


Article

Design and Optimization of Non-Coplanar Orbits for Orbital Photovoltaic Panel Cleaning Robots

Yingjie Zhao ^{1,2}, Yuming Qi ^{1,2,*} and Bing Xie ^{3,*} 

¹ Institute of Robotics and Intelligent Equipment, Tianjin University of Technology and Education, Tianjin 300222, China; shansuzhao@gmail.com

² Tianjin Key Laboratory of Intelligent Robot Technology and Application Affiliation, Tianjin 300222, China

³ College of Mechanical Engineering, Tianjin University of Science and Technology, Tianjin 300222, China

* Correspondence: chigym@163.com (Y.Q.); xiebing9502@126.com (B.X.); Tel.: +86-18622449127 (Y.Q.)

Abstract: Aiming at the problem that it is difficult for an orbital photovoltaic panel cleaning robot to span a large distance between photovoltaic panels, a method of designing and optimizing a non-coplanar orbit based on Bezier curves is proposed. Firstly, the robot's motion law is analyzed to obtain trajectory data for a single work cycle. Then, Bezier curves are utilized for trajectory design to ensure a smooth transition during the spanning motion phase. Thirdly, with the average value of the minimum distance between the Bezier curve and the point set data of the spanning motion phase as the optimization objective function, the nonlinear planning based on the SQP algorithm was adopted for the optimization of the upper and lower trajectories. Finally, the results of the case calculations indicate that the standard deviation of the optimized upper and lower trajectories was reduced by 35.63% and 40.57%, respectively. Additionally, the ADAMS simulation validation demonstrates that the trajectory errors of the four wheels decreased by a maximum of 8.79 mm, 23.78 mm, 10.11 mm, and 14.97 mm, respectively, thereby confirming the effectiveness of the trajectory optimization.

Keywords: non-coplanar orbits; Bezier curves; smoothing; nonlinear programming; dynamic simulation



Citation: Zhao, Y.; Qi, Y.; Xie, B. Design and Optimization of Non-Coplanar Orbits for Orbital Photovoltaic Panel Cleaning Robots. *Appl. Sci.* **2024**, *14*, 10388. <https://doi.org/10.3390/app142210388>

Academic Editor: Alessandro Lo Schiavo

Received: 18 September 2024

Revised: 25 October 2024

Accepted: 5 November 2024

Published: 12 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As global energy demand continues to rise and environmental issues become increasingly severe, the search for sustainable energy solutions has become imperative. The International Energy Agency [1] has developed a clean energy transition program in response to global warming, prompting the international community to seek effective strategies for reducing greenhouse gas emissions. China's National Emissions Trading System (ETS), which commenced operations in July 2021, aims to cap carbon dioxide (CO₂) emissions by 2030 and achieve carbon neutrality by 2060 [2]. In this context, renewable energy—particularly solar photovoltaic (PV) technology—has emerged as a crucial component of the global energy transition, owing to its clean and sustainable characteristics.

Despite the significant potential of solar PV technology, its efficiency is influenced by various factors, with the accumulation of dirt on the surface of PV panels being a major issue. In order to maintain the efficient power generation capacity of PV panels, regular cleaning is essential. Currently, there are several methods for cleaning PV panels, including manual cleaning, self-cleaning techniques, and mechanical cleaning. Manual cleaning is both costly and inefficient, making it unsuitable for the cleaning requirements of large-scale photovoltaic (PV) plants. While self-cleaning methods [3–5] can enhance the power generation efficiency of PV panels, this technology still relies on manual maintenance, which hinders the implementation of fully automated cleaning solutions. Given the limitations of manual and self-cleaning methods, PV panel cleaning robots have emerged to provide a more efficient and automated cleaning solution.

For example, Zhao et al. [6] developed a self-propelled PV panel cleaning robot based on hydraulic drive. Although the cleaning results are satisfactory, the robot is significantly

constrained by the terrain and lacks versatility. MG Antonelli et al. [7], Fan et al. [8], and Serbot Swiss Innovations [9] advocate for the use of autonomous mobile cleaning robots that can be deployed on solar photovoltaic (SPV) panels to facilitate their cleaning. The disadvantages of this solution include the difficulty the autonomous mobile cleaning robot faces in spanning large distances between PV panels as well as limitations imposed by the tilt angle of the panels. S. Rehman et al. [10] employed a drone to clean PV panels, representing a novel approach. However, it is important to note that the dust stirred up by the drone's rotating blades during operation is likely to settle back onto the surface of the PV panels. This phenomenon can be extremely detrimental to electricity generation from the panels.

As a result, various types of orbital cleaning robots have been developed [11–15]. The robots use the aluminum edges as tracks to facilitate their movement across the photovoltaic panels. The wheel assemblies maintain close contact with the edges of the panels, ensuring that the robots remain stable while on the surface. While they can achieve varying degrees of comprehensive cleaning by spanning linear orbits, they often overlook a critical factor: the installation environment of the PV panels. In fact, PV panels can exhibit highly complex relative positions based on the installation environment. For instance, in a desert setting, the foundation pillars of the PV panels may sink or tilt due to geological factors, leading to uneven relative positions among the panels. Consequently, the trajectory spanning the PV panels is no longer a simple linear path but rather a non-coplanar trajectory in three-dimensional space. Therefore, designing a non-coplanar orbital for robots to achieve efficient cleaning has become an urgent challenge.

The non-coplanar orbital serves as a crucial link for the PV panel cleaning robot to operate effectively, and its design concept is realized through the development and modeling of non-coplanar trajectories. In the design of non-coplanar trajectories, various mathematical methods are widely adopted to meet the needs of different application scenarios. Among them, the B-spline curve method [16], the NURBS curve method [17], and the Bezier curve method [18] are some of the primary techniques. For instance, Wang et al. [16] implemented adaptive motion planning for unmanned aerial vehicles (UAVs) using B-spline curves. However, B-spline curves may not accurately represent the desired shape in three-dimensional space, which can lead to potential issues in the design outcomes. Li et al. [17] employed non-uniform rational B-spline (NURBS) curves for Cartesian trajectory planning in industrial robots, significantly improving robotic motion performance. The fundamental linear decomposition of NURBS curves into 3D line shapes involves the computation of tangents and curvature, resulting in a highly complex decomposition process. Bezier curves, as a kind of spline curve, provide a more concise and controllable solution, which is quite widely used in the study of path planning [18–22]. In addition, Han et al. [23] combined aero-assisted orbital maneuvers with hp-adaptive pseudospectral methods to reduce fuel consumption. The application of this advanced approach to non-coplanar orbit design will also enhance the efficiency of cleaning robots. However, employing such sophisticated techniques in harsh non-vacuum environments necessitates the consideration of additional external factors, which undoubtedly increases the complexity and cost of the research and development process.

It is therefore particularly important to select an effective design method, as this will influence both the effectiveness and the complexity of the design. Pedro Villanueva et al. [24] discussed the Analytic Hierarchy Process (AHP), a structured decision-making framework that assists researchers in evaluating and comparing various design alternatives through multiple levels of criteria and sub-criteria. While applying AHP to non-coplanar trajectory design appears to facilitate the selection of an optimal design method, it is not fully effective as a decision-making tool due to its limited dynamic decision-making capabilities and reduced adaptability to uncertain environments. In conclusion, compared to the studies and findings of other researchers, Bezier curves seem to be a good method for designing non-coplanar orbits, simplifying the mathematical expressions and reducing the computational workload, making it more efficient in practical applications.

