agriculture

MDPI

Article

Task Space Model Predictive Control for Vineyard Spraying with a Mobile Manipulator

Ivo Vatavuk *, Goran Vasiljević and Zdenko Kovačić

Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia; goran.vasiljevic@fer.hr (G.V.); zdenko.kovacic@fer.hr (Z.K.)
* Correspondence: ivo.vatavuk@fer.hr

Abstract: In this paper, a Model Predictive Control (MPC)-based approach for vineyard spraying is presented, able to adapt to different vine row structures and suitable for real-time applications. In the presented approach, the mobile base moves along a row of vines while the robotic arm controls the position and orientation of the spray nozzle. A reference lawnmower pattern trajectory is generated from the vine canopy description, with the aim of minimizing waste while ensuring vine coverage. MPC is used to compute the trajectory of the vehicle along the row and the manipulator tool trajectory, which follow the spray reference, while minimizing vehicle acceleration and tool displacement. The manipulator tool velocity commands provided by the MPC algorithm are tracked using task space control. The presented approach is evaluated in two experiments: a vineyard spraying scenario and an external evaluation scenario in an indoor environment equipped with the Optitrack camera system.

Keywords: mobile manipulation; optimization and optimal control; agricultural robotics; viticulture

check for updates

Citation: Vatavuk, I.; Vasiljević, G.; Kovačić, Z. Task Space Model Predictive Control for Vineyard Spraying with a Mobile Manipulator. *Agriculture* **2022**, *12*, 381. https://doi.org/10.3390/agriculture12030381

Academic Editors: Jin Yuan, Wei Ji and Qingchun Feng

Received: 8 February 2022
Accepted: 7 March 2022
Published: 9 March 2022

1. Introduction

Agricultural robotics is an exciting, emerging research field that offers a potential solution to the problem of increasing global demand for food production due to exponential population growth and labor shortages [1]. Existing automated agricultural technologies use large and expensive machines that are strictly tied to a specific production process and usually have a significant environmental impact, especially ones used for pesticide, insecticide and herbicide application [2].

The research presented in this paper is a part of the HEKTOR project [3,4], which aims to replace such machines with heterogeneous autonomous robotic systems, capable of jointly performing different tasks in viticulture and mariculture. As a part of the project, a custom flipper-tracked mobile base was developed and equipped with a torque-controlled 7-DoF Kinova Gen3 robotic arm (Figure 1). For the task of vineyard spraying, a spray nozzle is attached to the manipulator end-effector, and its position and orientation are controlled by the robotic arm (Figure 1). This allows for precise control of the spray area, and is suitable for treating the entire vine canopy or specific areas of the plant. The HEKTOR project addresses viticulture activities that need to be carried out on steep terrain, typical of the Mediterranean islands. However, the method presented in this paper is general and was not developed specifically for steep terrains. The design of the mobile robot and the high torques of the track drives allow the storage and transport of a sufficient quantity of protective agent for spraying in conditions that are otherwise difficult for human workers.

This paper attempts to present a solution to the following problem: given a description of a row of grapevines, one must select coordinated mobile vehicle and robot arm commands that result in satisfactory canopy coverage, while aiming to minimize spraying agent waste, and perform the task as quickly as possible. This kind of problem setup calls for a control method that is able to adapt to different row structures, accelerate in areas of the row without grapevines and slow down in areas with the largest foliage heights.

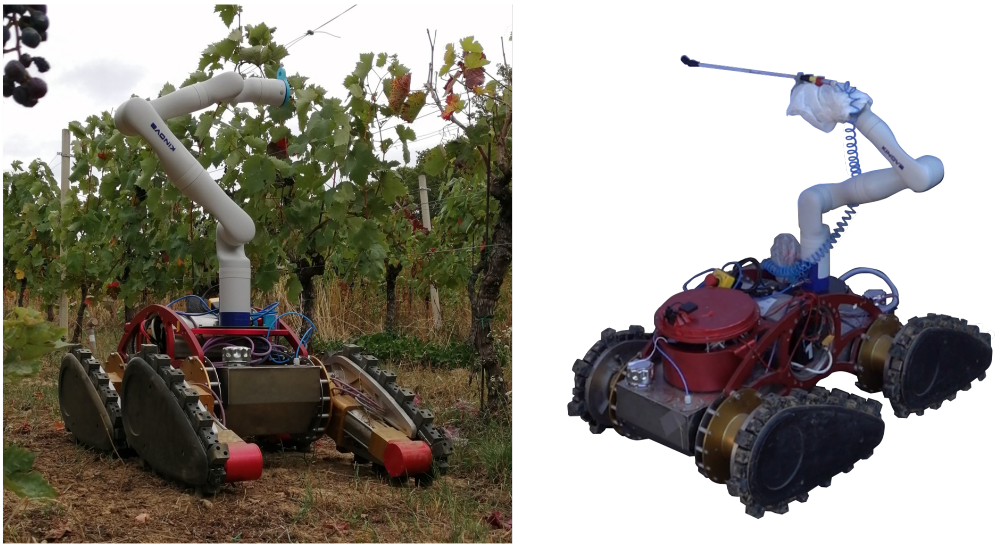

Figure 1. Mobile manipulator developed for the HEKTOR project, with an emphasis on manipulation ability and maneuverability in steep terrain.

1.1. Related Work

Research efforts have already been directed towards the development of robots for vineyard-related tasks [5–22]. The same applies to robots in agricultural spraying tasks [2,6–9,11,13,23]. In [21], Bouloumpasi et al. discuss the possibilities and limitations of autonomous robot technology for performing different tasks in viticulture. Vrochnidou et al. [22] present the system architecture of an autonomous robot for grape harvesting, and discuss the vision system used for the task of grape harvesting.

Monta et al. [5] designed a multipurpose robot for viticulture applications as early as 1995. Approaches for robotic harvesting, berry thinning, spraying and bagging are presented in the paper. At a conceptual level, the authors' approach to vineyard spraying is similar to the one presented in this paper, in which a spray nozzle is mounted on a robotic manipulator that follows a lawnmower pattern path. Monta et al. conclude that this kind of approach results in uniform spraying of the fixed-height foliage, but they do not discuss in detail the control algorithm used to achieve this. In this paper, an expansion of the following idea is given, able to handle different foliage shapes, with controls that allow for spraying where the mobile manipulator is able to adapt to a specific vine row description. Oberti et al. [7,8] mounted a precision spraying end-effector on a robotic manipulator and used it to spray specific disease-affected areas of the vine. In [13], Cantelli et al. present a mobile vehicle developed for autonomous spraying in agriculture, with emphasis on vehicle navigation and mission planning. Berenstein et al. [6] report on grape cluster and foliage detection algorithms for autonomous selective vineyard spraying. The authors present a mobile vehicle with multiple spray nozzles mounted at different heights and use visual feedback to select which nozzles should be active.

Similar work has been done with mobile manipulators for paint spraying [24–26]. Here, a lawnmower path is often used as a reference for the position of the tool, an approach that was adopted in the currently presented work.

Model Predictive Control (MPC) has been a subject of a number of research applications in robotics, such as manipulation, autonomous vehicle control and legged robot control [27–31]. Wieber [27] presents the application of linear MPC to the bipedal walking robot, and discusses the advantages of this kind of controller for this particular task. In [29], Elsisi presents an optimally designed nonlinear model predictive controller, and its application to the robot manipulator. The parameters for the MPC algorithm presented by Elsisi are selected by a modified multitracker optimization algorithm, rather than by trial and error. In later work, Elsisi et al. focus on autonomous vehicle control [30,31]. In [30], an MPC algorithm with a small number of parameters is presented, which is optimally designed using the social ski diver algorithm. In [31], an adaptive MPC algorithm is used for autonomous vehicle control, utilizing a modified grey wolf optimizer, based on opposition-based learning and quasi opposition-based learning. In this paper, MPC parameters are selected experimentally, while the more sophisticated MPC tuning algorithms will be considered as part of future work.

The use of different trajectory optimization algorithms in mobile manipulation is also a well-researched topic [28,32–35]. These trajectory planning algorithms are often designed to be as general as possible, capable of performing multiple tasks while considering obstacles, mobile manipulator dynamics and stability. In [28], Pankert et al. present a nonlinear MPC scheme for continuous mobile manipulation, and show its ability to perform several tasks. The strength of the authors' approach lies in the variety of tasks it can perform, involving position or force control, while avoiding obstacles. One of the experiments presented in the paper is spraying a pattern on a flat surface, which is similar to the presented vision of vineyard spraying with a mobile manipulator. However, the method presented in this paper simplifies the planning problem by using certain insights into this particular task. The complexity of the general mobile manipulation problem includes vehicle path planning while considering obstacles, and controlling the forces or positions of the robot arm end-effector. Grapevines are typically planted in structured rows, the area between the rows being obstacle free. During vineyard spraying, the mobile base is limited to movement between the rows. For this particular application, this paper proposes a task space MPC algorithm that considers only the motion of the mobile base along the row and the two-dimensional motion of the manipulator tool. The result is a low-dimensional linear MPC algorithm that can be solved in real time even with large prediction horizons. Additionally, in the spraying experiment presented in [28], the orientation of the spray nozzle is kept constant throughout the task execution. This greatly reduces the maximum and minimum heights that the spray area can reach, which also depends on the reach of the particular robot arm. Knowing the typical foliage heights and the reach of the Kinova Gen3 robotic arm, it was concluded that constant tool orientation was not appropriate for the vineyard spraying task.

1.2. Contribution

This paper presents a novel method for vineyard spraying with a mobile manipulator that combines the following methodological elements:

- Row-specific reference trajectory generation based on grapevine canopy description;
- Forward mobile base and two-dimensional task space manipulator command generation using linear reference tracking MPC;
- Manipulator joint space velocity command selection using task space control.

Compared to other state-of-the-art autonomous vineyard spraying methods, the presented method includes multiple novelties. The lawnmower spraying reference trajectory is generated based on a specific grapevine row description, aiming to minimize spraying waste and reduce the risk of excessive pollution. The control design based on task space model predictive control allows the spraying agent to follow this reference while optimizing coordinated mobile base and robot arm movements. This results in the desired behavior, where the mobile base keeps constantly moving, accelerating in areas of the row without grapevines and slowing down in areas with foliage of the largest heights. The predictive nature of MPC allows for such behavior, which would be difficult to achieve using standard, instantaneous control methods. Predictive control allows the controller to anticipate any changes in the reference trajectory and optimally select how the mobile base and the robot arm should react to it.

