actuators

MDPI

Article

Trajectory Planning and Performance Atlases of a New Omnidirectional Conveyor

Zhuo Zhang [1], Tianyu Sun [1], Zexing Wang [1] and Xuping Zhang [2,*]

[1] School of Modern Post, Xi'an University of Posts and Telecommunications, Xi'an 710061, China; zhangzhuozz123@126.com (Z.Z.); tysun@stu.xupt.edu.cn (T.S.); zxwang@stu.xupt.edu.cn (Z.W.)
[2] Department of Mechanical and Production Engineering, Aarhus University, 8000 Aarhus, Denmark
* Correspondence: xuzh@mpe.au.dk

Abstract: This paper proposes an omnidirectional conveyor as a novel alternative to existing omnidirectional conveyors. With a symmetric and compact layout, this new structure ensures consistent kinematics and enhanced flexibility in trajectory planning. The kinematic model of the proposed omnidirectional conveyor is developed and verified through simulation in CoppeliaSim. Four typical classes of trajectories are generated and verified in the simulation environment. Using PID control, the actual trajectories of a package on the conveyor closely match the desired trajectories. In addition, this paper outlines the workspace and corresponding wheel patterns for the conveyor, demonstrating how different supported wheel patterns emerge when packages move across various areas of the conveyor. The discussion extends to fault tolerance and obstacle avoidance, examining the workspace and wheel patterns with one or two omni wheels failed. Furthermore, this paper provides a comprehensive analysis of the feasible desired movements for the packages on the conveyor under the constrained wheel speed. This provides insights and guidance on trajectory planning and design of the conveyor.

Keywords: omnidirectional conveyor; kinematic model; trajectory generation; performance atlases

check for updates

Citation: Zhang, Z.; Sun, T.; Wang, Z.; Zhang, X. Trajectory Planning and Performance Atlases of a New Omnidirectional Conveyor. *Actuators* **2024**, *13*, 441. https://doi.org/10.3390/act13110441

Academic Editor: Guanghong Yang

Received: 25 September 2024
Revised: 29 October 2024
Accepted: 2 November 2024
Published: 4 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

In recent decades, an omnidirectional conveyor system has been introduced as an automated means of transportation of packages, goods, parts, and materials in the manufacturing and logistics industries [1,2]. The omnidirectional conveyor has the ability to move packages concurrently and independently in rotation and translation in the conveyor plane thanks to an array of omni wheels [3,4]. The omni wheels consist of a base wheel with a number of rollers evenly attached around the circumference of the wheel hub, as shown in Figure 1b. The rollers are oriented with their axes perpendicular to the axis of the base wheel in order to create additional mobility. With various wheel patterns, the omni wheels are commonly aligned to generate in-plane motion with three degrees of freedom.

The advantages of the omnidirectional conveyor over conventional conveyor transfer solutions are that they are compact and flexible in transportation and deployment [5,6]. Conventional conveyor technology is characterized by rollers and powered belts, which can only fulfill simple transferring tasks, namely moving objects in a straight line [7,8]. Thus, conventional conveyor systems are not suited to develop more flexible and adaptable production lines or logistics systems in the context of smart factories. To address this issue, many types of conveyors have been developed, such as flexible conveyors [9], cognitive conveyors [10], grid-sorters [11], etc. However, these designs have limited mobility and allow path planning to be conducted in a simple manner. As a result, more advanced operations, such as sorting, palletizing, and depalletizing, etc., cannot be realized.

The omnidirectional wheeled conveyors come as a whole new solution for addressing the above limitations of a conveyor. Various conveyor types with different wheel patterns

have been developed [12–15]. Among the various patterns, the most popular one is the omni cellular conveyor [16–20], which consists of an array of regular hexagon modules with three identical omni wheels, as shown in Figure 1a. This conveyor shares kinematic similarities with the three-omni-wheeled mobile robots [21–23]. Extensive research efforts have been devoted to trajectory tracking [24], path planning [25], dynamics [26], and control [27] of the cellular conveyor. However, the kinematics of the wheel patterns that are in between the adjacent regular hexagon modules have not been utilized to move the packages. In this work, the other two triangular wheel patterns between the hexagon modules are identified, and one has proved feasible. The feasible patterns between the modules are oriented rather randomly compared to the triangular pattern in the hexagon module, making controlling and analyses of the conveyor rather difficult, if not impossible. As a result, path planning usually is conducted in a module-to-module manner [28,29]. And a greater number of wheels are thus needed. In addition, the feasible minimum size of the package and density of the wheel distributions have not been investigated. The operation of the conveyor relies on experience provided the package can always cover at least one module. Further, while the cellular conveyor has fault tolerance capabilities when a wheel fails, the entire module has to be deactivated, causing packages to be bypassed. In reality, some areas within an affected module may remain functional, making the complete deactivation of the module inefficient. This approach leads to wasted potential and reduces the feasible areas for trajectory planning.