In order to fill the research gap concerning non-coplanar orbits for photovoltaic panel cleaning robots, this paper proposes a method for design and optimization utilizing Bezier curves. By analyzing the motion law of the cleaning robot, trajectory data for a single working cycle and the trajectory during the spanning motion phase are computed. Considering the challenges associated with accurately controlling the shape of trajectory design, this study proposes the Bezier curve method with scaling parameters and completes the optimization of non-coplanar trajectories using a nonlinear planning method based on the Sequential Quadratic Programming (SQP) algorithm.

The remainder of this paper is organized as follows: Section 2 presents a three-dimensional model of the cleaning robot, detailing the complex work scenarios it may encounter during operation. Additionally, the trajectory model, smoothing processing model, and optimization model are established. Section 3 verifies the validity of the proposed methods through case-based calculations and dynamic simulation results. Finally, Section 4 summarizes the entire paper and discusses future research directions.

2. Construction of Models

The physical model is constructed as illustrated in Figure 1. Specifically, Figure 1a presents the three-dimensional model of the PV panel cleaning robot, which primarily consists of the box structure, the main framework, and the sweeping component; Figure 1b illustrates the physical model of the PV panels for the most complex scenario regarding their relative positioning based on data collected from fieldwork. By analyzing the motion laws of photovoltaic panel cleaning robots during their operational processes, this study establishes a trajectory model, a smoothing processing model, and an optimization model to achieve the design and optimization of non-coplanar trajectories.

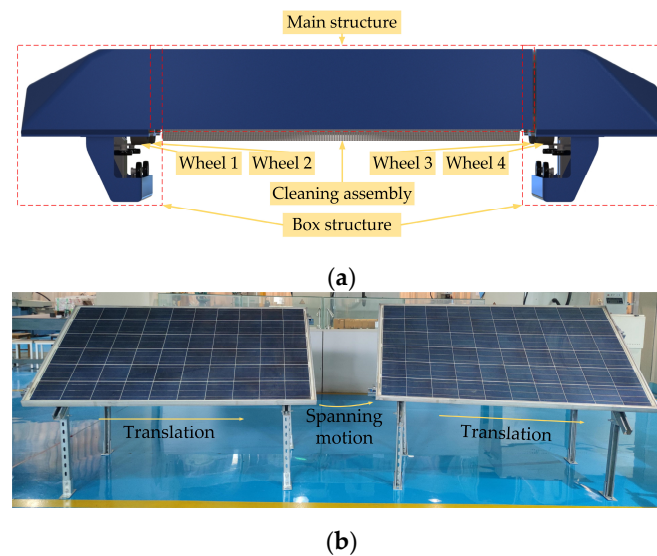


Figure 1. The physical model: (a) the three-dimensional model of the PV panel cleaning robot; (b) the physical model of the PV panels.

2.1. Motion Law Analysis of the PV Panel Cleaning Robot

The motion of the PV panel cleaning robot follows a certain regularity. It always moves from the left side (right side) of one PV panel to the right side (left side) with a particular attitude. It then spans the distance between PV panels to reach the left side (right side) of another PV panel before finally moving to the right side (left side) with a different attitude. The PV panel cleaning robot operates in a cyclical manner, moving back and forth to complete its cleaning tasks. This process occurs sequentially through three distinct states: translation, spanning, and translation again. It may be beneficial to define the law of motion of the cleaning robot, which sequentially executes three phases, translation, spanning, and panning, as a work cycle. The start (end) of a work cycle corresponds to the end (start) of

the previous (next) work cycle, while the end of a work cycle marks the commencement of the next one.

In this paper, the relative positions between PV panels are found to be of different sizes based on the actual inspection of desert PV power plants, as illustrated in Figure 2. One of the most complex situations can arise between two neighboring photovoltaic (PV) panels. This scenario involves angular projections on three coordinate planes and translations along three coordinate axes. Specifically, the cleaning robot may rotate by angles ψ , θ , and ϕ around the X, Y, and Z axes of a fixed coordinate system, as well as translate along the X, Y, and Z axes. It is during this relatively complex phase of spanning motion that the non-coplanar trajectory must be analyzed and resolved. A diagram illustrating the pose transformation is presented in Figure 3.



Figure 2. The actual inspection of desert PV power plants.

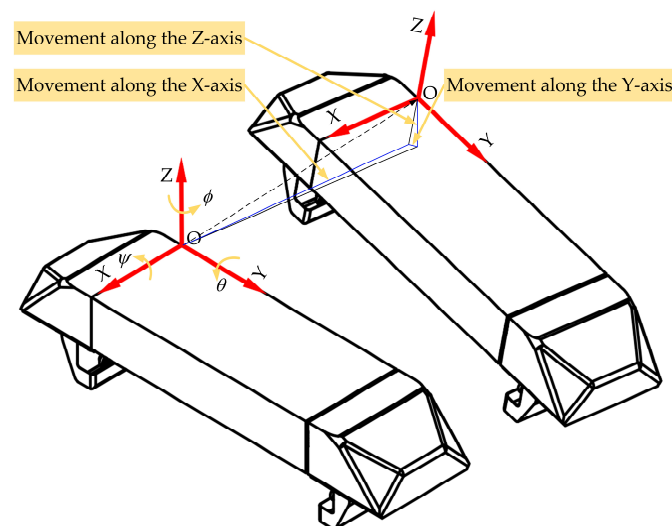


Figure 3. A diagram illustrating the pose transformation.

2.2. Trajectory Model

According to the analysis of the cleaning robot's motion law in Section 2.1, the translational phase at the beginning of one work cycle coincides with the translational phase at the end of another work cycle, thereby forming a closed loop of motion. In order to ensure the integrity of the trajectory, the non-coplanar trajectory during the spanning motion phase is solved by examining the motion law of the PV panel cleaning robot throughout a single work cycle. The simplified model of the cleaning robot's motion is illustrated in Figure 4. The world coordinate system, denoted as $o\text{-}xyz$, and the local coordinate system, designated as $O\text{-}XYZ$, are established. In order to facilitate the representation of the

cleaning robot’s positional state, it is simplified to a straight line in space and aligned with the world coordinate system to establish the initial position (state 0). Consequently, the initial positional matrix of the rigid body is represented as a fourth-order identity matrix, denoted as T_0 . In other words, the cleaning robot moves from State 0 to State 1, then spans the distance between the photovoltaic panels to reach State 2, and finally moves to State 3 to complete a work cycle, as described in detail below.

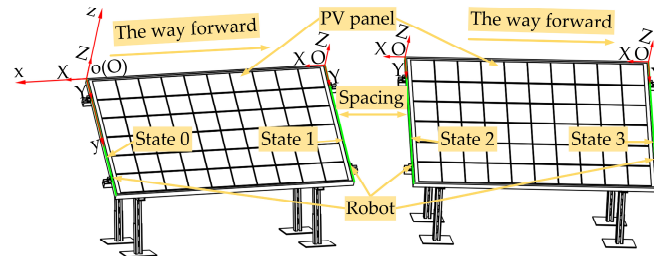


Figure 4. The simplified model of the cleaning robot’s motion.