Compared to the state of the art in mobile manipulation, a control algorithm for the specific purpose of vineyard spraying is developed instead of using a general trajectory optimization algorithm. Unlike the general trajectory optimization algorithms, which usually include end-effector pose planning, the presented algorithm combines MPC and task space control in such a way that the MPC plans only the positions of the end-effector while disregarding its orientations, which are handled by an instantaneous task space control algorithm.

This kind of approach results in two quadratic programming problems, for MPC and task space control, which can be solved in real time even for large prediction horizons.

To summarize, our contributions are as follows:

- A novel method for vineyard spraying with mobile manipulators able to adapt to a specific grapevine row description;
- Reference trajectory generation based on grapevine row description;
- Control design based on computationally efficient task space trajectory tracking MPC that exploits the insight into the motion constraints imposed by the specific task of vineyard spraying.

2. Task Space Model Predictive Control Approach

Since the vines are usually planted in structured rows, it is assumed that the vehicle moves in a straight line at a constant distance from the vines, and, because of this, the spray reference for the MPC is generated in a two-dimensional space. As shown in Figure 2, three coordinate frames are defined: a ground-level global frame L_G, a mobile base frame L_B and the spray frame L_S. The spray frame L_S is defined at a fixed distance from the spray nozzle, and its position is computed using a single static transformation from the last link of the robot manipulator. The goal is to control the global position of the L_S frame, which depends on the position of the L_B frame with respect to L_G, and the pose of the robot arm. The y-coordinate of the L_S frame is assumed to be constant at the MPC stage, which is enforced by task space control. The z-coordinate of the L_S frame, $p_{S,z}$, is controlled by the robot arm alone, while the x-coordinate $p_{S,x}$ is a sum of two components

$$p_{S,x} = p_{B,x} + p_{A,x} \tag{1}$$

where $p_{A,x}$ is the x coordinate of the position of the L_S frame with respect to L_B, controlled by the robot arm, and $p_{B,x}$ is the x coordinate of the position of L_B with respect to L_G, controlled by the mobile base.

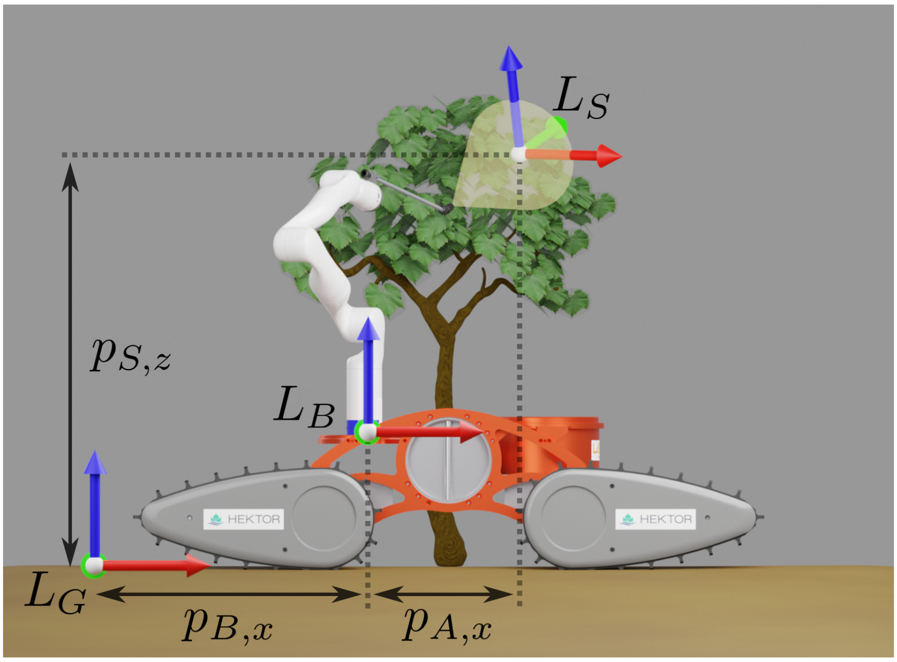

Figure 2. Three coordinate frames are defined: a global frame at the ground level L_G, mobile base frame L_B and the spray frame L_S. The x, y and z axes of the coordinate frames are represented with red, green and blue arrows, respectively.

The overall control diagram of the system is shown in Figure 3. The reference trajectory for the spray frame is computed based on the canopy description, explained in Section 2.1. This reference is used as an input to the MPC solver, which selects optimal mobile base and manipulator task space trajectories. Only the first set of optimized control inputs is sent to the robot, after which the trajectories are recalculated based on the system state feedback and the updated reference trajectory. The task space commands for the manipulator are converted into joint velocities by the task space controller.

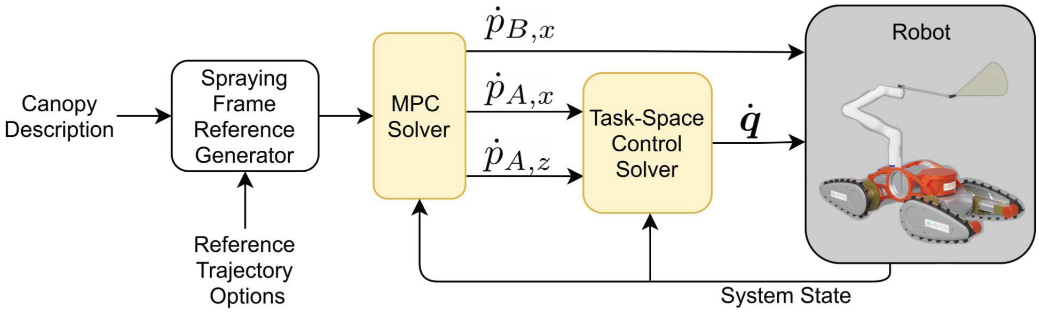

Figure 3. Overall system control diagram. The trajectory of the reference spray frame is generated based on the canopy description and used as input to the MPC solver. The MPC solver provides the velocity of the mobile base along the row and the velocities of the robot arm in the task space. The task space control solver converts the desired velocities in the task space into joint velocity commands $\dot{\boldsymbol{q}}$.

2.1. Reference Spray Frame Trajectory

A reference lawnmower trajectory is generated within the upper and lower boundaries of the foliage, with the aim of providing spray coverage while reducing waste (Figure 4). The foliage canopy description is used to compute the vertices of the lawnmower pattern, along with the Spray Width and Height Offset parameters (Figure 4). These parameters are tuned with respect to the spray pattern of the nozzle mounted at the end-effector of the robot arm. A constant velocity piecewise linear function is used as the reference trajectory between the lawnmower pattern vertices. The reference velocity of the spray frame, together with the flow of the spraying agent through the nozzle, affects the coverage of a fixed area of the canopy. For a fixed amount of spray flow, lower reference velocity will result in more coverage.

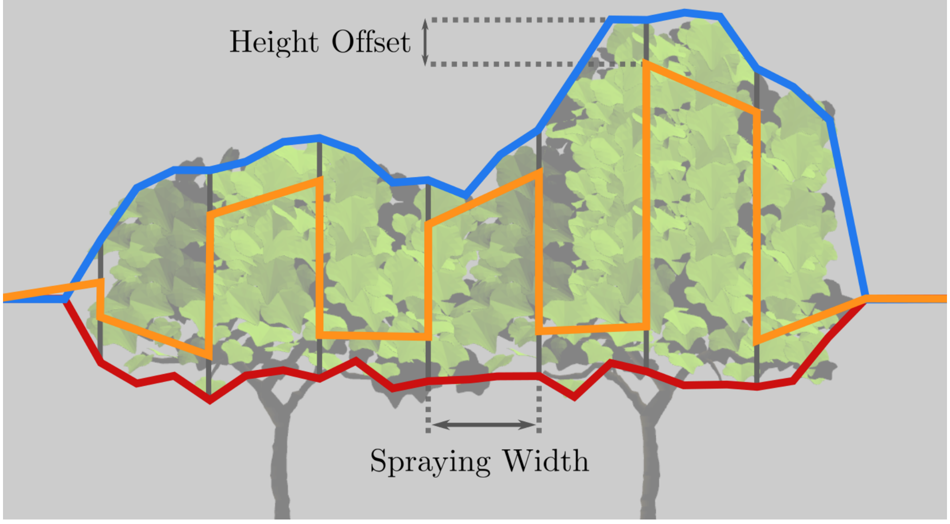

Figure 4. A reference lawnmower trajectory is shown with an orange line. The canopy description is represented by a blue and a red line, representing the upper and lower boundaries of the foliage, respectively.

The canopy description can be created either manually, by an operator, or using foliage detection. In the experiments presented in this paper, the manual approach was used to select the boundaries. In the future work, the plan is to use a foliage detection algorithm for this purpose.

2.2. MPC Algorithm

The MPC algorithm selects trajectories for the mobile base and the robot arm task space that follow a two-dimensional reference, while minimizing an objective function described in this section. The optimization is performed over a prediction horizon N, with fixed discretization time steps T. Although the velocities of the mobile base and the robot arm are used for low-level control (as seen in Figure 3), MPC operates in the acceleration space to achieve the desired behavior. The system state and control inputs in the k-th step, $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$, respectively, are

$$\boldsymbol{x}(k) = \begin{bmatrix} p_{B,x}(k) \\ p_{A,x}(k) \\ p_{S,z}(k) \\ \dot{p}_{B,x}(k) \\ \dot{p}_{A,x}(k) \\ \dot{p}_{S,z}(k) \end{bmatrix} \in \mathbb{R}^6 \quad \boldsymbol{u}(k) = \begin{bmatrix} \ddot{p}_{B,x}(k) \\ \ddot{p}_{A,x}(k) \\ \ddot{p}_{S,z}(k) \end{bmatrix} \in \mathbb{R}^3 \tag{2}$$

System state consists of positions $p_{B,x}(k)$, $p_{A,x}(k)$ and $p_{S,z}(k)$, seen in Figure 2, as well as the corresponding velocities. System inputs are the accelerations of the spray frame $\ddot{p}_{B,x}$, $\ddot{p}_{A,x}$ and $\ddot{p}_{S,z}$. The output of the system is a two-dimensional position of the spray frame L_S, relative to the global frame L_G. The output of the system in the k-th step $\boldsymbol{y}(k)$ is

$$\boldsymbol{y}(k) = \begin{bmatrix} p_{B,x}(k) + p_{A,x}(k) \\ p_{S,z}(k) \end{bmatrix} \in \mathbb{R}^2 \tag{3}$$

As already mentioned, the z component of the spray frame position $p_{S,z}$ is controlled solely by the robot arm, and the x component is controlled by both the robot arm and the mobile base (Equation (1)).