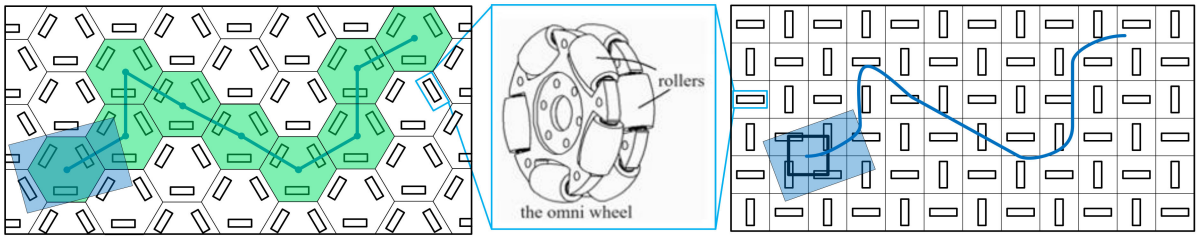


Figure 1. Diagram of omnidirectional conveyors (The blue rectangles represent the parcels): (**a**) the three-omni-wheeled cellular conveyor; (**b**) the omni wheel; (**c**) our proposed conveyor.

To address these problems, this paper proposes a novel omnidirectional conveyor. The proposed conveyor consists of a mixed array of horizontal and vertical omni wheels with a symmetric structure, as shown in Figure 1c. Similar to the cellular conveyor, although a four-wheeled pattern is identified to contribute to the basic kinematics of the conveyor, as highlighted in Figure 1c, the package actually moves with a number of different wheel patterns depending on the location and size of the packages. This means that everywhere the package moves, there is a corresponding pattern. Unlike the wheel patterns, in between the modules for the cellular conveyor, the wheels that occur everywhere on the conveyor are either horizontal or vertical, making kinematics and control of the proposed conveyor with great ease and consistency. The trajectory planning scheme can thus be conducted in a continuous manner rather than a module-to-module manner. Therefore, in this design, the basic module is the individual wheels. If one of the wheels fails, only the failed wheel is deactivated. In this work, a comprehensive study is conducted to depict the feasible workspace for the conveyor with different package sizes and failed wheels. The minimum number of wheels ensuring the whole area of the conveyor is feasible is then identified. Further, based on the minimum wheel number layout, comprehensive analyses were conducted to determine the feasible desired velocities of the package under constrained wheel velocities. This study reveals that some theoretically feasible areas may become infeasible if the desired package velocities are unachievable due to the wheel velocity limits.

The remainder of this paper is organized as follows: the kinematic model of our proposed conveyor is established in Section 2. Section 3 develops the mathematics of the trajectory planning problem. The trajectory planning is then verified using CoppeliaSim

software in Section 4. Section 5 illustrates the workspace of the conveyor based on different wheel patterns. Furthermore, workspace and the corresponding wheel patterns for the conveyor with failure of one or two omni wheels are discussed in Section 6. Last but not least, under constrained wheel speed, the feasible motions at the velocity level are discussed, and related feasible areas are depicted in Section 7. Finally, conclusion remarks and future research plan are given in Section 8.

2. Kinematic Modelling

The structure of the proposed omnidirectional conveyor is depicted as shown in Figure 1, where the omni wheels are arranged in an array evenly distributed on the platform with the horizontal and vertical wheels adjacent to each other. It shares similarities with the three-omni-wheel conveyor as a repeated core pattern of four omni wheels. However, the supporting wheels are not necessarily to be the four-wheel pattern. The covered wheels can actually form various patterns depending on the size and location of the package on it, which exert an influence on the workspace and performances of the conveyor.

In this section, a kinematic model is established to reveal the relationship between wheel rotational velocity and the package velocity. The package, goods, or other materials on the conveyor are referred to as "box" hereinafter for simplicity. As shown in Figure 2, the position of the box center O' is described by (x, y) relative to the world frame {W} attached to the conveyor. The base frame $\{b\}$ is attached to the center of the box. The angle between the axis x of the world frame and the axis x of the base frame, i.e., θ, represents the box rotational angle. Apparently, $\dot{x}$, $\dot{y}$ and $\dot{\theta}$ denote the box translational velocities in axes x and y directions and the box rotational velocity with respect to the world frame.