The robot moves from State 0 to State 1, completing the translation transformation of the initial position. The translation matrix $Trans(x_1, 0, 0)$ at this stage is equal to the transformation matrix 0_1T . The model is established as follows:

$${}^0_1T = Trans(x_1, 0, 0) = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

The robot moves from State 1 to State 2, completing the spanning motion. Since there may be more complex relative positions between the PV panels, a translation matrix $Trans(x_2, y_2, z_2)$ and a rotation matrix $R(\phi, \theta, \psi)$ are generated accordingly during the spanning motion phase. Specific analyses are provided in Appendix A. Therefore, the transformation matrix 1_2T for this phase is as follows:

$${}^1_2T = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi & x_2 \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi & y_2 \\ -s\theta & c\theta s\psi & c\theta c\psi & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

The robot moves from State 2 to State 3, completing a translational transformation of another attitude. The translation matrix $Trans(x_3, y_3, z_3)$ at this stage is equal to the transformation matrix 2_3T . The model is established as follows:

$${}^2_3T = Trans(x_3, y_3, z_3) = \begin{bmatrix} 1 & 0 & 0 & x_3 \\ 0 & 1 & 0 & y_3 \\ 0 & 0 & 1 & z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

The associative Formulations (1)–(3) can represent the motion of the robot during a working cycle. Consequently, the trajectory data of the PV panel cleaning robot during its spanning motion can be described by a set of three-dimensional (3D) point sets, denoted as O_i , and represented as follows:

$$O_i = \begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ z_1 & \dots & z_n \end{bmatrix}, \tag{4}$$

where n denotes the number of columns in the 3D point set and $O_i (i = 1, 2, \dots, n)$ is an $n \times 3$ matrix that stores the 3D data point set, with each column in the matrix representing a spatial data point. Any point in this matrix can be denoted as $O_i = (x_i, y_i, z_i)$.

2.3. Smoothing Processing Model

The trajectory-solving process incorporates data from three phases, with the translational transformation trajectory during a single work cycle being fixed, i.e., the walking trajectory of the cleaning robot along the edge of the PV panel, while the trajectory during the spanning motion stage is non-coplanar. Consequently, abrupt changes in the connection between these trajectories are unavoidable, which can adversely affect the robot's operation. Therefore, it is essential to smooth the trajectories during the spanning motion to achieve the goal of seamlessly splicing the two translational trajectories. To obtain a smooth trajectory for the cleaning robot during the spanning motion phase, this paper employs Bezier curves to smooth non-coplanar trajectories. The general form of a Bezier curve is as follows:

$$B(t) = \sum_{i=0}^n P_i \cdot B_{i,n}(t), \tag{5}$$

where n is the order of the curve, $P_i = (0, 1, 2, \dots, n)$ denotes the position of the $(i + 1)$ -th control point of the Bezier curve, and t is a parameter that varies within the range $[0, 1]$. $B_{i,n}(t)$ is referred to as the Bernstein basis function and is defined as follows:

$$B_{i,n}(t) = C_n^i t^i (1 - t)^{n-i}, \tag{6}$$

where $C_n^i = n! / (i!(n - i))$, $i = 1, 2, \dots, n$, denotes the polynomial coefficients.

Because Bezier curves possess excellent shape control capabilities, they are widely utilized in curve modeling. In practical engineering applications, third-order Bezier curves are the most common and can satisfy the requirements of most projects. Therefore, this paper adopts the third-order Bezier curve for trajectory smoothing. These curves are defined by four control points: $P_{i,0}$, $P_{i,1}$, $P_{i,2}$, and $P_{i,3}$. The parameter equations are as follows:

$$B(t) = (1 - t)^3 P_{i,0} + 3t(1 - t)^2 P_{i,1} + 3t^2(1 - t) P_{i,2} + t^3 P_{i,3}, \tag{7}$$

In order to achieve smooth splicing of the two translational trajectories, the constraint equations for control points $P_{i,1}$ and $P_{i,2}$ are as follows:

$$\begin{cases} P_{i,1} = P_{i,0} + r_{i,1}n_1 \\ P_{i,2} = P_{i,3} - r_{i,2}n_2' \end{cases} \tag{8}$$

where $i = 1, 2$ denotes Bezier curve 1 and Bezier curve 2, $r_{i,1}$ and $r_{i,2}$ are the scale parameters of the control points, and n_1 and n_2 are the tangent vectors at points $P_{i,0}$ and $P_{i,3}$, respectively. The association of Equations (7) and (8) can be established as follows:

$$B^i(t) = [(1 - t)^3 + 3t(1 - t)^2]P_{i,0} + [3t^2(1 - t) + t^3]P_{i,3} + 3r_{i,1}n_1(1 - t)^2 + 3r_{i,2}n_2t^2(1 - t), \tag{9}$$

It unfolds in the following form:

$$\begin{aligned} B_x^i(t) &= [(1 - t)^3 + 3t(1 - t)^2]P_{i,0}(x) + [3t^2(1 - t) + t^3]P_{i,3}(x) + 3r_{i,1}n_1(x)(1 - t)^2 - 3r_{i,2}n_2(x)t^2(1 - t) \\ B_y^i(t) &= [(1 - t)^3 + 3t(1 - t)^2]P_{i,0}(y) + [3t^2(1 - t) + t^3]P_{i,3}(y) + 3r_{i,1}n_1(y)(1 - t)^2 - 3r_{i,2}n_2(y)t^2(1 - t) , \\ B_z^i(t) &= [(1 - t)^3 + 3t(1 - t)^2]P_{i,0}(z) + [3t^2(1 - t) + t^3]P_{i,3}(z) + 3r_{i,1}n_1(z)(1 - t)^2 - 3r_{i,2}n_2(z)t^2(1 - t) \end{aligned} \tag{10}$$

Here, Equation (10) is the Bezier curve equation with two scaling parameters, where $t \in [0, 1]$; by using this equation, the trajectory data points of the spanning motion phase are smoothed.

2.4. Modelling Optimisation Analysis

2.4.1. Optimization Principles and Processes

The cleaning robot traveling mechanism moves along the upper and lower edges of the photovoltaic panels and has high requirements for stability during the passage through the non-coplanar orbit. Therefore, the curvature change in the trajectory should be as small as possible, and the designed track should tend to be smooth. In order to determine an appropriate scale parameter, this paper proposes using the average distance as the objective function. It identifies a non-coplanar trajectory that satisfies the requirements by incorporating additional constraints. The specific optimization process is illustrated in Figure 5.

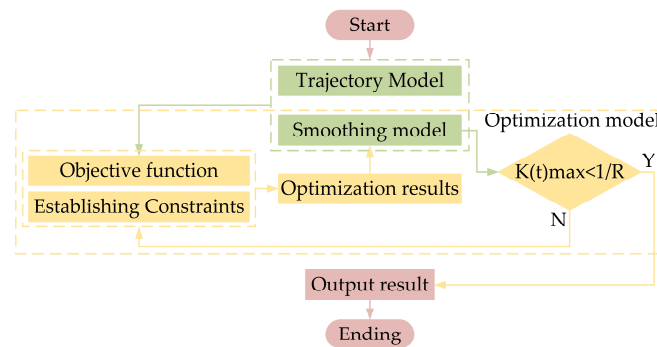


Figure 5. The specific optimization process.

2.4.2. Optimization Method

This study identifies a non-coplanar trajectory that meets specific constraints by optimizing the distance-averaged objective function and completing the scaling parameter optimization using the presented nonlinear planning method.

1. Optimization function.

The FMINCON function in MATLAB is a widely utilized optimization tool designed to find the minimum of a constrained nonlinear multivariate function. To achieve faster convergence, enhanced computational efficiency, and improved boundary search capabilities, this paper employs the Sequential Quadratic Programming (SQP) algorithm in conjunction with the FMINCON function. The formulation of the FMINCON function is as follows:

$$r = \text{Fmincon}(f, r, A, b, Aeq, beq, lb, ub, @(r) \text{nonlcon}(r, n_1, n_2), options), \quad (11)$$

The parameter f represents the objective function, while r denotes the initial scaling parameter, which includes two parameter values. The matrices A and b define the linear inequality constraints, expressed as $A \times r < b$. The matrices Aeq and beq represent the equation constraints, formulated as $Aeq \times r < beq$. The variables lb and ub indicate the lower and upper bound constraints, respectively, such that $lb < r < ub$. Additionally, the function nonlcon is a nonlinear constraint function that can return two output values: the nonlinear inequality $C(r) \leq 0$ and the equation $C(r) = 0$.