The system is therefore linear and its discretization results in the following system dynamics:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x}(k) + \underbrace{\begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^2}{2} \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{u}(k) \tag{4}$$

$$\boldsymbol{y}(k) = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{C}} \boldsymbol{x}(k) \tag{5}$$

MPC optimizes over a system input vector $\boldsymbol{U}$ containing N control inputs,

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}(0) \\ \boldsymbol{u}(1) \\ \vdots \\ \boldsymbol{u}(N-1) \end{bmatrix} \in \mathbb{R}^{3N} \tag{6}$$

The system state and output vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ are, respectively,

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}(0) \\ \boldsymbol{x}(1) \\ \vdots \\ \boldsymbol{x}(N-1) \end{bmatrix} \in \mathbb{R}^{6N}, \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}(0) \\ \boldsymbol{y}(1) \\ \vdots \\ \boldsymbol{y}(N-1) \end{bmatrix} \in \mathbb{R}^{2N} \tag{7}$$

The MPC optimization problem is defined as the following QP problem:

$$\begin{aligned} \min_{\boldsymbol{U}} \quad & W_y \|\boldsymbol{Y} - \boldsymbol{Y}_d\|^2 + \|\boldsymbol{W}_u \boldsymbol{U}\|^2 + \|\boldsymbol{W}_x \boldsymbol{X}\|^2 \\ \text{s.t.} \quad & \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ & \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ & \underline{\boldsymbol{x}} \leq \boldsymbol{x}(k) \leq \overline{\boldsymbol{x}} \\ & \underline{\boldsymbol{u}} \leq \boldsymbol{u}(k) \leq \overline{\boldsymbol{u}} \end{aligned} \tag{8}$$

where $\boldsymbol{Y}_d$ is the desired system output and W_y is the reference tracking weight. $\boldsymbol{W}_u$ and $\boldsymbol{W}_x$ are control input and system state weight matrices, respectively.

The reference tracking part of the criterion function ($W_y \|\boldsymbol{Y} - \boldsymbol{Y}_d\|^2$) is used to ensure that the system output follows the reference trajectory for the spray frame. The second part ($\|\boldsymbol{W}_u \boldsymbol{U}\|^2$) minimizes the system inputs: the acceleration of the vehicle along the row and the task space acceleration of the manipulator. The control input weight matrix is

$$\boldsymbol{W}_u = \begin{bmatrix} \boldsymbol{w}_u \\ \boldsymbol{w}_u \\ \vdots \\ \boldsymbol{w}_u \end{bmatrix}^{\mathrm{T}}, \boldsymbol{w}_u = \begin{bmatrix} w_{\ddot{p}_{B,x}} & 0 & 0 \\ 0 & w_{\ddot{p}_{A,x}} & 0 \\ 0 & 0 & w_{\ddot{p}_{S,z}} \end{bmatrix} \tag{9}$$

where the scalars $w_{\ddot{p}_{B,x}}$, $w_{\ddot{p}_{A,x}}$ and $w_{\ddot{p}_{A,z}}$ represent weights with respect to the acceleration of the mobile base along the row, and the acceleration of the robot arm in the x and z directions, respectively.

The third and final part of the criterion function ($\|\boldsymbol{W}_x \boldsymbol{X}\|^2$) minimizes a function of the system state vector. In particular, it minimizes the displacement of the robot arm in the x direction, with a weight matrix of the following form:

$$\boldsymbol{W}_x = \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_x \\ \vdots \\ \boldsymbol{w}_x \end{bmatrix}^{\mathrm{T}}, \ \boldsymbol{w}_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{p_{A,x}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

2.2.1. MPC Parameter Tuning

The main challenge in tuning the parameters of the criterion function (8) is the balance between robot arm and mobile base motion. The x component of the reference trajectory can be tracked by either robot arm or mobile base. The idea behind minimizing arm displacement and mobile base acceleration is to achieve the effect where the mobile base is responsible for slower, global changes in the reference trajectory and the arm is responsible for faster, local changes. Parameters are tuned by trial and error to achieve this effect. Extreme examples of mobile base and robot arm trajectories are shown in Figure 5, with the corresponding optimization weights presented in Table 1.

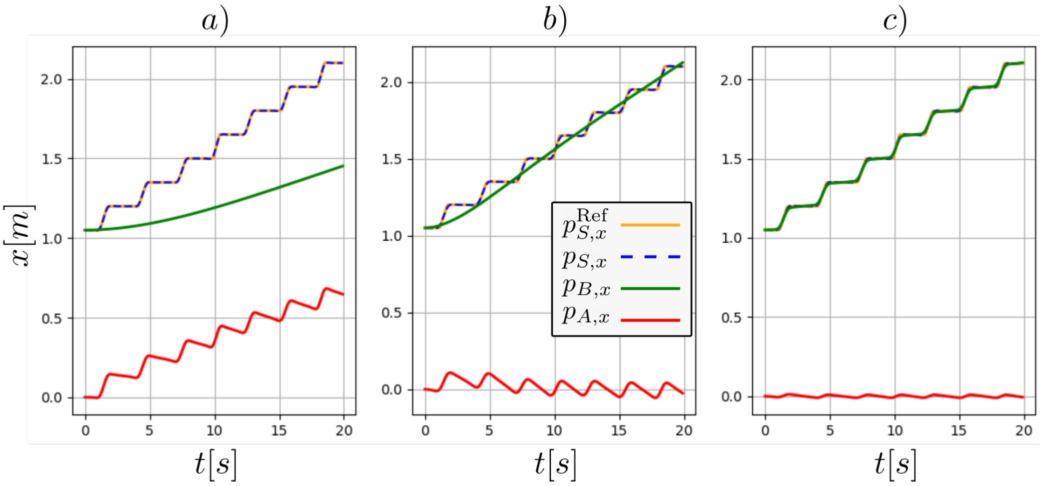

Figure 5. Solutions to the MPC optimization problem in the x direction using different criterion function parameters. Parameters are given in Table 1. Spray frame trajectory $p_{S,x}$ is a sum of $p_{A,x}$ and $p_{B,x}$.

Table 1. MPC criterion function parameters resulting in trajectories shown in Figure 5.

Figure 5.	**(a)**	**(b)**	**(c)**
W_y	800.0	800.0	800.0
$w_{\ddot{p}_{B,x}}$	800.0	80.0	8.0
$w_{\ddot{p}_{A,x}}, w_{\ddot{p}_{S,z}}$	4.0	8.0	16.0
$w_{p_{A,x}}$	0.5	1.0	2.0

The behavior of the trajectories selected by the MPC algorithm depends on the ratios between the different optimization weights rather than their exact value. Figure 5 shows solutions to the MPC problem with different ratios between the optimization weights, where the system assumes an initial state with zero velocity. Increasing the weight corresponding to the vehicle acceleration $w_{\ddot{p}_{B,x}}$ too much leads to small accelerations of the mobile base and thus to a large arm displacement (Figure 5a). Lowering the value of $w_{p_{A,x}}$ and $w_{\ddot{p}_{A,x}}$ has a similar effect. Lowering $w_{\ddot{p}_{B,x}}$ or increasing $w_{p_{A,x}}$ and $w_{\ddot{p}_{A,x}}$ too much (Figure 5c) leads to the effect where the mobile base follows the reference with large accelerations, while the robot arm displacement remains close to zero. To achieve the desired behavior (Figure 5b), parameters are tuned following a general tuning rule: a larger weight is used for minimization of the vehicle acceleration than for that of the robot arm. This, in combination with the minimization of the arm displacement, leads to the already discussed effect.

For a set of reference trajectories generated as shown in Section 2.1, it was concluded that different values can be used as weights for the MPC criterion function and still result in satisfactory overall system behavior. The values for the sampling period, horizon and weights of the criterion function used in the experiments can be found in Section 3.

2.2.2. MPC Constraints

The constraints on the system state and the control inputs have the following form:

$$\begin{bmatrix} -p_{B,x}^{\max} \\ -p_{A,x}^{\max} \\ -p_{S,z}^{\max} \\ -\dot{p}_{B,x}^{\max} \\ -\dot{p}_{A,x}^{\max} \\ -\dot{p}_{S,z}^{\max} \end{bmatrix} \leq \boldsymbol{x}(k) \leq \begin{bmatrix} p_{B,x}^{\max} \\ p_{A,x}^{\max} \\ p_{S,z}^{\max} \\ \dot{p}_{B,x}^{\max} \\ \dot{p}_{A,x}^{\max} \\ \dot{p}_{S,z}^{\max} \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} -\ddot{p}_{B,x}^{\max} \\ -\ddot{p}_{A,x}^{\max} \\ -\ddot{p}_{S,z}^{\max} \end{bmatrix} \leq \boldsymbol{u}(k) \leq \begin{bmatrix} \ddot{p}_{B,x}^{\max} \\ \ddot{p}_{A,x}^{\max} \\ \ddot{p}_{S,z}^{\max} \end{bmatrix} \tag{12}$$

The motion of the mobile base is left unconstrained ($p_{B,x}^{\max} = \infty$). The maximum velocity and acceleration of the mobile base, $\dot{p}_{B,x}^{\max}$ and $\ddot{p}_{B,x}^{\max}$, respectively, are known and directly enforced by these constraints. The maximum values of the robot arm velocities and accelerations in the task space ($\dot{p}_{A,x}^{\max}$, $\dot{p}_{S,z}^{\max}$, $\ddot{p}_{A,x}^{\max}$ and $\ddot{p}_{S,z}^{\max}$) must be determined experimentally, as they depend on the current arm configuration and the velocity and acceleration limits of each joint, which are not considered by the MPC algorithm. Instead, the joint space constraints are enforced at the level of the task space control algorithm.