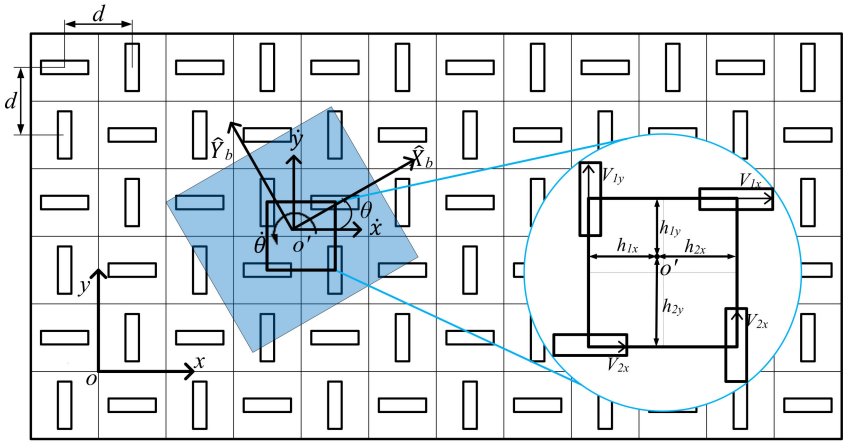


Figure 2. Kinematic representation of the omnidirectional conveyor.

Based on rigid body kinematics, the rotational velocity of each wheel can be derived as in (1).

$$\begin{pmatrix} u_{1h}^n \\ u_{2h}^n \\ u_{1v}^n \\ u_{2v}^n \end{pmatrix} = \frac{1}{r}\begin{bmatrix} 1 & 0 & -(n-1)d - h_{1y} \\ 1 & 0 & (n-1)d + h_{2y} \\ 0 & 1 & -(n-1)d - h_{1x} \\ 0 & 1 & (n-1)d + h_{2x} \end{bmatrix}\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \frac{1}{r}\begin{bmatrix} 1 & 0 & L_{1y} \\ 1 & 0 & L_{2y} \\ 0 & 1 & L_{1x} \\ 0 & 1 & L_{2x} \end{bmatrix}\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \quad (1)$$

where n refers to the serial number of rows of horizontal wheels or columns of vertical wheels counting from the center of the box. For example, $n = 1$ represents the nearest wheels to the box center. u_{ih}^n and u_{iv}^n $(i = 1, 2)$ are rotational velocities of the horizontal and vertical wheels, respectively. d is the wheel distance in both horizontal and vertical directions. h_{ix} and h_{iy} $(i = 1, 2)$ denote the horizontal and vertical positions of the vertical and the horizontal wheels with respect to the box center when n equals zero, respectively. L_{ix} and L_{iy} $(i = 1, 2)$ calculate the horizontal and vertical positions of the vertical and the horizontal wheels with respect to the box center, r is the radius of the omni wheel.

Equation (1) can be rewritten as $u = T\dot{q}$, where u and $\dot{q}$ refer to the wheel and box velocity vectors, respectively. q is the box position and orientation vector, written as $q = (x,\ y,\ \theta)$. T represents the transformation matrix. Clearly, the transformation matrix is redundant, and the rank is three. The redundant wheels provide stable support for the box. In contrast, the existing cellular conveyor has a basic module with three wheels, and the kinematics is not redundant.

In order to compare our proposed conveyor to the cellular conveyor, the kinematics for the cellular conveyor are also discussed here. According to the literature, only the triangular wheel pattern within each module is utilized, as shown in Figure 3a. However, two triangular wheel patterns exist between adjacent modules, as shown in Figure 3b,c. In order to simplify the discussion, the equilateral triangle for the three-wheel patterns is assumed.

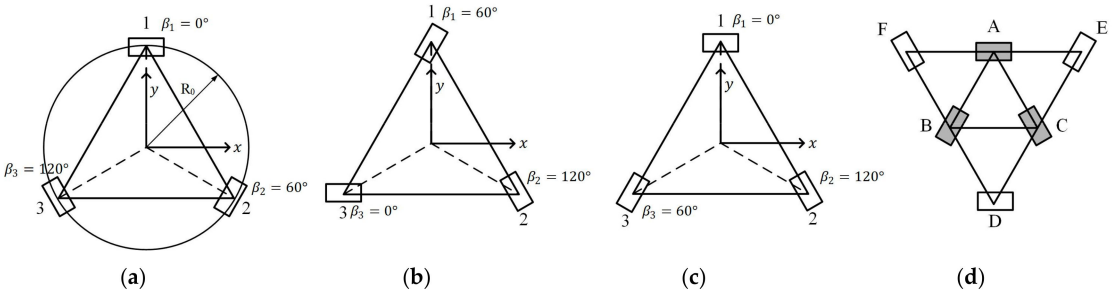


Figure 3. Three triangular wheel patterns in the cellular conveyor: (**a**) feasible wheel pattern in the module; (**b**) feasible wheel pattern in-between the modules; (**c**) infeasible wheel pattern in-between the modules; (**d**) the infeasible wheel pattern surrounded by feasible wheel patterns.

The kinematic models for the three triangular wheel patterns are given in (2)–(4), respectively. Equation (2) is derived based on the published work [20,30].