2. Establishment of an objective function.

The optimization objective is defined as the average value of the minimum distance between the Bezier curve and the data points in the point set during the spanning motion phase. Optimization of the Bezier curve is accomplished by minimizing this average value using the FMINCON function. According to the matrix of Equation (4) presented in Section 2.2, any data point within the spanning phase can be represented as O_i . Furthermore, based on the Bezier curve Equation (9) outlined in Section 2.3, any point on the curve can be denoted as $B(t_i)$. Then, for each point in the given point set, the Euclidean distances

between that point and all points on the Bezier curve are calculated. Simultaneously, the minimum distances are recorded, and the average of these minimum distances is utilized as the optimization objective for the proportional parameter optimization. The objective function is as follows:

$$D(r) = \frac{1}{n} \sum_{i=1}^n \min_j \|o_i - B(t_j)\|, \tag{12}$$

where n represents the size of the data point set and $\|\cdot\|$ denotes the Euclidean norm. The scaling parameter that satisfies the optimization condition is determined by minimizing the mean value $D(r)$.

3. Setting constraints.

By establishing optimization conditions, the orientation of unknown control points can be effectively constrained, thereby ensuring the desired shape of the Bezier curve. Based on the requirements of trajectory design, the optimization conditions include orientation optimization and curvature optimization. Specifically, orientation optimization defines the precise direction and position of the control points in space, while curve optimization determines the degree of bending of the Bezier curve.

The orientation optimization condition is established to ensure that the Bezier curve smoothly connects the direction vectors of the two translation phases. Specifically, the Bezier curve is tangent to the direction vectors n_1 and n_2 of the two translation phases at the starting point $P_{i,0}$ and the ending point $P_{i,3}$, respectively. Therefore, the control point $P_{i,1}$ must be constrained to the line defined by the direction vector n_1 , while the control point $P_{i,2}$ must be constrained to the line defined by the direction vector n_2 . The relationship between these constraints is detailed in Section 2.3, specifically in Equation (8). To ensure the validity of the control point solution, the range of values for the scale parameter is restricted by the inequality $lb < r < ub$.

The curvature optimization conditions are established to ensure the overall curvature degree of the Bezier curve line shape. Based on the radius R of the cleaning robot’s walking wheel, the curvature threshold is defined as $1/R$. While ensuring that the maximum curvature of the Bezier curve remains below this threshold, it is also essential to minimize the curvature as much as possible. According to Equation (9), the first-order and second-order derivatives of the Bezier curve can be derived as follows:

$$\frac{dB^i}{dt} = 6t(1-t)(P_{i,3} - P_{i,0}) + 3r_{i,1}n_1(1-t)(1-3t) - 3r_{i,2}n_2t(2-3t), \tag{13}$$

$$\frac{d^2B^i}{dt^2} = 6(1-2t)(P_{i,3} - P_{i,0}) + 6r_{i,1}n_1(3t-2) - 6r_{i,2}n_2t(1-3t), \tag{14}$$

The formula for curvature is as follows:

$$K(t)_{max} = \frac{\left\| \frac{dB^i}{dt} \times \frac{d^2B^i}{dt^2} \right\|}{\left\| \frac{dB^i}{dt} \right\|^3}, \tag{15}$$

$$C = K(t)_{max} - \frac{1}{R} \leq 0, \tag{16}$$

where $K(t)_{max}$ is the maximum curvature of the Bezier curve and C represents the inequality constraint in the FMINCON function.

2.4.3. Optimization Model

By compiling the aforementioned formulations, the final optimization problem for two non-coplanar trajectories is expressed as follows:

$$\begin{cases} \min_r (\frac{1}{n} \sum_{i=1}^n \min_j \|o_i - B(t_j)\|) \\ \begin{cases} P_{i,1} = P_{i,0} + r_{i,1}n_1 \\ P_{i,2} = P_{i,3} - r_{i,2}n_2 \end{cases} \\ \text{s.t. } lb_i \leq r_i \leq ub_i \\ \frac{\| \frac{dB^i}{dt} \times \frac{d^2B^i}{dt^2} \|}{\| \frac{dB^i}{dt} \|^3} - \frac{1}{R} \leq 0 \end{cases}, \quad (17)$$

The non-coplanar trajectory is optimized using Equation (17) to determine the scale parameter that meets specific constraints. This scale parameter is then incorporated into the smoothing processing model to derive a mathematical model of the non-coplanar trajectory that fulfills the specified requirements.

3. Results and Discussion

Based on the field study, the PV panel with the most complicated relative position is used as a case study for the design and optimization of non-coplanar trajectories. First, the necessary data points for the two PV panels are obtained using high-precision measuring instruments. Next, the design and optimization of the non-coplanar trajectory are conducted using the method described in the previous section. Finally, the effects of the optimized data and the simulation results are evaluated. In addition, this study used a computer running Windows 11 (Microsoft Corporation, Redmond, WA, USA) with a 13th generation Intel(R) Core(TM) i5-13600KF 3.5 GH processor and 32 GB of baseband RAM (Intel Corporation, Santa Clara, CA, USA).

3.1. Data Collection

A world coordinate system (fixed to the PV panels) and a local coordinate system (fixed to the cleaning robot) are established by collecting the coordinates of the PV panel vertices at complex relative positions. In order to simplify the calculation process, the data for the PV panels are initialized, and the local coordinate system is aligned with the world coordinate system, as illustrated in Figure 4. Consequently, the coordinate information for two PV panels with complex relative positions is presented in Table 1.

Table 1. The coordinate information for two PV panels with complex relative positions.

The Vertex	PV Panel 1	PV Panel 2
The Vertex 1	(0, 0, 0)	(−3712.88, 219.84, 116.35)
The Vertex 2	(−1600, 0, 0)	(−2126.71, 17.93, 58.86)
The Vertex 3	(−1600, 1000, 0)	(−2005.03, 1004.59, −49.23)

Based on the available data points, the trajectory of the cleaning robot during a single work cycle can be derived using the trajectory model outlined in Section 2.2. The known data points are incorporated into Equations (1) through (3), and the motion trajectories of the cleaning robot during phases A (translation), B (spanning motion), and C (translation) are recorded and plotted using MATLAB software (Matlab R2022a version), as illustrated in Figure 6. Trajectory 1 and Trajectory 2 represent the upper and lower trajectory data, respectively. The figure illustrates that the trajectory of Stage B as well as the trajectories of Stages A and C exhibit issues with trajectory mutation at their connections. Given the specific structure of the cleaning robot and the need for stable robot operation, it is essential to smooth the non-coplanar trajectories during the spanning motion phase.

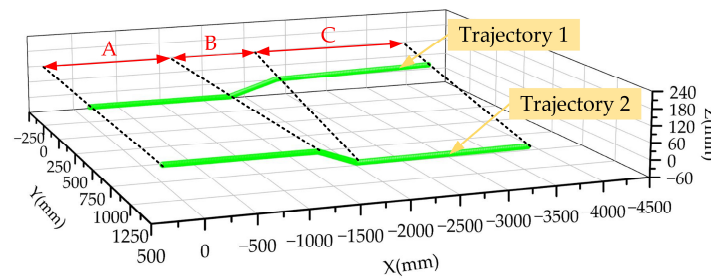


Figure 6. The motion trajectories of the cleaning robot during phases A (translation), B (spanning motion), and C (translation).