2.3. Manipulator Task Space Control

Joint velocities $\dot{\boldsymbol{q}}$ are selected through task space control and are used to control the robot arm ($\boldsymbol{q}$ represents joint positions). As mentioned earlier, the MPC algorithm only considers the x and z coordinates of the spray frame position. In the planning phase (phase of the MPC algorithm), the position of the spray frame in the y direction ($p_{S,y}$) is assumed to be constant, and its roll, pitch and yaw angles are not considered (ϕ_T, θ_T and ψ_T, respectively, as shown in Figure 6).

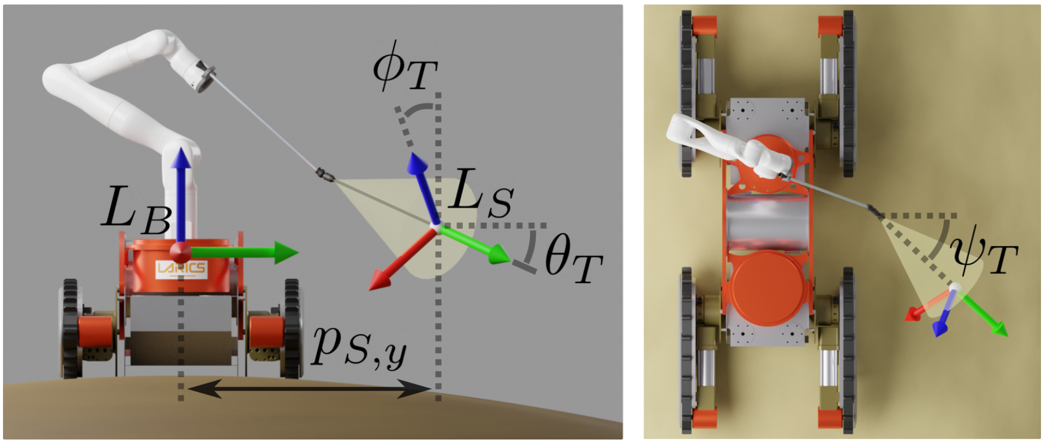

Figure 6. The orientation of the spray frame depends on the joint configuration of the robot arm. Roll, pitch and yaw angles are referred to as ϕ_T, θ_T and ψ_T, respectively.

To achieve the assumption of constant $p_{S,y}$, the desired task space velocity in the y direction, $\dot{p}_{S,y}$, is computed with a proportional controller

$$\dot{p}_{S,y} = K_{P,y}(p_{S,y}^{d} - p_{S,y}) \tag{13}$$

where $p_{S,y}^{d}$ is a desired value of $p_{S,y}$ and $K_{P,y}$ is the controller gain.

The desired linear spray frame velocity can be achieved with different angular velocities. To achieve the largest possible linear spray frame velocities, only the roll angle is controlled directly, while the pitch and yaw angles are not considered. The desired roll angle velocity is calculated with a proportional controller

$$\dot{\phi}_T = -K_{P,\phi}\phi_T \tag{14}$$

where $K_{P,\phi}$ is the controller gain and the desired roll angle is zero.

The desired task space velocity is four-dimensional and considers x, y, z position coordinates and the roll angle of the L_S frame,

$$\boldsymbol{v}_d^{4D} = \begin{bmatrix} \dot{p}_{A,x}^M \\ \dot{p}_{S,y} \\ \dot{p}_{S,z}^M \\ \dot{\phi}_T \end{bmatrix} \tag{15}$$

$\dot{p}_{A,x}^M$ and $\dot{p}_{S,z}^M$ are provided by MPC, and $\dot{p}_{S,y}$ and $\dot{\phi}_T$ are provided by proportional controllers (13) and (14), respectively. The four-dimensional task space velocity depends on the joint velocities as follows:

$$\boldsymbol{v}^{4D} = \boldsymbol{J}^{4D}\dot{\boldsymbol{q}} \tag{16}$$

where $\boldsymbol{J}^{4D}$ is the task space Jacobian.

In general, there are multiple solutions for $\dot{\boldsymbol{q}}$ that achieve the desired task space velocities. For this reason, an additional criterion by which the joint commands are selected is introduced. The desired joint velocities $\dot{\boldsymbol{q}}_d$ that drive the robot arm to a desired pose $\boldsymbol{q}_d$ are computed by another proportional controller,

$$\dot{\boldsymbol{q}}_d = K_{P,q}(\boldsymbol{q}_d - \boldsymbol{q}) \tag{17}$$

where $K_{P,q}$ is the controller gain.

Finally, the joint velocity commands are selected by solving the following QP problem:

$$\begin{aligned} \min_{\dot{\boldsymbol{q}}} \quad & \left\| \boldsymbol{J}^{4D}\dot{\boldsymbol{q}} - \boldsymbol{v}_d^{4D} \right\|^2 + w_p \|\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}\|^2 \\ \text{s.t.} \quad & \underline{\dot{\boldsymbol{q}}} \leq \dot{\boldsymbol{q}} \leq \overline{\dot{\boldsymbol{q}}} \end{aligned} \tag{18}$$

where $\underline{\dot{\boldsymbol{q}}}$ and $\overline{\dot{\boldsymbol{q}}}$ are the lower and upper bounds of the joint velocities, respectively, and w_p is the arm pose weight.

3. Results

Two experiments were conducted to evaluate the presented approach: a spraying demonstration in a vineyard and an external validation in an indoor environment equipped with Optitrack cameras. Both experiments were performed with the following optimization weights: $W_y = 800$, $w_{\ddot{p}_{B,x}} = 80$, $w_{\ddot{p}_{A,x}} = 8$, $w_{\ddot{p}_{A,z}} = 8$, $w_{p_{A,x}} = 1$. Parameter values were determined experimentally, as described in Section 2.2.1. Values of proportional controller gains used in the experiments were $K_{P,y} = 0.9$, $K_{P,\phi} = 0.1$ and $K_{P,q} = 1.5$. These values were also determined experimentally. The velocity of the reference lawnmower trajectory used in all experiments was 0.3 m/s. A prediction horizon of 40 steps was used, with a sampling period of $T = 0.1$ s. The optimization problems for the MPC and task space control were both solved using the BPMPD interior point solver for convex QP problems [36]. The average computation times of the MPC and task space control optimization problems were 3.998 ms and 0.201 ms, respectively, for 600 trials. This allows the presented approach to be used in a real-time scenario, even with larger prediction horizons. All experiments were performed on an Intel Core i7-10710U CPU @ 1.60 GHz.

3.1. Equipment

The control equipment used in the experiments is shown in Figure 7. All the software was run on an Intel NUC 10 PC with a Ubuntu 20.04 operating system. The Robot Operating System (ROS) was used as the middleware for controlling the robot arm, the mobile vehicle and the spraying agent pump.

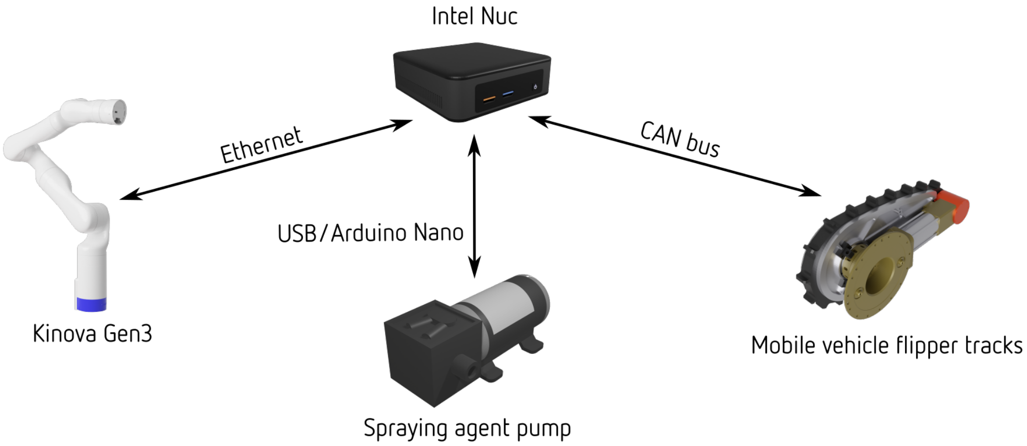

Figure 7. Communication diagram of the equipment used in the vineyard spraying experiments.

The Kinova Gen3 7 DoF robot arm uses an ethernet connection to communicate with ROS drivers running on the PC. As can be seen in Figure 3, low-level control of the robot arm is achieved via joint velocity commands $\dot{\boldsymbol{q}}$. Encoder measurements from the robot arm are used as process feedback. As feedback for the MPC algorithm, the robot arm components of the spray frame position $p_{A,x}$ and $p_{A,z}$ are calculated at every step of the control loop, using forward kinematics and joint position measurements $\boldsymbol{q}$. For the task space control algorithm, joint position data from the encoder are also used as feedback, to calculate the task space Jacobian $\boldsymbol{J}^{4D}$.

Mobile vehicle flipper tracks are actuated by Maxon EC 45 brushless motors, which are controlled using EPOS 70/10 brushless motor drives. EPOS drives communicate with the PC via a CAN bus. ROS drivers for the EPOS drives are used to control the velocities and gather encoder data of each motor. The mobile base component of the spray frame position $p_{B,x}$ is calculated using encoder data odometry and is used as feedback for the MPC algorithm.

Finally, the spraying agent pump is controlled with an Arduino Nano microcontroller board and is either turned on or off based on the current position of the spray frame with respect to the row description.

3.2. Vineyard Spraying Demonstration

The first experiment is a demonstration of the presented approach in a vineyard (Figure 8). The canopy description used in the experiment (Figure 9) was selected manually by an operator through a simple graphical user interface (GUI).