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{1}{r}\begin{bmatrix} -R_0 & 1 & 0 \\ R_0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -R_0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}\begin{pmatrix} \dot{\theta} \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{2}$$

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{1}{r}\begin{bmatrix} -\frac{1}{2}R_0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2}R_0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2}R_0 & 1 & 0 \end{bmatrix}\begin{pmatrix} \dot{\theta} \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{1}{r}\begin{bmatrix} -R_0 & 1 & 0 \\ \frac{1}{2}R_0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2}R_0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}\begin{pmatrix} \dot{\theta} \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{4}$$

where R_0 refers to the circumscribed circle of the triangle. u_1, u_2, and u_3 are the angular speeds of the wheels. It is seen that the transformation matrix of (4) is singular, and the rank is equal to two rather than three, making the third type of wheel pattern infeasible for all translational movements and rotations. However, according to the cellular conveyor layout, adding one wheel to the infeasible pattern to form a diamond wheel pattern with four wheels results in a feasible pattern. This is because any adjacent triangular pattern to the infeasible pattern in Figure 3c is one of the feasible patterns in Figure 3a,b. As shown in Figure 3d, though the triangular pattern ABC is infeasible, with one extra wheel, the adjacent wheel patterns ABF, ACE, and BCD are feasible. As a result, all of the diamond wheel patterns with four wheels are feasible. However, because the patterns in Figure 3b,c are oriented rather randomly compared to the pattern in the hexagon module as in Figure 3a, the actual control and analysis of the conveyor based on the two patterns are rather difficult, if not impossible. As a result, according to the literature [1,5,20], the path planning was conducted only in a module-to-module manner.

3. Trajectory Planning

In this section, we consider methods of computing a trajectory that describes the desired motion of the box on the conveyor. Usually, a trajectory is generated with given initial and final positions and orientations by humans and then further specified by a couple of choices of functions that define these trajectories. The choice of functions could be linear, polynomial, circular, sinusoidal, etc., as long as they satisfy the boundary conditions. Sometimes, we wish to allow a path to be specified that includes intermediate via points, and then additional continuous conditions at the via points are required to be satisfied. In general, it is desirable for the motion of the box to be smooth. For this purpose, the trajectory function should be continuous and has a continuous first derivative.

In the following, we consider four types of trajectories that are typical for actual applications: a prescribed trajectory [31], trajectories defined by intermediate via points [32], trajectories defined by Bezier curves [32], and trajectories with combined straight-line and pure rotations [33]. These trajectory planning problems are formulated in this section and verified through simulation in the next section.

3.1. Case 1: Trajectories Specified by Particular Functions

The first type of trajectory generation problem concerns the box following a prescribed trajectory defined by functions. Here, the parabolic trajectory is taken as an example. As shown in Figure 4a, the box travels from the origin of the world frame to location $(10d, 3.5d)$ in 7 s, following a trajectory function as in (5), rigorously, where x and y are horizontal and vertical positions relative to the world frame.

$$y(t) = \frac{1}{2}x(t)^2 \tag{5}$$

We consider a simple linear form of $x(t)$, as in (6). To rigorously follow the trajectory, $y(t)$ is calculated by substituting (6) into (5), as in (7).

$$x(t) = at + b \tag{6}$$

$$y(t) = \frac{1}{2}a^2t^2 + abt + \frac{1}{2}b^2 \tag{7}$$

where a and b are unknown coefficients. With (6) and (7), the velocity along this path is clearly $\dot{x}(t) = a; \dot{y}(t) = a^2t + ab$. The orientation of the box is assumed to be tangent to the trajectory and directed forward. Of course, it can be defined otherwise. Thus, we have

$$tan\theta = y' = x \tag{8}$$

Obviously, the orientation angle $\theta(t)$ is not independent; rather, it is related to the translational position of the vehicle. From (5), we can derive the orientation angle $\theta(t)$ and its derivative, i.e., the rotational velocity of the vehicle $\dot{\theta}(t)$, as in (9) and (10).

$$\theta(t) = \arctan\,(at + b) \tag{9}$$

$$\dot{\theta}(t) = \frac{a}{1 + (at + b)^2} \tag{10}$$

Given the initial and goal locations, i.e., $(0, 0)$ and $(10d, 3.5d)$ for initial and goal locations, respectively, the unknown parameters a and b can be computed. The calculated box velocities $\dot{x}(t)$, $\dot{y}(t)$, and $\dot{\theta}(t)$, are then converted into the angular velocity of each wheel using (1). Note that in simulations, the distance between the box center and the farthest actuated wheels is set to be within $2d$, i.e., two times the horizontal or vertical wheel distances, to ensure the covered wheels are being actuated (same for the next three simulations). The values of n depends on the locations of the actuated wheels to the box center. In fact, horizontal wheels in the same line or vertical wheels in the same column

share the same velocities according to the kinematics. In actual simulations, only the wheels within the distance $2d$ to the box center are actuated to increase simulation speed. It should be noted that the initial velocity for the box is not zero in this example; to generate a smoother trajectory, the initial velocity may be zero. For this purpose, a higher degree of functions may be used rather than a linear one, as in (6). In this case, more constraints need to be used to specify a particular curve.