3.2. Smoothing and Optimization

3.2.1. Smoothing of Non-Coplanar Trajectories

First, based on the data presented in Table 1, the vertex 2 of PV panel 1 and PV panel 2 are designated as the start and end points of Bezier curve 1, corresponding to control points $P_{1,0}$ and $P_{1,3}$, respectively. Similarly, the vertex 3 of PV panel 1 and PV panel 2 serve as the start and end points of Bezier curve 2, represented by control points $P_{2,0}$ and $P_{2,3}$, respectively. Then, the non-coplanar trajectory design is executed using the Bezier curve formula with two scaling parameters. This ensures that the trajectory of Stage B is tangent at the node positions connecting the two PV panels (control points $P_{i,0}$ and $P_{i,3}$). In other words, the smoothing of the non-coplanar trajectory is achieved as described by Equation (10). The effects of the smoothing process of Stage B during one working cycle are illustrated in Figure 7.

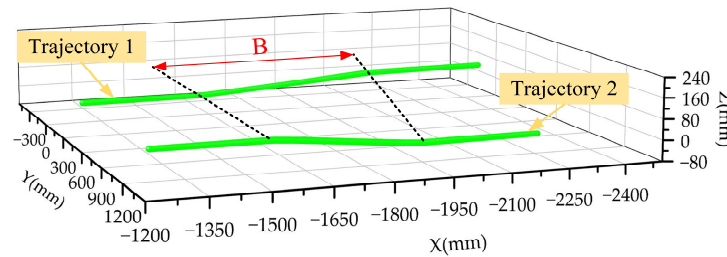


Figure 7. The effects of the smoothing process of Stage B during one working cycle.

3.2.2. Optimization of Non-Coplanar Trajectories

The designed Bezier curve is optimized according to the optimization model presented in Section 2.4. The optimization function employs the FMINCON function in MATLAB, utilizing the Sequential Quadratic Programming (SQP) algorithm to enhance convergence speed. The optimization process is ultimately completed after nine iterations, with a total simulation time of approximately 0.08 s. The optimization objective is defined as the average of the minimum distances between the Bezier curve and the data points in the point set during the spanning motion phase. Additionally, azimuthal and curvature constraints are established to facilitate the determination of the scaling parameter, thereby achieving the optimization of the Bezier curve. The final optimization problem can be described as follows: optimize the Bezier curve by Equation (17) to obtain the scaling parameters that satisfy the constraints. The optimized proportional parameters are illustrated in Table 2.

Table 2. The optimized proportional parameters.

Bezier Curve	Scale Parameter	Unoptimized	Optimized
Curve 1	r_{11}	0.040 mm	0.025 mm
	r_{12}	0.033 mm	0.020 mm
Curve 2	r_{21}	0.033 mm	0.020 mm
	r_{22}	0.029 mm	0.017 mm

The optimized Bezier curve equation can be derived by substituting the optimized scale parameters from Table 2 into Equation (10). The tangent vector n_1 is calculated by taking the difference between vertex 2 and vertex 1 of PV panel 1, as illustrated in Table 1. Similarly, n_2 is determined by calculating the difference between vertex 1 and vertex 2 of PV panel 2 in Table 1. Specifically, $n_1 = (-1600, 0, 0)$ and $n_2 = (-1586.17, 201.91, 57.49)$. The vectors n_1 and n_2 represent the tangent vectors at the starting and ending points of the Bezier curves, respectively. The tangent vectors of Bezier curve 1 and Bezier curve 2 share the same direction but are located at different positions. Consequently, the parametric equations for Bezier curve 1 and Bezier curve 2 can be derived.

All known data in Stage B are plotted in three-dimensional space, and the optimization effects of the non-coplanar trajectory are illustrated in Figure 8. Bezier curve 1 represents the trajectory of Trajectory 1 during the spanning motion phase, while Bezier curve 2 represents the trajectory of Trajectory 2 during the same phase. P_1 and P_2 represent the control points $P_{1,1}$ and $P_{1,2}$, and $P_{2,1}$ and $P_{2,2}$, before and after the optimization, respectively. The initial trajectory refers to the actual path taken by the cleaning robot during Stage B. The smooth trajectory is produced by the smoothing processing model, while the optimized trajectory is generated by the optimization model. The optimization effects are illustrated in Figure 8a,b. The optimized Bezier curves are more closely aligned with the known data points.

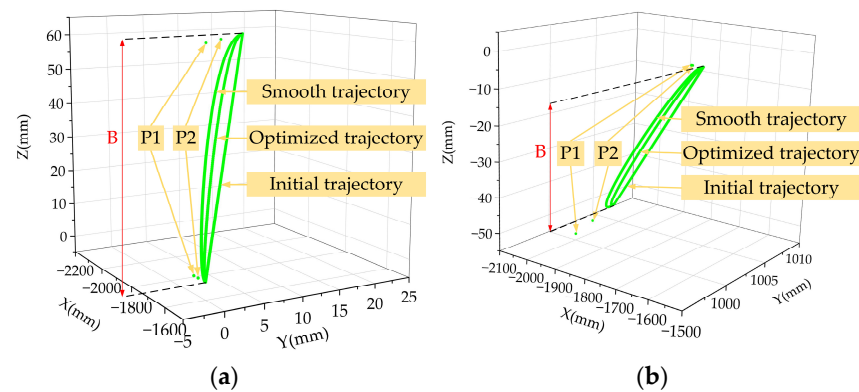


Figure 8. The optimization effects: (a) optimization results of Bezier curve 1; (b) optimization results of Bezier curve 2.

3.3. Modeling and Simulation of Non-Coplanar Orbits

According to the Bezier curve equations, both before and after optimization, the 3D models of the photovoltaic (PV) panel and the non-coplanar orbit were created and imported into ADAMS for simulation to verify the passing ability. Figure 9A illustrates the non-coplanar track 3D model that connects the PV panels, while Figure 9B depicts the simulation of the spanning motion phase of the cleaning robot, which includes three scenarios: (a), (b), and (c). By observing the simulation animation, it is evident that the cleaning robot spanning the non-coplanar track does not experience jamming or falling off the track. Furthermore, it successfully executes the spanning motion in a complete and smooth manner.

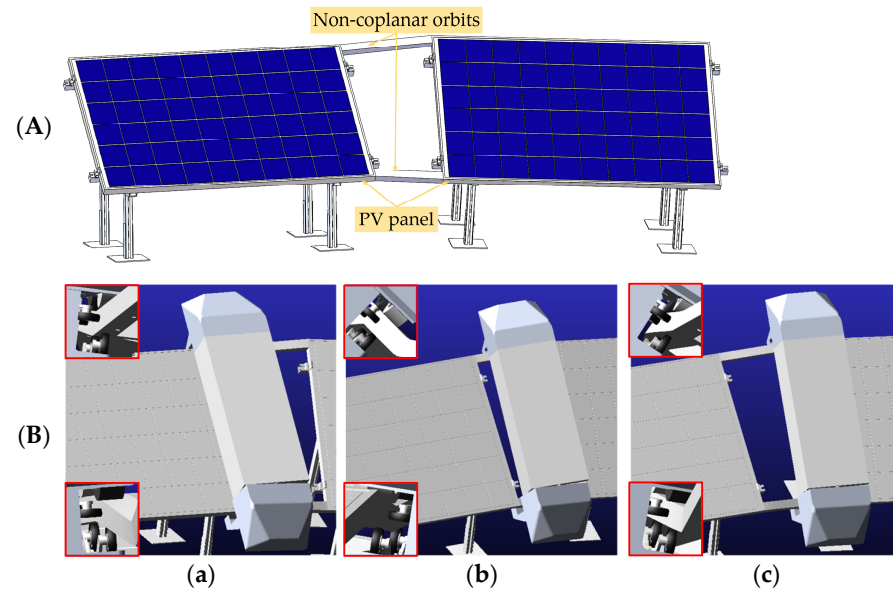


Figure 9. Modeling and simulation: (A) the non-coplanar track 3D model that connects the PV panels; (B) the spanning motion phase: (a) incompletely driving into the orbit; (b) completely driving into the orbit; (c) incompletely driving out of the orbit.