A graph showing the reference lawnmower trajectory tracking during the experiment is given in Figure 10. Here, the position of the spray frame $\boldsymbol{p}_S$ is calculated based on the encoder feedback from the robot arm, and the odometry of the mobile base. Since only the odometry feedback is used for mobile base control, its reference tracking is also evaluated with an external sensor, as part of the second experiment in Section 3.3. The MPC criterion function is tuned to follow the reference trajectory imperfectly, to minimize the accelerations of the mobile base and the robot arm end-effector. Figure 11 shows the reference tracking with respect to the actual reference trajectory of the spray frame $\boldsymbol{p}_S^*$, which is formed by accumulating the first control inputs of the optimal control sequences calculated by the MPC algorithm at each control step. There is still some error in the reference tracking caused by the non-ideal following of the joint velocity commands and the error in task space control. The task space Jacobian depends on the joint positions and is computed at the beginning of each control time step. During this time step, the joint positions, and the Jacobian, change. Larger joint velocities and larger time steps result in larger task space control errors. The position tracking errors are shown in Table 2. Spray frame position tracking was evaluated, with a measured root mean square (RMS) error of

4.32 mm, and the maximum error of 22.16 mm. The magnitude of the errors is a result of the fact that the spraying frame is defined at a certain distance to the last link of the robot arm, which makes its position sensitive to small errors in the robot arm joint positions. Based on the imperfections of the spray nozzle and its spraying area, the presented errors are adequate to ensure the effectiveness of the vineyard spraying task. There is a trade-off between reference tracking errors and the velocity of the reference spray frame trajectory. Reducing this velocity would result in slower task execution, but would also reduce the reference tracking errors.

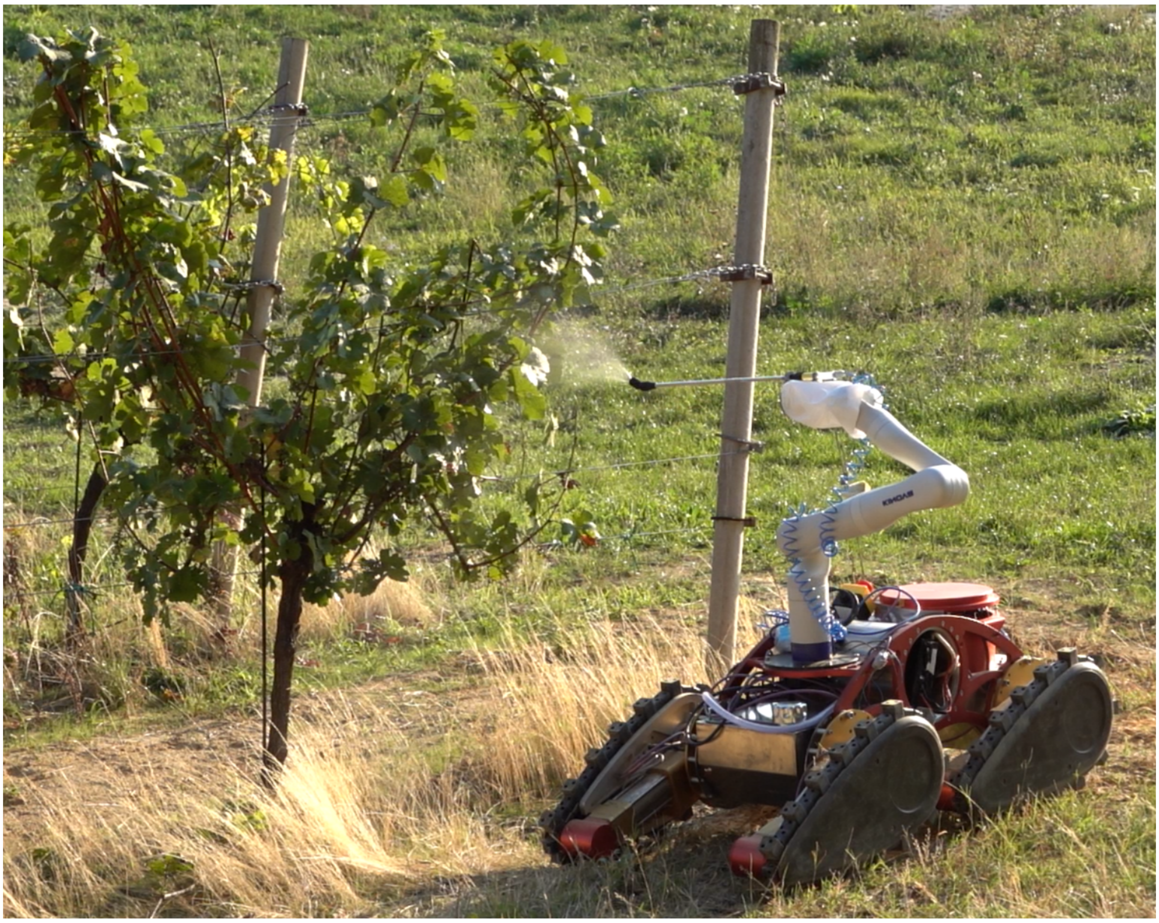

Figure 8. Mobile manipulator performing a spraying experiment in a vineyard.

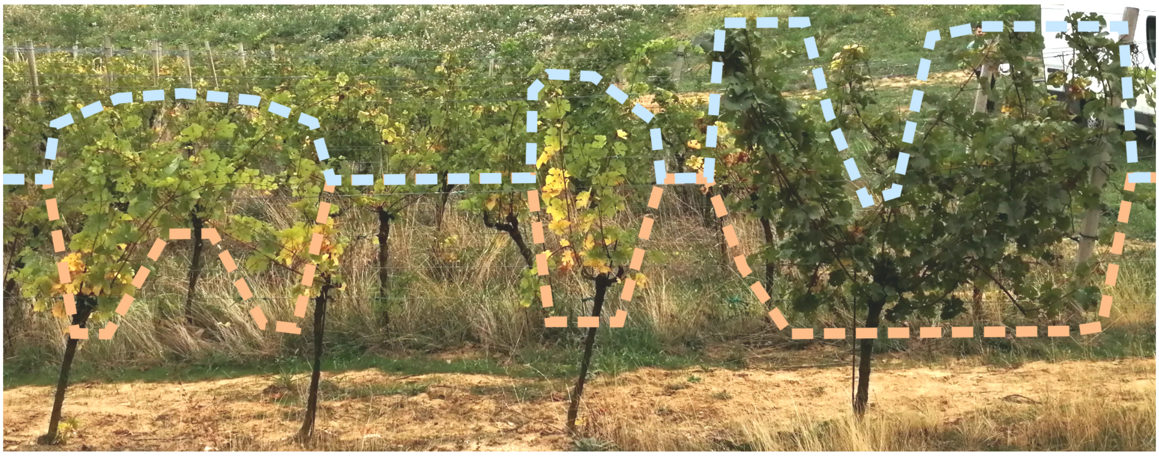

Figure 9. The canopy description used for the experiment selected by the operator.

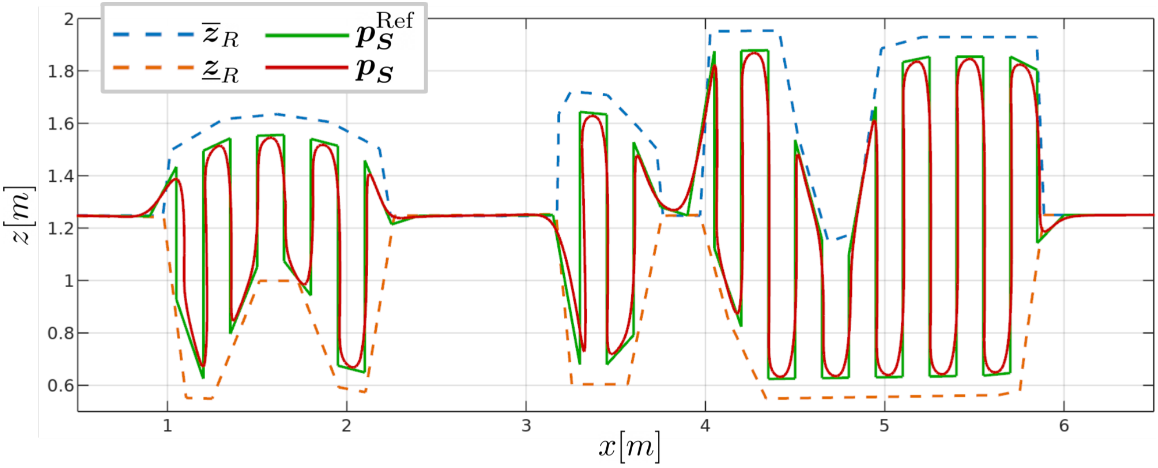

Figure 10. Overall spray frame reference tracking. $\boldsymbol{p}_S^{\text{Ref}}$ represents a reference lawnmower trajectory generated based on the row description, where $\overline{z}_R$ and $\underline{z}_R$ represent the upper and lower foliage boundaries, respectively. The spray frame position $\boldsymbol{p}_S$ during the experiment is represented by a red line.

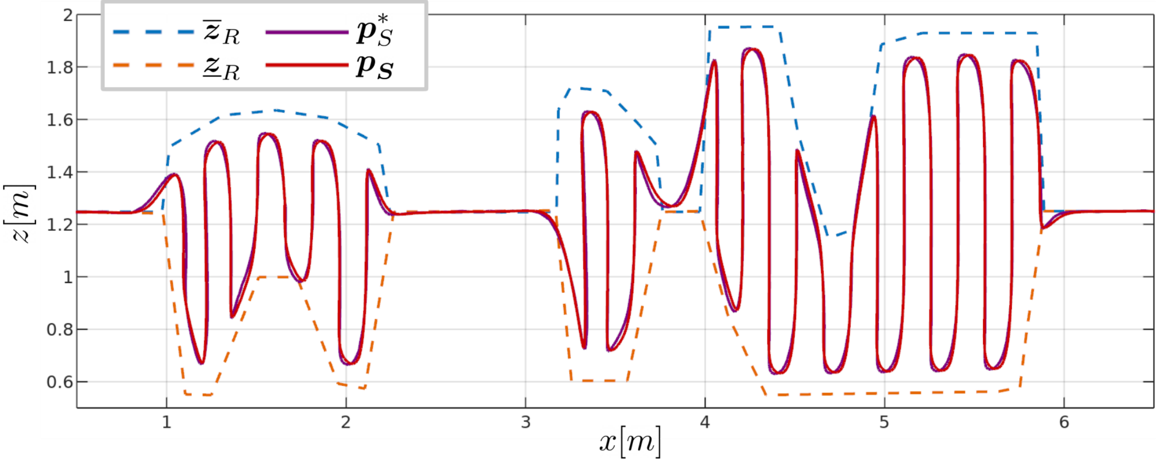

Figure 11. Spray frame tracking with respect to the optimal reference trajectory generated by the MPC algorithm. $\boldsymbol{p}_S^*$ represents the optimal trajectory of the spray frame. This differs from the ideal lawnmower trajectory due to MPC tuning that sacrifices reference tracking to minimize the accelerations of the mobile base and the manipulator end-effector. $\overline{z}_R$ and $\underline{z}_R$ represent the upper and lower foliage boundaries, respectively. The spray frame position $\boldsymbol{p}_S$ during the experiment is represented by a red line.