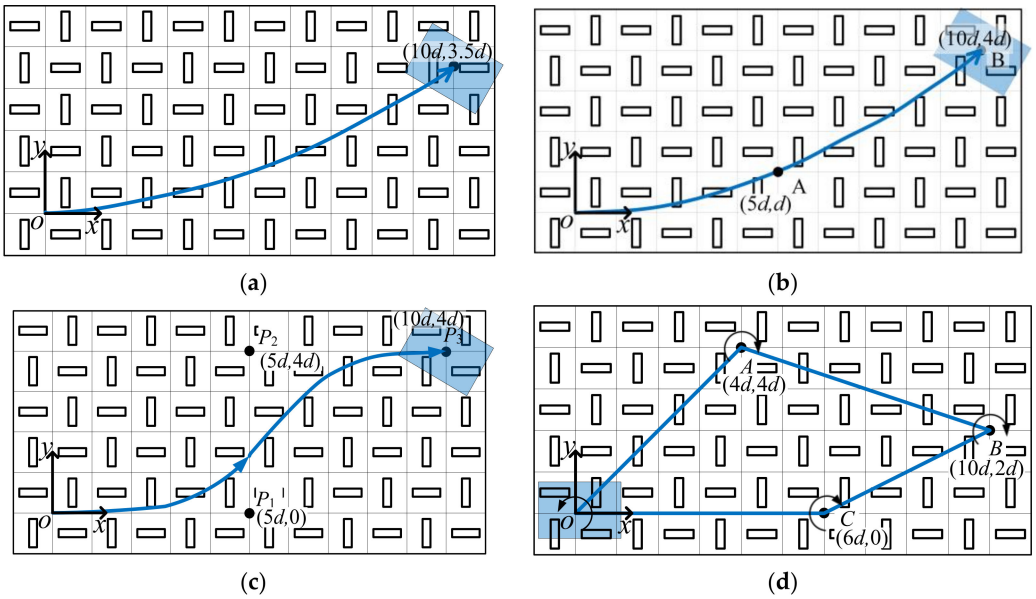


Figure 4. The box following four trajectories: (**a**) a parabolic trajectory; (**b**) two segments of cubic polynomials through a via point; (**c**) a cubic Bezier curve; (**d**) trajectories of straight lines and pure rotations.

3.2. Case 2: Trajectories Include via Points

The second type of trajectory considered in this section concerns the box passing through a set of via points. More via points included in the trajectory result in more constraints being specified to derive the unknowns of the trajectory functions. A simple case with only one via point is examined in this section. As shown in Figure 4b, the initial, via, and goal points are defined as $(0, 0)$, $(5d, d)$, and $(10d, 4d)$, respectively. Unlike the first trajectory example, where the orientation of the box is tangent to the trajectory all the time, in this case, the orientation of the box is tangent to the trajectory only at the initial, via, and goal points. The rotational speed is specified to be constant. The box is to move 3 s from the initial position to the via points, and it takes another 4 s to arrive at the goal point. To make the whole trajectory smooth, the trajectory, velocity, and acceleration at the via point should be continuous. In addition, the velocity of the box at the initial and goal points is required to be zero. Here, we use polynomials as the trajectory function. And all these mentioned constraints uniquely specified two particular cubic polynomials, as in (11) and (12).

$$x_i(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + a_{i3}t^3 \tag{11}$$

$$y_i(t) = b_{i0} + b_{i1}t + b_{i2}t^2 + b_{i3}t^3 \tag{12}$$

where $i = 1$ and $i = 2$ refer to the trajectory segment OA and AB, respectively. $x_i(t)$ and $y_i(t)$ refer to the box locations in axes x and y, respectively. Combining the above-mentioned constraints yields 16 equations in 16 unknown coefficients [34]. Solving for the sixteen coefficients, it is easy to derive the box velocities in the two trajectory segments, i.e., $\dot{x}_1(t)$, $\dot{y}_1(t)$, $\dot{x}_2(t)$, and $\dot{y}_2(t)$. By application of the kinematics, the angular velocity of each wheel can be computed. Compared to the first trajectory of a simple parabolic curve, this trajectory is defined by an extra via point, and more constraints are employed to make the

trajectory smoother. Based on this example, the trajectory that includes more via points could be defined, and more constraints are obviously required to solve the unknowns.