3.4. Effectiveness Evaluation and Analysis

3.4.1. Standard Deviation of Distance

After determining the Bezier curve, the optimization effect is evaluated by calculating the standard deviation of the distances between the curve and the data points during the spanning motion phase, both before and after optimization, using the following formula:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(d_i - \frac{1}{n} \sum_{i=1}^n d_i \right)^2}, \tag{18}$$

where d_i represents the distance between the Bezier curve and the known data points, and σ denotes the standard deviation of the distances, with a smaller value of σ indicating a better fitting effect.

The standard deviation of the distances for the Bezier curves, both before and after optimization, is calculated using Equation (18), as illustrated in Table 3. The results indicate that the optimized Bezier curves 1 and 2 are closer to the data points of Trajectory 1 and Trajectory 2 in Stage B, with improvements of 35.63% and 40.57%, respectively. This indicates that the optimized Bezier curves more accurately represent the actual trajectory data.

Table 3. The standard deviation of the distance for The Bezier curves.

Bezier Curve	Standard Deviation	Optimization Percentage
Unoptimized curve 1	0.8197	
Optimized curve 1	0.5276	35.63%
Unoptimized curve 2	0.7484	
Optimized curve 2	0.4448	40.57%

3.4.2. Curvature

The curvature of the pre-optimized and post-optimized Bezier curves is presented in Table 4 below. The results indicate that the maximum curvature of the optimized Bezier curve exceeds that of the pre-optimized Bezier curve while still satisfying the curvature threshold. This suggests that the optimized Bezier curve aligns more closely with the actual trajectory data. Additionally, the minimum curvature of the optimized Bezier curve is less

than that of the pre-optimized Bezier curve, indicating that the optimized Bezier curve is smoother.

Table 4. The curvature of the pre-optimized and post-optimized Bezier curves.

Bezier Curve	Maximum Curvature	Minimum Curvature	Curvature Threshold
Unoptimized curve 1	4.8696×10^{-4}	6.6388×10^{-8}	1.6×10^{-2}
Optimized curve 1	1.0472×10^{-3}	3.5739×10^{-8}	
Unoptimized curve 2	5.7751×10^{-4}	1.0840×10^{-7}	
Optimized curve 2	1.4655×10^{-3}	5.6493×10^{-8}	

3.4.3. Simulation Data

Dynamics simulations were conducted using ADAMS to measure the x, y, and z coordinates of the center of mass of the four walking wheels of the PV panel cleaning robot, both for the pre-optimized and optimized orbits. As illustrated in Figure 10, the trajectory component plots for the centers of mass of the four walking wheels during Stage B are presented. From the figure, it is evident that the trajectory data trends of the four walking wheels, both before and after optimization, are nearly identical, with only a slight variation observed during a specific time segment. This indicates that the cleaning robot can operate smoothly on the non-coplanar orbit, thereby validating the effectiveness of the orbital design.

In order to more intuitively illustrate the differences between the pre-optimization and post-optimization B phases, the simulation error curves for the four walking wheels are presented in Figure 11. Error 1 denotes the simulation error prior to optimization, while error 2 indicates the simulation error following optimization. From the figure, it is evident that there are peaks and valleys in both the pre- and post-optimization simulation errors. Additionally, there is an irrational optimization of the simulation errors for wheels 1 and 3 during a specific time period. This phenomenon is attributed to the deflection of the robot's position caused by the differential movement of the traveling wheels during the robot's spanning motion. In addition, it is evident that the simulation errors for the four wheels achieve optimal performance at 15 s, 57 s, 50 s, and 59 s, respectively, with reductions of 8.79 mm, 23.78 mm, 10.11 mm, and 14.97 mm. The data and simulation results confirm that the theoretical model design and optimization are reasonable to a significant extent.

In this study, we use Bezier curves with scaling parameters to design non-coplanar trajectories. This approach improves the adaptability and scalability of photovoltaic (PV) panel cleaning robots in variable installation environments and relative positions. Compared to traditional Bezier curves, our approach shows significant advantages in dealing with the dynamic changes of PV panels. Although B-spline curves offer a high degree of flexibility in trajectory design, they can be computationally more complex and it can be difficult for the method to accurately model the desired shape in 3D space, as pointed out by Wang et al. [16]. To overcome these limitations, our approach reduces the computational burden by simplifying the mathematical model, thereby increasing its feasibility in practical applications. Although the NURBS curves used by Li et al. [17] in industrial robot trajectory planning provide a powerful tool for Cartesian trajectory planning, the method has a complicated decomposition process, including the calculation of tangent and curvature, which can lead to a cumbersome operation. In contrast, our proposed method exploits the simplicity and controllability of Bezier curves, which are widely recognized properties in path planning research [18–22]. Furthermore, our study shows that our proposed method has greater adaptability to photovoltaic (PV) panels in unstructured scenarios compared to existing studies [7,11,13,16], a finding that provides strong support for cleaning robots operating on PV panels with inconsistent layouts.