Table 2. Spray frame position errors during the vineyard spraying experiment.

	$\boldsymbol{p}_S$	$p_{S,x}$	$p_{S,y}$	$p_{S,z}$
RMS error [mm]	4.32	0.90	3.60	2.20
max error [mm]	22.16	3.92	22.16	18.93

A non-uniform vine row structure allows the demonstration of the adaptability of the presented approach. The velocity of the vehicle adapts to the row description by automatically accelerating in areas without vines and decelerating to the lowest velocities in areas with the highest vines (Figure 12). The footage of the vineyard spraying experiment can be seen in the accompanying video https://youtu.be/BDO7qQldmyQ, accessed on 8 March 2022.

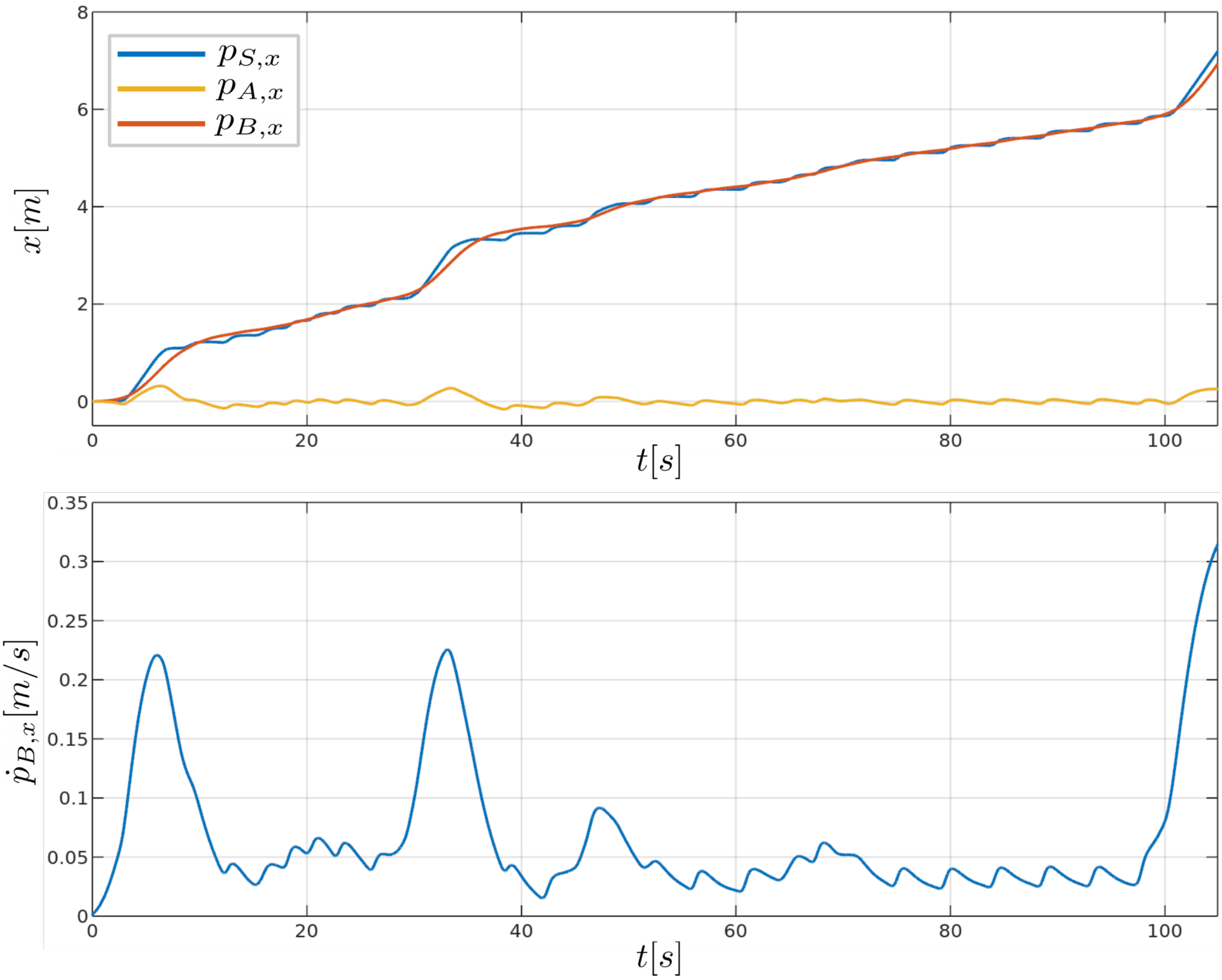

Figure 12. The upper diagram shows the x component of the reference trajectory $p_{S,x}$, along with the robot arm and mobile base components, $p_{A,x}$ and $p_{B,x}$, respectively. The bottom graph shows the forward velocity of the vehicle during the experiment.

3.3. Optitrack Validation

The second experiment was conducted in an indoor environment equipped with the Optitrack camera system, which was used to evaluate the reference tracking with an external sensor. The reference tracking error in this experiment includes the error of the odometry-based mobile base control, the vibrations of the mobile base, the errors present in the robot arm encoder measurements and the Optitrack measurement noise. For practical reasons, the Optitrack markers were placed at the last joint of the robot arm (Figure 13), and the position of the spray frame was calculated using a single static transformation.

Graphs showing the x and z components of the spray frame position during the experiment, along with the tracking errors, are shown in Figures 14 and 15, respectively, and the corresponding error data are given in Table 3. In this experiment, the measured root mean square (RMS) error and the maximum error are equal to 9.76 mm and 52.81 mm, respectively. Errors are larger than the ones in the previous experiment, which is expected due to the already mentioned additional errors that the external sensors are able to capture, and a significant amount of measurement noise. External sensor data confirm that the odometry-based control does not result in a significant drift of the mobile base, as seen in Figure 14.

In Figure 16, the overall spray frame position calculated from the Optitrack data is compared to the position calculated using the joint encoder and vehicle odometry feedback.

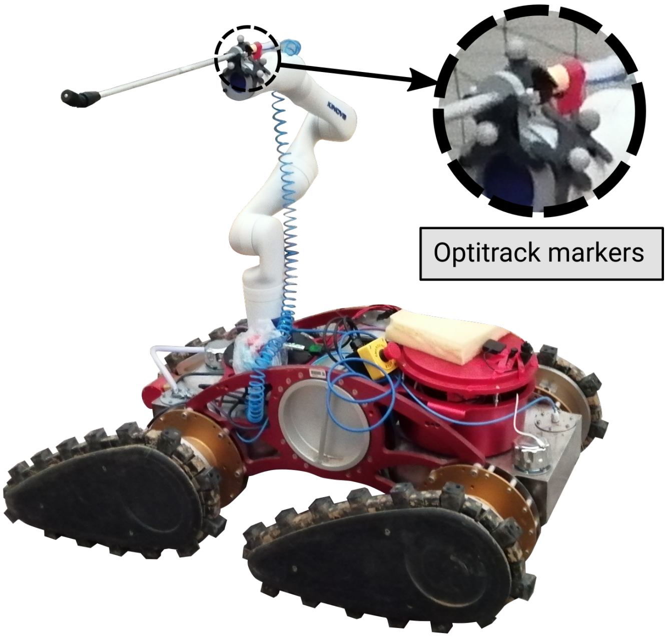

Figure 13. For the second experiment, reference tracking is externally validated using Optitrack cameras to measure the position of the spray frame in the real world. Optitrack markers are attached to the end-effector of the robot arm.

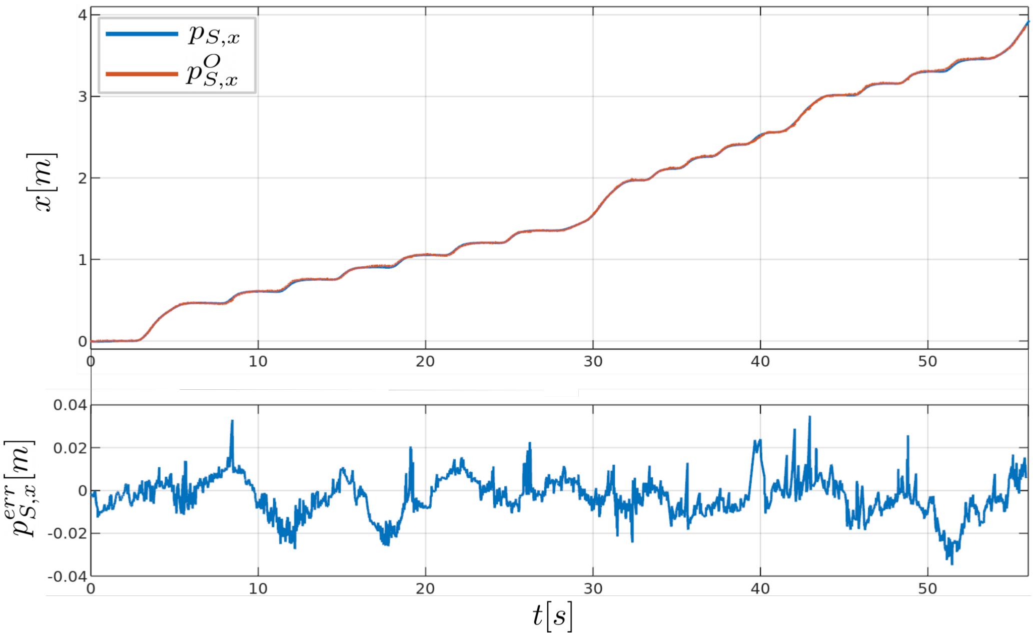

Figure 14. Comparison between the x component of the spray frame position determined by the encoder measurements, and that determined externally via the Optitrack camera system, denoted $p_{S,x}$ and $p^O_{S,x}$, respectively. The bottom plot shows the corresponding error $p^{err}_{S,x}$.