3.3. Case 3: Bezier Curves

The third type of trajectory that is commonly used in trajectory planning is the Bezier curve. Here, the cubic Bezier curve is used. The cubic Bezier curve requires four given points, as shown in Figure 4c. The box travels from the starting point $P_0(0,0)$ to reach the end point $P_3(10d, 4d)$. The trajectory is further shaped by the other two points $P_1(5d, 0)$ and $P_2(5d, 4d)$ such that the curve is tangent to line $\overline{P_0P_1}$ and $\overline{P_2P_3}$. The detailed equation for the cubic Bezier curve is in (13).

$$B(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3 \tag{13}$$

where $t \in (0,1)$ $P_i = (x_i(t), y_i(t))^T$ $(i = 0,1,2,3)$, and $B(t) = (x(t), y(t))^T$. From (13), it is seen that four points are required to uniquely specify the cubic Bezier curve, which essentially is the polynomials. The Bezier curve is also frequently applied in Computer-Aided Design (CAD) for drawing complex smooth curves. If a trajectory is defined through human drawings combined with CAD software, the Bezier curve can be considerably useful.

3.4. Case 4: Trajectories Through Straightline Segments and Pure Rotations

The fourth type of trajectory generation problem concerns the box passing through a set of via points that are connected through straight lines. The orientation of the box remains unchanged when moving along the straight lines until it comes to rest at each via point. The orientation of the box only changes at the initial and via points to make the orientation of the box the same as the incline angle of the next straight-line segment, as shown in Figure 4d. In this case, the translational and rotational motions are separated. A constant rotational velocity of 1 rad/s at the stopping points is assumed to simplify the trajectory planning. Cubic polynomials are used for the line segments, as in (14)–(19).

$$x_i(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + a_{i3}t^3 \tag{14}$$

$$y_1(t) = x_1(t) \tag{15}$$

$$y_2(t) = -\frac{1}{3}x_2(t) \tag{16}$$

$$y_3(t) = \frac{1}{2}x_3(t) \tag{17}$$

$$y_4(t) = 0 \tag{18}$$

$$y_5(t) = 0 \tag{19}$$

where $i = 1$ refers to segment OA, $i = 2$ refers to the next line segment AB, and so on. For each line segment, it is assumed that the box velocity at the start and end locations is zero. The time spent at the line segments OA, AB, BC, and CO is set to be 4, 6, 4, and 6 s, respectively. Solving for the sixteen unknowns, the trajectories of each line segment are determined. Then, using (1), the wheel rotational velocities can be calculated.

4. Simulation Analyses

4.1. Creating a 3D Model in CoppeliaSim

In this section, the 3D model of the omnidirectional conveyor is created in CoppeliaSim software (version 4.5.1). Since the omni wheel has complex structures, it is built in professional CAD software (Solidworks, version 2022) and then imported to the CoppeliaSim. Firstly, the CAD model of an omni wheel is downloaded from the GrabCAD website. Secondly, the base placer, shaft connector, and wheel base of the omni wheel are merged together to make a single stationary base part. Then, the CAD model of the omni wheel is converted to URDF (Universal Robot Description Format) format, with a clearly defined

hierarchy of the wheel rollers and the single stationary base and coordinate systems. The URDF format of the omni wheel is finally imported to CoppeliaSim.

The omni wheels and the platform are set as non-respondable to each other, thus simplifying the cuboid body structure design and decreasing computational cost. The schematic top view of the conveyor is presented in Figure 5, where the wheel's horizontal and vertical distances are all equal to $d = 0.07$ m. The box is modeled as a cuboid directly with an edge size equal to 0.21 m, and its radius of inscribed circle is 0.105 m, slightly larger than $\sqrt{2}d$ in order to ensure that everywhere on the conveyor is feasible for trajectory generation. $\sqrt{2}d$ is the minimum size of the box to ensure that everywhere on the conveyor is feasible to move and rotate. Detailed feasible areas are depicted in Sections 5 and 6.

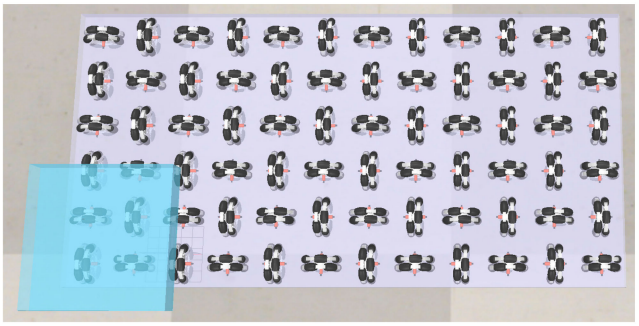


Figure 5. A top-view screenshot of an omnidirectional conveyor model in CoppeliaSim (the box is modeled in blue).

The wheel joints, as indicated in red color, are set to dynamic mode. The joints of the rollers are set in free control mode such that the rollers rotate passively. The conveyor is interfaced with Python(Version 3.12) child script using the regular API provided by CoppeliaSim.