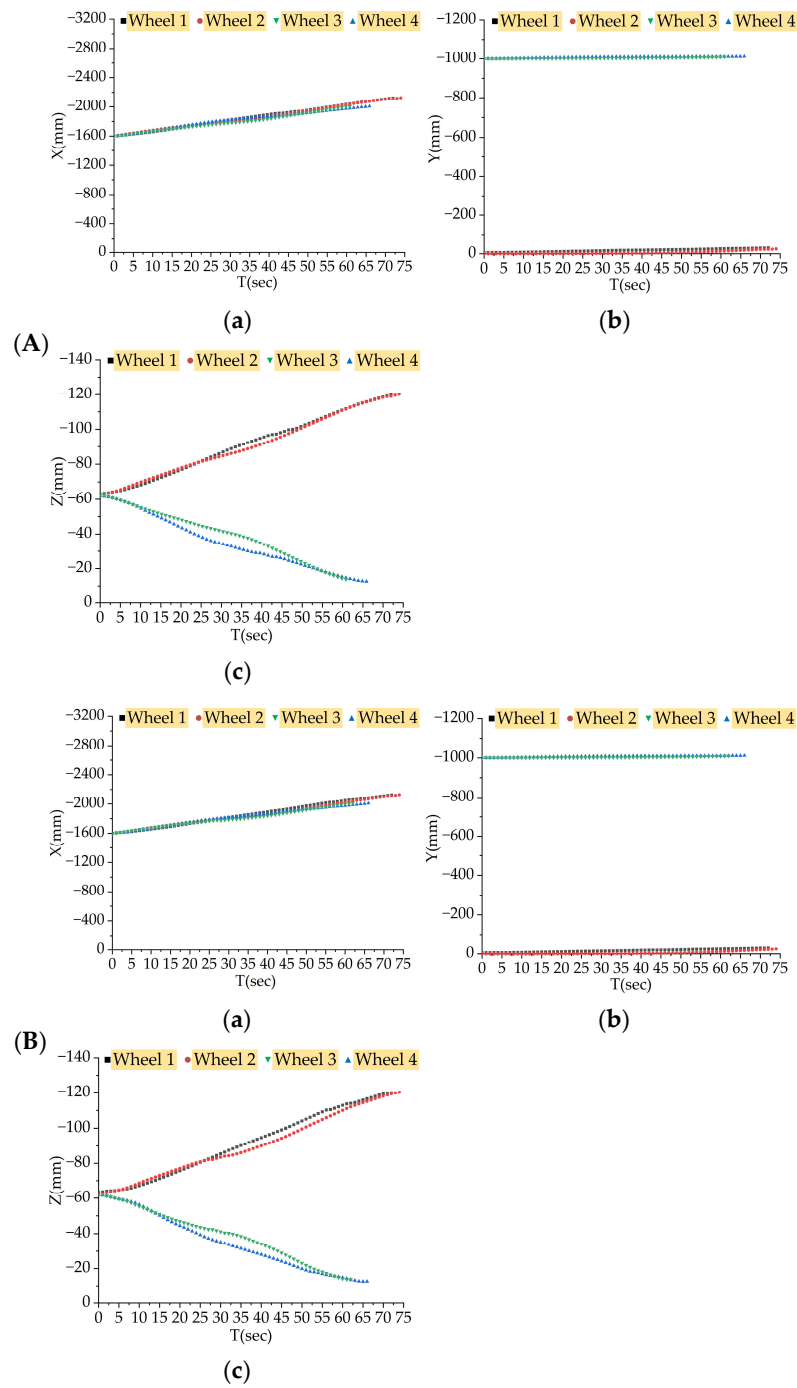


Figure 10. The trajectory component plots for the centers of mass of the four walking wheels during Stage B: (A) unoptimized walking wheel trajectory; (B) optimized walking wheel trajectory: (a) x-values for the 4 walking wheels, (b) y-values for the 4 walking wheels, and (c) z-values for the 4 walking wheels.

The method proposed in this paper not only improves the flexibility and adaptability of trajectory design for photovoltaic (PV) panel cleaning robots but also reduces the complexity of practical applications by simplifying the computational process. This provides an efficient and reliable solution for the automated operation and maintenance of PV power plants.

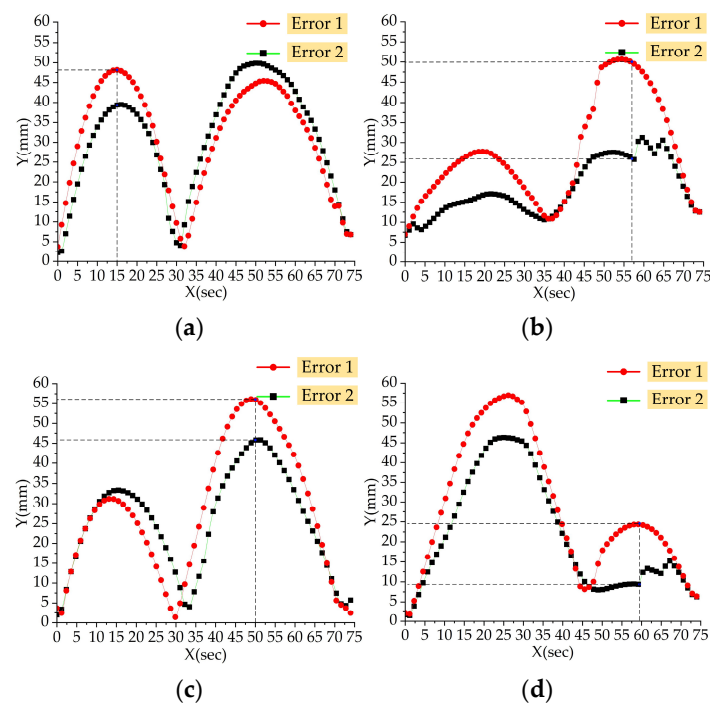


Figure 11. The simulation error curves for the four walking wheels: (a) wheel 1 error; (b) wheel 2 error; (c) wheel 3 error; (d) wheel 4 error.

3.5. Shortcomings and Prospects

In this paper, significant progress has been made in the field of non-coplanar orbit research; however, there remains considerable room for improvement. The primary limitations are as follows: (1) This study uses a single-segment Bezier curve in the non-coplanar trajectory design, which may result in unsatisfactory trajectory errors during simulation. (2) This study focuses exclusively on one approach to non-coplanar orbit design and optimization. A comparative analysis with other state-of-the-art techniques would enhance the credibility of this article. (3) The current study is limited to the simulation stage and has not yet been validated in practice.

In future work, segmented or higher-order Bezier curves may be employed to design non-coplanar trajectories. Additionally, other methodologies can be explored, and it is essential to consider integrating Analytic Hierarchy Process (AHP) techniques into the design process to enhance efficiency. Furthermore, advanced methods for aeronautically assisted rendezvous-intercept trajectory optimization can be utilized to improve the optimization of non-coplanar orbits. It is also important to construct photovoltaic (PV) panels and non-coplanar orbits in real environments in conjunction with additional field tests and environmental data to experimentally validate the effectiveness and practicality of the proposed methodology.

4. Conclusions

This study explores and solves the problem of non-coplanar trajectory design and optimization for a photovoltaic panel cleaning robot for the first time. Based on the analyzed motion laws of the cleaning robot, this study established a trajectory model to obtain trajectory data. To ensure the robot's stability along its path, Bezier curves were employed as the smoothing model for designing the non-coplanar trajectory. Recognizing the significant impact of the scale parameter on the trajectory's shape, a nonlinear planning method based on the Sequential Quadratic Programming (SQP) algorithm was utilized to optimize the trajectory. The optimization results indicate that the performance of Bezier curves 1 and 2 improved by 35.63% and 40.57%, respectively. Furthermore, the optimization effects are corroborated by ADAMS simulation data, which reveal that the error values of the

four wheels of the cleaning robot were reduced by a maximum of 8.79 mm, 23.78 mm, 10.11 mm, and 14.97 mm, respectively. The optimized robot demonstrated no jamming or derailment during the spanning motion phase, and its stability was significantly enhanced. This research provides a foundation for the photovoltaic panel cleaning robot to perform large-area-spanning cleaning tasks, ensuring the automated cleaning of large-scale photovoltaic power stations, which is crucial for environmental protection and improving power generation efficiency.

Author Contributions: Conceptualization, Y.Q.; Methodology, Y.Q.; Software, B.X.; Formal analysis, B.X.; Investigation, Y.Z. and Y.Q.; Data curation, Y.Z.; Writing—original draft, Y.Z.; Writing—review & editing, Y.Q. and B.X.; Visualization, Y.Z. and B.X.; Supervision, Y.Q.; Project administration, Y.Q.; Funding acquisition, Y.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the “Natural Science Key Project of the Scientific Research Program of the Tianjin Municipal Education Commission under Grant 2022ZD026 and 2022ZD032”; in part by the “Tianjin Key Research and Development Program Projects under Grant 23YFYSHZ00280”, led by the corresponding author, Yuming Qi.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the reviewers for their constructive comments on the paper.

