157±0.673	**0.002±0.002**	0.393±0.002	0.041±0.003	**0.068±0.004**	**0.007±0.001**
	RGB-D cam	**51.763±0.228**	-0.011±0.002	0.394±0.001	0.056±0.007	0.188±0.005	0.051±0.003

TABLE II

RESULTS OF A SERIES OF EXPERIMENTS IN REAL VINEYARDS.

Field	Sensor	Position	$v_{x,max}$ [m/s]	γ_{std} [rad]	v_{avg} [m/s]	ω_{std} [rad/s]	MAE [m]	MSE [m^2]
Straight	PCD camera	Centered	0.4	0.031±0.007	0.399±0.000	0.042±0.002	0.165±0.007	0.035±0.000
		Right lane	0.5	0.388±0.395	0.488±0.007	0.184±0.108	0.204±0.098	0.070±0.044
	LIDAR	Right lane	0.5	0.0153	0.4989	0.0271	0.1519	0.0294
Pergola	PCD camera	Centered	0.4	0.122	0.399	0.063	0.313	0.129
		Right lane	0.4	0.047	0.399	0.04	0.092	0.011

IV. CONCLUSIONS

The position-agnostic NMPC controller proposed in this paper has demonstrated robustness in effectively handling the diverse challenges presented in traversing row-based fields with different characteristics without accessing any localization information. Its resilient navigation on rough terrains underscores its adaptability to real-world agricultural conditions with a lower platform cost. This research significantly contributes to the continuous advancement of precision agriculture and the evolution of autonomous navigation systems tailored for row-based crop environments.

ACKNOWLEDGMENT

This work has been developed with the contribution of Politecnico di Torino Interdepartmental Centre for Service Robotics PIC4SeR[2].

REFERENCES

[1] Z. Zhai, J. F. Martínez, V. Beltran, and N. L. Martínez, "Decision support systems for agriculture 4.0: Survey and challenges," *Computers and Electronics in Agriculture*, vol. 170, p. 105256, 2020.

[2] N. Virlet, K. Sabermanesh, P. Sadeghi-Tehran, and M. J. Hawkesford, "Field scanalyzer: an automated robotic field phenotyping platform for detailed crop monitoring," *Functional Plant Biology*, vol. 44, no. 1, pp. 143–153, 2017.

[3] W. Jia, Y. Zhang, J. Lian, Y. Zheng, D. Zhao, and C. Li, "Apple harvesting robot under information technology: A review," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, p. 1729881420925310, 2020.

[4] R. Berenstein, O. B. Shahar, A. Shapiro, and Y. Edan, "Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer," *Intelligent Service Robotics*, vol. 3, no. 4, pp. 233–243, 2010.

[5] D. Kohanbash, A. Valada, and G. Kantor, "Irrigation control methods for wireless sensor network," in *2012 Dallas, Texas, July 29-August 1, 2012*. American Society of Agricultural and Biological Engineers, 2012, p. 1.

[6] J. Katupitiya, R. Eaton, and T. Yaqub, "Systems engineering approach to agricultural automation: new developments," in *2007 1st Annual IEEE Systems Conference*. IEEE, 2007, pp. 1–7.

[7] S. Cerrato, V. Mazzia, F. Salvetti, and M. Chiaberge, "A deep learning driven algorithmic pipeline for autonomous navigation in row-based crops," *arXiv preprint arXiv:2112.03816*, 2021.

[8] F. Salvetti, S. Angarano, M. Martini, S. Cerrato, and M. Chiaberge, "Waypoint generation in row-based crops with deep learning and contrastive clustering," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2022, pp. 203–218.

[9] J. Zheng, Y. Wang, and N. Nihan, "Quantitative evaluation of gps performance under forest canopies," in *Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.*, 2005, pp. 777–782.

[10] S. Zaman, L. Comba, A. Biglia, D. R. Aimonino, P. Barge, and P. Gay, "Cost-effective visual odometry system for vehicle motion control in agricultural environments," *Computers and Electronics in Agriculture*, vol. 162, pp. 82–94, 2019.

[11] D. Aghi, S. Cerrato, V. Mazzia, and M. Chiaberge, "Deep Semantic Segmentation at the Edge for Autonomous Navigation in Vineyard Rows," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sept. 2021.

[12] A. Navone, M. Martini, A. Ostuni, S. Angarano, and M. Chiaberge, "Autonomous navigation in rows of trees and high crops with deep semantic segmentation," in *2023 European Conference on Mobile Robots (ECMR)*, 2023, pp. 1–6.

[13] J. Radcliffe, J. Cox, and D. M. Bulanon, "Machine vision for orchard navigation," *Computers in Industry*, vol. 98, pp. 165–171, 2018.

[14] M. Martini, S. Cerrato, F. Salvetti, S. Angarano, and M. Chiaberge, "Position- Agnostic Autonomous Navigation in Vineyards with Deep Reinforcement Learning," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 08 2022.

[15] P. Huang, L. Zhu, Z. Zhang, and C. Yang, "An End-to-End Learning-Based Row- Following System for an Agricultural Robot in Structured Apple Orchards," *Mathe- matical Problems in Engineering*, vol. 2021, pp. 1–14, 09 2021.

[16] A. Villemazet, A. Durand-Petiteville, and V. Cadenat, "Multi-Camera GPS-Free Nonlinear Model Predictive Control Strategy to Traverse Orchards," in *2023 European Conference on Mobile Robots (ECMR)*. IEEE, 2023, pp. 1–7.

[17] M. Martini, A. Eirale, S. Cerrato, and M. Chiaberge, "Pic4rl-gym: a ros2 modular framework for robots autonomous navigation with deep reinforcement learning," in *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, 2023, pp. 198–202.

[18] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[19] F. Fiedler, B. Karg, L. Lüken, D. Brandner, M. Heinlein, F. Brabender, and S. Lucia, "do-mpc: Towards FAIR nonlinear and robust model predictive control," *Control Engineering Practice*, vol. 140, p. 105676, 2023.

[2]www.pic4ser.polito.it