4.2. Trajectory Generation with PID Control

To reduce the tracking errors, the PID control strategy is applied. As shown in Figure 6, the tracking error of the box positions and orientations in world frame $e(t)$ can be calculated readily (i.e., $e(t) = q_{ref} - q$, where q_{ref} and q are desired and actual position and orientation vector of the box, respectively) and fed to the PID controller to generate the control signal, i.e., the box velocity with respect to the global frame $(\dot{\theta}, \dot{x}, \dot{y})$. Then, the box velocity is used to obtain the wheel rotational velocities using (1).

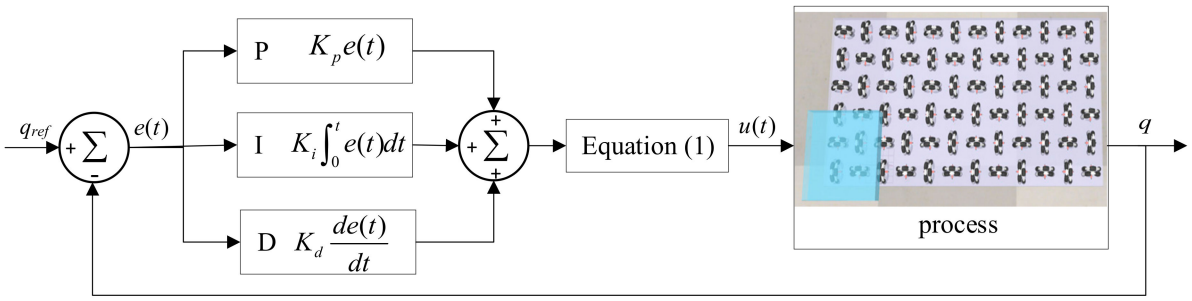


Figure 6. Block diagram of the omnidirectional conveyor under PID control.

4.3. Simulation Results

The snapshot of the final trajectory planning result is depicted in Figure 7. The actual trajectories are depicted in red color. During the simulations, only the wheels covered by the box rotate while other wheels remain stationary to reduce the computational and simulation time. The simulation time interval is set to 0.05 s for one cycle of computations. The PID gains $(K_p,\ K_i, K_d)$ for directions x, y and rotation are (1, 1, 1.5), (2, 1.5, 1.5), and (2, 1, 1.8), respectively. The same PID gains are used for other trajectories. Figure 8 illustrates

the trajectory following results for the four trajectory cases, in which it shows that the actual and desired trajectories and orientations agree well with each other.

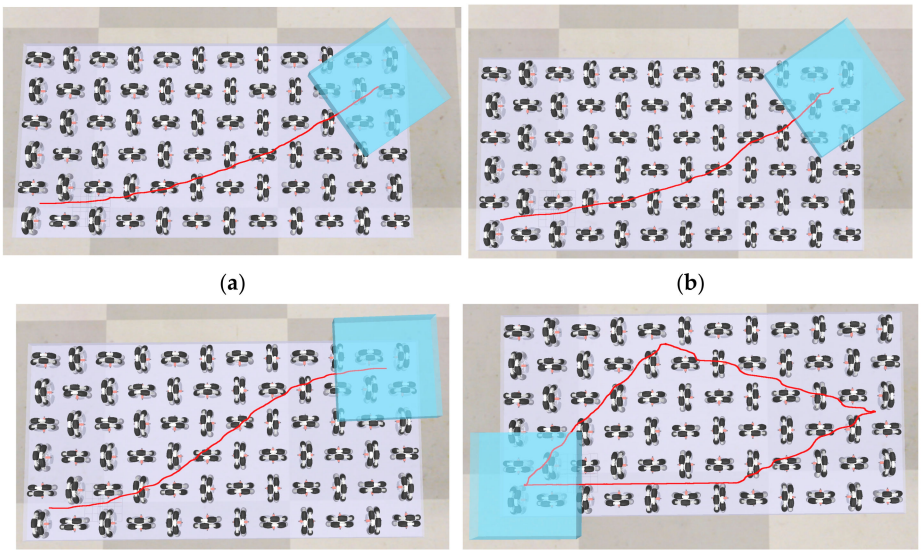


Figure 7. A top-view screenshot of actual trajectories in CoppeliaSim (The blue rectangles represent the box and the red lines refer to the trajectories.): (**a**) case 1; (**b**) case 2; (**c**) case 3; (**d**) case 4.