Conflicts of Interest: The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

The translation matrix for the robot moving from State 1 to State 2 is as follows:

$$Trans(x_2, y_2, z_2) = \begin{bmatrix} 1 & 0 & 0 & x_2 \\ 0 & 1 & 0 & y_2 \\ 0 & 0 & 1 & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (A1)$$

The rotation matrix of the robot moving from State 1 to State 2 is as follows:

$$R(\phi, \theta, \psi) = Rot(z, \phi)Rot(y, \theta)Rot(x, \psi), \quad (A2)$$

where $Rot(x, \psi)$ denotes the rotation of the cleaning robot around the X axis of the local coordinate system by an angle ψ . The specific equation is as follows:

$$Rot(x, \psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\psi & -s\psi & 0 \\ 0 & s\psi & c\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (A3)$$

where $Rot(y, \theta)$ denotes the rotation of the cleaning robot around the Y axis of the local coordinate system by an angle θ . The specific equation is as follows:

$$Rot(y, \theta) = \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (A4)$$

where $Rot(z, \phi)$ denotes the rotation of the cleaning robot around the Z axis of the local coordinate system by an angle ϕ . The specific equation is as follows:

$$Rot(z, \phi) = \begin{bmatrix} c\phi & -s\phi & 0 & 0 \\ s\phi & c\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A5}$$

The transformation matrix for the robot to move from State 1 to State 2 is as follows:

$$\begin{aligned} {}^1_2T &= Trans(x_2, y_2, z_2)R(\phi, \theta, \psi) \\ &= Trans(x_2, y_2, z_2)Rot(z, \phi)Rot(y, \theta)Rot(x, \psi) \\ &= \begin{bmatrix} 1 & 0 & 0 & x_2 \\ 0 & 1 & 0 & y_2 \\ 0 & 0 & 1 & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\phi & -s\phi & 0 & 0 \\ s\phi & c\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\psi & -s\psi & 0 \\ 0 & s\psi & c\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c\phi & -s\phi & 0 & x_2 \\ s\phi & c\phi & 0 & y_2 \\ 0 & 0 & 1 & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\psi & -s\psi & 0 \\ 0 & s\psi & c\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c\phi c\theta & -s\phi & c\phi s\theta & x_2 \\ s\phi c\theta & c\phi & s\phi s\theta & y_2 \\ -s\theta & c\theta & 1 & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\psi & -s\psi & 0 \\ 0 & s\psi & c\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi & x_2 \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi & y_2 \\ -s\theta & c\theta s\psi & c\theta c\psi & z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{A6}$$

References

1. International Energy Agency. *Overcoming the Energy Trilemma: Secure and Inclusive Transitions*; IEA: Paris, France, 2023. Available online: <https://www.iea.org/reports/overcoming-the-energy-trilemma-secure-and-inclusive-transitions> (accessed on 19 June 2024).
2. International Energy Agency. *Enhancing China's ETS for Carbon Neutrality: Introducing Auctioning*; IEA: Paris, France, 2024. Available online: <https://www.iea.org/reports/enhancing-chinas-ets-for-carbon-neutrality-introducing-auctioning> (accessed on 19 June 2024).
3. Walz, K.A.; Hoege, T.D.; Duensing, J.W.; Zeltner, W.A.; Anderson, M.A. Field tests of a self-sintering, anti-soiling, self-cleaning, nanoporous metal oxide, transparent thin film coating for solar photovoltaic modules. *Sol. Energy Mater. Sol. Cells* **2023**, *262*, 112560. [CrossRef]
4. Hosseini, F.; Sheikholeslami, M. Effects of self-cleaning technique and nanofluid cooling on performance of photovoltaic solar unit employing sinusoidal surfaces. *Appl. Therm. Eng.* **2024**, *240*, 122223. [CrossRef]
5. Hariri, N. A novel dust mitigation technology solution of a self-cleaning method for a PV module capable of harnessing reject heat using shape memory alloy. *Case Stud. Therm. Eng.* **2022**, *32*, 101894. [CrossRef]
6. Yang, J.; Zhao, X.; Gao, Y.; Guo, R.; Zhao, J. Research on Mechanism Design and Kinematic Characteristics of Self-Propelled Photovoltaic Cleaning Robot. *Appl. Sci.* **2023**, *13*, 6967. [CrossRef]
7. Antonelli, M.G.; Zobel, P.B.; De Marcellis, A.; Palange, E. Autonomous robot for cleaning photovoltaic panels in desert zones. *Mechatronics* **2020**, *68*, 102372. [CrossRef]
8. Fan, S.; Liang, W.; Wang, G.; Zhang, Y.; Cao, S. A novel water-free cleaning robot for dust removal from distributed photovoltaic (PV) in water-scarce areas. *Sol. Energy* **2022**, *241*, 553–563. [CrossRef]
9. Gekko Solar Robot. Available online: <https://serbot.ch/en/solar-panels-cleaning/gekko-solar-robot> (accessed on 19 August 2024).
10. Rehman, S.; Mohandes, M.A.; Hussein, A.E.; Alhems, L.M.; Al-Shaikhi, A. Cleaning of Photovoltaic Panels Utilizing the Downward Thrust of a Drone. *Energies* **2022**, *15*, 8159. [CrossRef]
11. Figgis, B.; Bermudez, V.; Garcia, J.L. PV module vibration by robotic cleaning. *Sol. Energy* **2023**, *250*, 168–172. [CrossRef]
12. Cai, S.; Bao, G.; Ma, X.; Wu, W.; Bian, G.B.; Rodrigues, J.J.; de Albuquerque, V.H.C. Parameters optimization of the dust absorbing structure for photovoltaic panel cleaning robot based on orthogonal experiment method. *J. Clean. Prod.* **2019**, *217*, 724–731. [CrossRef]

13. Menemmeche, H.; Abderrezek, M.; Ahriche, A. Design and development of cleaning solar plants robots. *Energy Sources Part A Recovery Util. Environ. Eff.* **2023**, *45*, 7031–7049. [[CrossRef](#)]
14. Yan, S.; Jian, Y.; Xu, L. Research on Design of Intelligent Cleaning Robot for Solar Panel. 2020. Available online: <https://aisel.aisnet.org/iceb2020/22> (accessed on 1 July 2024).
15. Khadka, N.; Bista, A.; Adhikari, B.; Shrestha, A.; Bista, D. Smart solar photovoltaic panel cleaning system. In *IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2020. [[CrossRef](#)]
16. Wang, L.; Guo, Y. Speed adaptive robot trajectory generation based on derivative property of b-spline curve. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1905–1911. [[CrossRef](#)]
17. Li, X.; Zhao, H.; He, X.; Ding, H. A novel cartesian trajectory planning method by using triple NURBS curves for industrial robots. *Robot. Comput. Integr. Manuf.* **2023**, *83*, 102576. [[CrossRef](#)]
18. Zhang, Z.; Liu, X.; Feng, B. Research on obstacle avoidance path planning of UAV in complex environments based on improved Bézier curve. *Sci. Rep.* **2023**, *13*, 16453. [[CrossRef](#)] [[PubMed](#)]
19. Bulut, V. Path planning of mobile robots in dynamic environment based on analytic geometry and cubic Bézier curve with three shape parameters. *Expert Syst. Appl.* **2023**, *233*, 120942. [[CrossRef](#)]
20. Vinayak, A.; Zakaria, M.A.; Baarath, K.; Peeie, M.H.; Ishak, M.I.A. Novel Triangular-Based Estimation Technique for Bezier Curve Control Points Generation on Autonomous Vehicle Path Planning at the Roundabout Intersection. *J. Intell. Robot. Syst.* **2023**, *109*, 89. [[CrossRef](#)]
21. Liu, X.; Nie, H.; Li, D.; He, Y.; Ang, M.H. High-Fidelity and Curvature-Continuous Path Smoothing with Quadratic Bézier Curve. *IEEE Trans. Intell. Veh.* **2024**, *9*, 3796–3810. [[CrossRef](#)]
22. Yan, J.; Deng, T.; Xu, B. Three dimensional path planning for flying car based on improved A* algorithm and Bezier curve fusion. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2024**, 09544070241227094. [[CrossRef](#)]
23. Han, H.; Qiao, D.; Chen, H. Optimization of aeroassisted rendezvous and interception trajectories between non-coplanar elliptical orbits. *Acta Astronaut.* **2019**, *163*, 190–200. [[CrossRef](#)]
24. Villanueva, P.; Bona, S.; Lostado-Lorza, R.; Veiga, F. Morphological Design of a Bicycle Propulsion Component Using the Hierarchical Analysis Process (AHP). *Appl. Sci.* **2023**, *13*, 7792. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.