Table 3. Spray frame position errors measured with the Optitrack camera system, during the indoor experiment.

	p_S	$p_{S,x}$	$p_{S,z}$
RMS error [mm]	9.76	7.86	5.79
max error [mm]	52.81	36.59	52.779

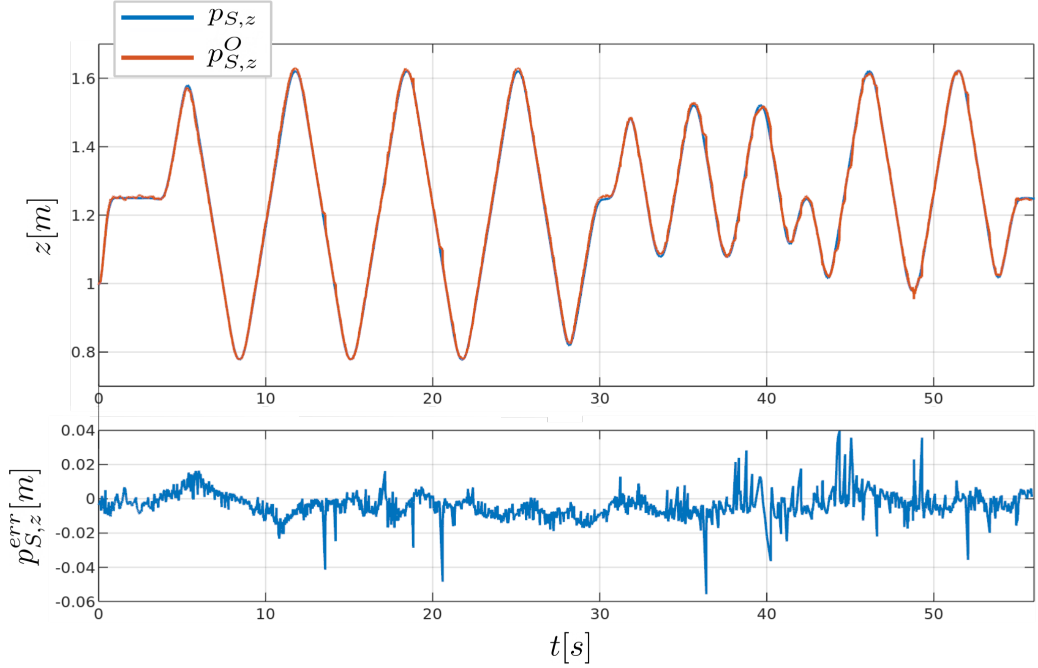

Figure 15. Comparison between the z component of the spray frame position determined by the encoder measurements, and that determined externally via the Optitrack camera system, denoted $p_{S,z}$ and $p^O_{S,z}$, respectively. The bottom plot shows the corresponding error $p^{err}_{S,z}$.

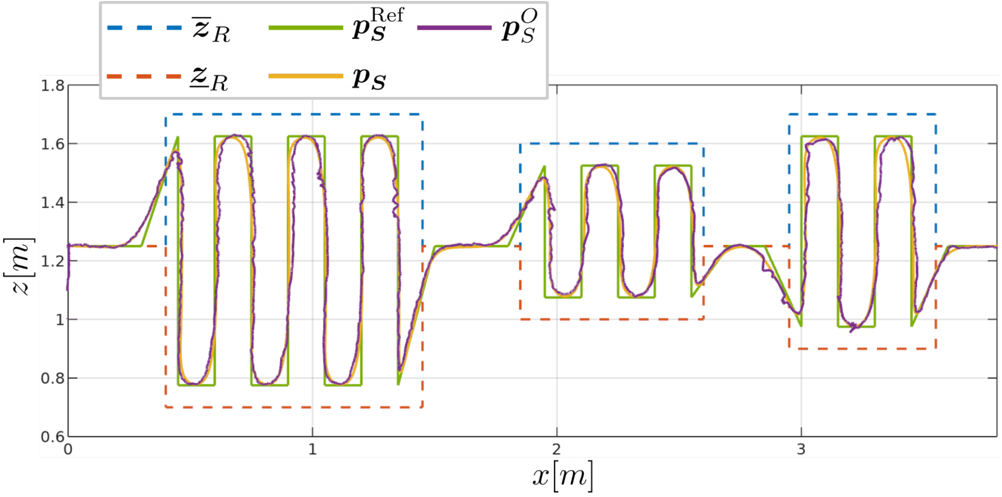

Figure 16. Comparison between the position of the spray frame obtained by the encoder measurements and the position obtained externally via the Optitrack camera system, denoted as $\boldsymbol{p}_S$ and $\boldsymbol{p}_S^O$, respectively. $\boldsymbol{p}_S^{Ref}$ represents the reference lawnmower trajectory, and $\bar{z}_R$ and $\underline{z}_R$ represent the upper and lower foliage boundaries, respectively.

As mentioned earlier, the task space controller selects joint velocities that follow the desired linear and roll spray frame velocities, while attempting to maintain the desired robot arm joint configuration. This results in the yaw and pitch angles of the spray frame shown in Figure 17. This type of control results in a pitch orientation (θ_T) graph similar to the z position, as shown in Figure 15. Similarly, the yaw orientation graph (ψ_T) follows the motion of the robot arm in the x direction.

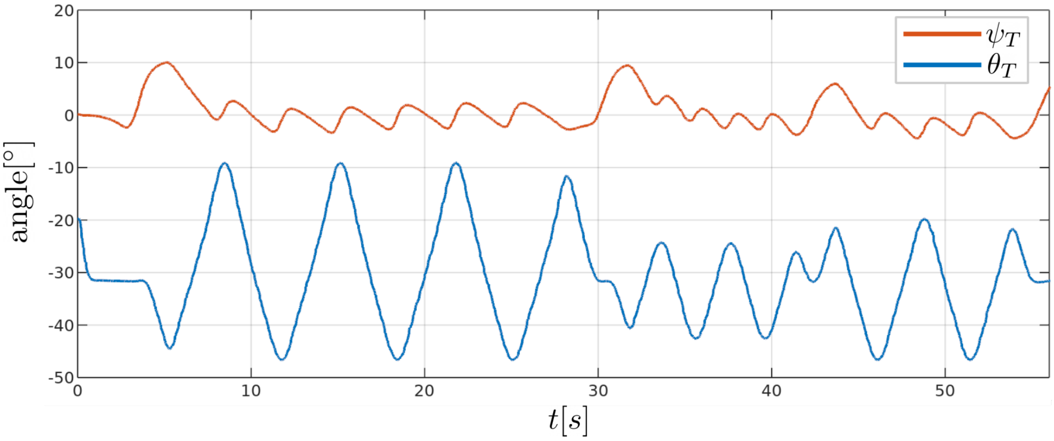

Figure 17. Spray frame orientation during the indoor experiment. The pitch and yaw angles of the spray frame are denoted as θ_T and ψ_T, respectively. These angles are not directly controlled, but are a result of the task space control criterion function.

4. Conclusions and Future Work

In this paper, a vineyard spraying algorithm for mobile manipulators is presented, based on task space model predictive control. The reference is generated based on grapevine canopy description, with the aim of minimizing unnecessary spraying waste and pollution.

There are certain limitations to the presented method. The time required to spray a row of grapevines is limited by the maximum velocity of the vehicle, as well as the maximum joint velocities of the robot arm. Task space control is used to calculate the joint velocity commands for the robot arm, which are not considered in the planning phase (MPC phase) of the algorithm. This could potentially lead to large spray frame velocities that cannot be tracked by the task space controller. Therefore, some experimentation is required to determine the maximum feasible velocity of the lawnmower pattern reference trajectory. Moreover, the task space control algorithm has no direct way of considering joint position constraints of the robot arm. This problem is dealt with indirectly, by allowing different angular velocities of the spraying frame, and constraining the optimization problem in such a way that the solutions moving the joints towards the desired configuration are preferred. No problems were encountered in the experiments regarding joint position constraints. The mobile base is controlled based on odometry feedback, which may lead to certain reference tracking problems since there is no external sensing. The second experiment shows that the open loop control performs well, mainly due to the fact that the vehicle moves in a straight line, which allows precise odometry. In future work, the plan is to close this control loop using a localization algorithm. Moreover, the tilt of the vehicle and other effects of uneven terrain are not taken into account in the current state of the algorithm, which could also be incorporated into future work. In the presented experiments, operator-selected grapevine row description was used. Manual selection of canopy areas proved to be error-prone, tedious and time-consuming. In the future work, a foliage detection algorithm is going to be incorporated for the purpose of generating a grapevine row description. Since the detection algorithm must be robust to changing lighting conditions, it is planned to be based on a combination of deep learning and depth information captured by an RGBD camera. The depth information acquired by the RGBD camera using infrared projection is sensitive to sunlight, so a camera based on pure stereo vision would be suitable for this task.

The presented method was evaluated in a vineyard spraying experiment, demonstrating its ability to adapt to a specific grapevine row structure. Mobile base velocity adapts to the row structure, which can be seen in the accompanying video and the graphs presented in Section 3.2. An additional experiment was performed evaluating the reference tracking with Optitrack cameras as external sensors. Error data show the 4.32 mm and 9.76 mm RMS errors in spray frame position, during the first and second experiment, respectively.

Since the spray frame is located at a certain distance to the last link of the robot arm, its position is sensitive to small joint position errors. The presented error values are sufficiently small for the task of vineyard spraying, while a trade-off exists between reference tracking precision and the time required to execute the task.

The focus of this work was on the control algorithm that sprays a single row of vines. In the future, mission planning and navigation would allow the mobile manipulator to autonomously treat the entire vineyard by entering each row and executing the presented algorithm. Experiments evaluating the spray quality using a water-sensitive paper are planned in the future. Extensive experiments to determine the impact of the presented method on plant health and fruit production and compare it to manual spraying are to be conducted. The presented method will be tested for the task of fruit spraying rather than spraying the entire foliage, which is the focus of this article. Another challenge is the presence of dust in the vineyard, from the influence of which the equipment must be adequately protected. Moreover, while excessive robot arm heating was not noticed during the presented experiments, it could present a potential problem in the case of prolonged robot operation. In this case, some form of active cooling could be used to mitigate the problem. Currently, the spray tank has a volume of 30 L, which will be increased in the future.

Author Contributions: Conceptualization, Z.K., G.V. and I.V.; methodology, I.V.; software, I.V.; validation, I.V. and G.V.; investigation, I.V.; writing—original draft preparation, I.V.; writing—review and editing, G.V. and Z.K.; visualization, I.V.; supervision, Z.K. and G.V.; project administration, Z.K. and G.V.; funding acquisition, Z.K. and G.V. All authors have read and agreed to the published version of the manuscript.

Funding: The research work presented in this article has been supported by the project titled Heterogeneous Autonomous Robotic system in Viticulture and Mariculture (HEKTOR), financed by the European Union through the European Regional Development Fund—The Competitiveness and Cohesion Operational Programme (KK.01.1.1.04.0041).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MPC	Model Predictive Control
QP	Quadratic Programming
CPU	Central Processing Unit
GUI	Graphical User Interface
RMS	Root Mean Square
ROS	Robot Operating System

References

1. Oliveira, L.F.P.; Moreira, A.P.; Silva, M.F. Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead. *Robotics* **2021**, *10*, 52. [CrossRef]
2. Adamides, G. Agricultural Robots in Targeted Spraying: A mini State-of-the-Art review. *Robot. Autom. Eng. J.* **2017**, *2*, 5581. [CrossRef]
3. Hektor Project Website. Available online: hektor.fer.hr (accessed on 8 March 2022).
4. Goričanec, J.; Kapetanović, N.; Vatavuk, I.; Hrabar, I.; Vasiljević, G.; Gledec, G.; Stuhne, D.; Bogdan, S.; Orsag, M.; Petrović, T.; et al. Heterogeneous autonomous robotic system in viticulture and mariculture-project overview. In Proceedings of the 2021 16th International Conference on Telecommunications (ConTEL), Zagreb, Croatia, 30 June–2 July 2021; pp. 181–188. [CrossRef]
5. Monta, M.; Kondo, N.; Shibano, Y. Agricultural robot in grape production system. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–27 May 1995; Volume 3, pp. 2504–2509. [CrossRef]
6. Berenstein, R.; Shahar, O.B.; Shapiro, A.; Edan, Y. Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. *Intell. Serv. Robot.* **2010**, *3*, 233–243. [CrossRef]

7. Oberti, R.; Marchi, M.; Tirelli, P.; Calcante, A.; Iriti, M.; Hočevar, M.; Baur, J.; Pfaff, J.; Schütz, C.; Ulbrich, H. Selective spraying of grapevine's diseases by a modular agricultural robot. *J. Agric. Eng.* **2013**, *44*, 149–153. [CrossRef]
8. Oberti, R.; Marchi, M.; Tirelli, P.; Calcante, A.; Iriti, M.; Tona, E.; Hočevar, M.; Baur, J.; Pfaff, J.; Schütz, C.; et al. Selective spraying of grapevines for disease control using a modular agricultural robot. *Biosyst. Eng.* **2016**, *146*, 203–215. [CrossRef]
9. de Soto, M.G.; Emmi, L.; Perez-Ruiz, M.; Aguera, J.; de Santos, P.G. Autonomous systems for precise spraying—Evaluation of a robotised patch sprayer. *Biosyst. Eng.* **2016**, *146*, 165–182. [CrossRef]
10. Botterill, T.; Paulin, S.; Green, R.; Williams, S.; Lin, J.; Saxton, V.; Mills, S.; Chen, X.; Corbett-Davies, S. A Robot System for Pruning Grape Vines. *J. Field Robot.* **2016**, *34*, 1100–1122. [CrossRef]
11. Berenstein, R. The use of agricultural robots in crop spraying/fertilizer applications. In *Robotics and Automation for Improving Agriculture*; Burleigh Dodds Science Publishing: Cambridge, UK, 2019; pp. 109–136. [CrossRef]
12. Santos, L.; Santos, F.; Mendes, J.; Costa, P.; Lima, J.; Reis, R.; Shinde, P. Path Planning Aware of Robot's Center of Mass for Steep Slope Vineyards. *Robotica* **2019**, *38*, 684–698. [CrossRef]
13. Cantelli, L.; Bonaccorso, F.; Longo, D.; Melita, C.D.; Schillaci, G.; Muscato, G. A Small Versatile Electrical Robot for Autonomous Spraying in Agriculture. *AgriEngineering* **2019**, *1*, 391–402. [CrossRef]
14. Santos, L.C.; Aguiar, A.S.; Santos, F.N.; Valente, A.; Ventura, J.B.; Sousa, A.J. Navigation Stack for Robots Working in Steep Slope Vineyard. In *Advances in Intelligent Systems and Computing*; Springer International Publishing: Cham, Switzerland, 2020; pp. 264–285. [CrossRef]
15. Santos, L.C.; Aguiar, A.S.; Santos, F.N.; Valente, A.; Petry, M. Occupancy Grid and Topological Maps Extraction from Satellite Images for Path Planning in Agricultural Robots. *Robotics* **2020**, *9*, 77. [CrossRef]
16. de Aguiar, A.S.P.; dos Santos, F.B.N.; dos Santos, L.C.F.; de Jesus Filipe, V.M.; de Sousa, A.J.M. Vineyard trunk detection using deep learning—An experimental device benchmark. *Comput. Electron. Agric.* **2020**, *175*, 105535. [CrossRef]
17. Sarri, D.; Lombardo, S.; Lisci, R.; Pascale, V.D.; Vieri, M. AgroBot Smash a Robotic Platform for the Sustainable Precision Agriculture. In *Lecture Notes in Civil Engineering*; Springer International Publishing: Cham, Switzerland, 2020; pp. 793–801. [CrossRef]
18. Roure, F.; Moreno, G.; Soler, M.; Faconti, D.; Serrano, D.; Astolfi, P.; Bardaro, G.; Gabrielli, A.; Bascetta, L.; Matteucci, M. GRAPE: Ground Robot for vineyArd Monitoring and ProtEction. In *ROBOT 2017: Third Iberian Robotics Conference*; Springer International Publishing: Cham, Switzerland, 2017; pp. 249–260. [CrossRef]
19. Astolfi, P.; Gabrielli, A.; Bascetta, L.; Matteucci, M. Vineyard Autonomous Navigation in the Echord++ GRAPE Experiment *IFAC-PapersOnLine* **2018**, *51*, 704–709. [CrossRef]
20. Kerkech, M.; Hafiane, A.; Canals, R. Vine disease detection in UAV multispectral images using optimized image registration and deep learning segmentation approach. *Comput. Electron. Agric.* **2020**, *174*, 105446. [CrossRef]
21. Bouloumpasi, E.; Theocharis, S.; Karampatea, A.; Pavlidis, S.; Mamalis, S.; Koundouras, S.; Merou, T.; Vrochidou, E.; Pachidis, T.; Manios, M.; et al. Exploration of viticultural tasks to be performed by an autonomous robot: Possibilities and limitations. In Proceedings of the 11th International Scientific Agriculture Symposium (AGROSYM 2020), Jahorina, Bosnia and Herzegovina, 8–9 October 2020; pp. 56–61.
22. Vrochidou, E.; Tziridis, K.; Nikolaou, A.; Kalampokas, T.; Papakostas, G.A.; Pachidis, T.P.; Mamalis, S.; Koundouras, S.; Kaburlasos, V.G. An Autonomous Grape-Harvester Robot: Integrated System Architecture. *Electronics* **2021**, *10*, 1056. [CrossRef]
23. Adamides, G.; Katsanos, C.; Constantinou, I.; Christou, G.; Xenos, M.; Hadzilacos, T.; Edan, Y. Design and development of a semi-autonomous agricultural vineyard sprayer: Human-robot interaction aspects. *J. Field Robot.* **2017**, *34*, 1407–1426. [CrossRef]
24. Ren, S.; Xie, Y.; Yang, X.; Xu, J.; Wang, G.; Chen, K. A Method for Optimizing the Base Position of Mobile Painting Manipulators. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 370–375. [CrossRef]
25. Liu, G.; Sun, X.; Liu, Y.; Liu, T.; Li, C.; Zhang, X. Automatic spraying motion planning of a shotcrete manipulator. *Intell. Serv. Robot.* **2021**. [CrossRef]
26. Wang, G.; Yu, Q.; Ren, T.; Hua, X.; Chen, K. Task planning for mobile painting manipulators based on manipulating space. *Assem. Autom.* **2018**, *38*, 57–66. [CrossRef]
27. Wieber, P.B. Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. In Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, 4–6 December 2006. [CrossRef]
28. Pankert, J.; Hutter, M. Perceptive Model Predictive Control for Continuous Mobile Manipulation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6177–6184. [CrossRef]
29. Elsisi, M. Optimal design of nonlinear model predictive controller based on new modified multitracker optimization algorithm. *Int. J. Intell. Syst.* **2020**, *35*, 1857–1878. [CrossRef]
30. Elsisi, M.; Ebrahim, M.A. Optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics. *Int. J. Intell. Syst.* **2021**, *36*, 6968–6987. [CrossRef]
31. Elsisi, M. Improved grey wolf optimizer based on opposition and quasi learning approaches for optimization: Case study autonomous vehicle including vision system. *Artif. Intell. Rev.* **2022**. [CrossRef]
32. Furuno, S.; Yamamoto, M.; Mohri, A. Trajectory planning of mobile manipulator with stability considerations. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), Newport Beach, CA, USA, 14–19 September 2003. [CrossRef]

33. Avanzini, G.B.; Zanchettin, A.M.; Rocco, P. Constrained model predictive control for mobile robotic manipulators. *Robotica* **2017**, *36*, 19–38. [CrossRef]
34. Minniti, M.V.; Farshidian, F.; Grandia, R.; Hutter, M. Whole-Body MPC for a Dynamically Stable Mobile Manipulator. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3687–3694. [CrossRef]
35. Korayem, M.; Nazemizadeh, M.; Azimirad, V. Optimal trajectory planning of wheeled mobile manipulators in cluttered environments using potential functions. *Sci. Iran.* **2011**, *18*, 1138–1147. [CrossRef]
36. Mészáros, C. The BPMPD interior point solver for convex quadratic problems. *Optim. Methods Softw.* **1999**, *11*, 431–449. [CrossRef]