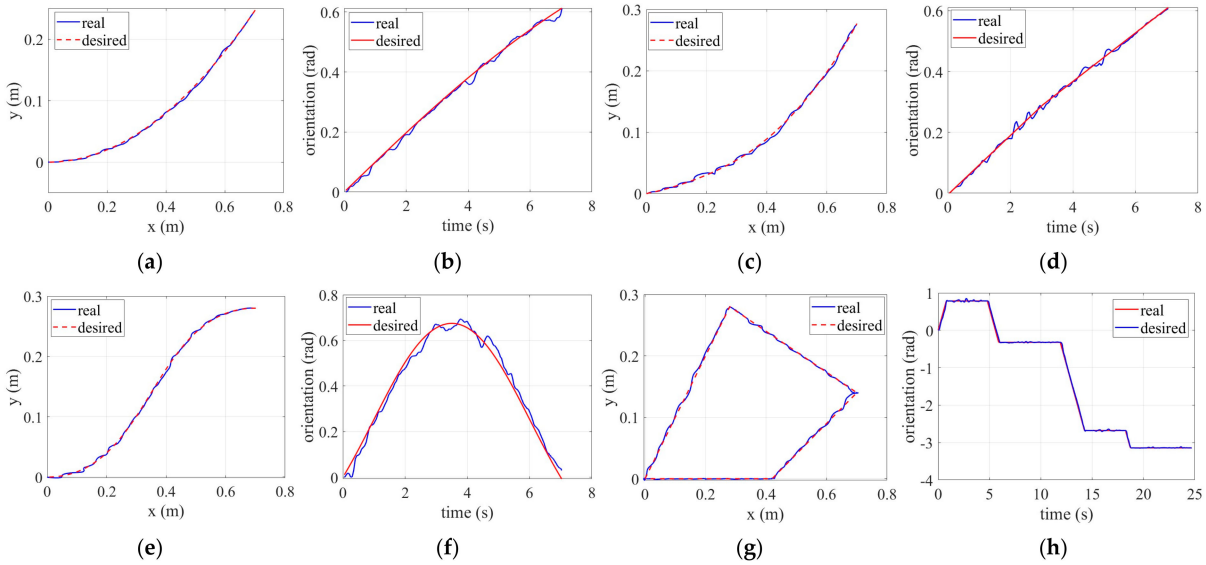


Figure 8. Simulation results: (**a**) trajectory of case 1; (**b**) orientation of case 1; (**c**) trajectory of case 2; (**d**) orientation of case 2; (**e**) trajectory of case 3; (**f**) orientation of case 3; (**g**) trajectory of case 4; (**h**) orientation of case 4.

5. Workspace and Wheel Patterns

We have seen in previous sections that the proposed omnidirectional conveyor is feasible for the transfer and rotation of the box. Clearly, when the box is moving, the number and patterns of covered wheels may change. The number and patterns of the covered wheels may also be different for different sizes of the boxes, as shown in Figure 9. In Figure 9 and hereinafter, we used a small square and a circle to replace the omni wheels for simplicity, such that the squares and circles represent horizontal and vertical wheels, respectively, or vice versa. In this work, an inscribed circle located at the center of a box is used to represent the shape and size of a box, as shown in Figure 9. The inscribed circle is used as the representation circle because if the inscribed circle can cover a particular wheel

pattern that is feasible for moving and rotating the circle, then the actual wheel pattern that the original box covers is apparently feasible. The inscribed circle ensures that the wheel pattern remains unchanged with the change in box orientation.

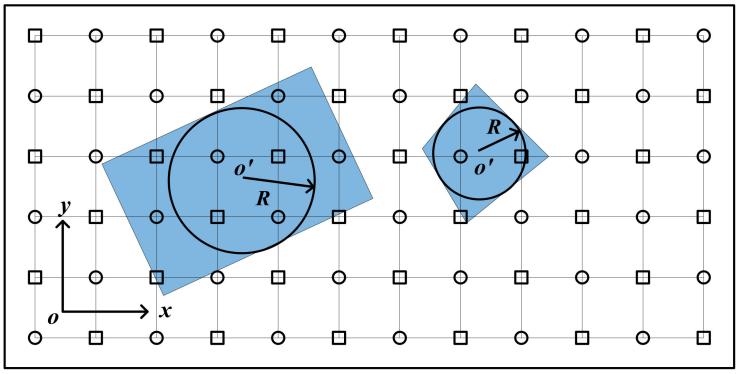


Figure 9. An inscribed circle for representation of the box (The circle and the square refer to the horizontal and vertical wheels, respectively, or vice versa.).

As shown in Figure 9, for the box on the left, the inscribed circle covers a particular four-wheeled pattern, while on the right, the inscribed circle covers only a particular two-wheeled pattern. For the left box, the four-wheeled pattern is feasible based on (1), while for the right box, the two-wheeled pattern is infeasible. The corresponding areas (where the center of the inscribed circle is located) for the infeasible wheel patterns are referred to as the infeasible areas. Apparently, for large enough box sizes (represented by the radius of the inscribed circle), the number of covered wheels is large, and all the areas on the conveyor can be feasible. However, for smaller box sizes, some infeasible areas exist.

What is the smallest box size to ensure all areas are feasible? In other words, given the smallest box size, what is the largest horizontal or vertical distance between two adjacent wheels to ensure all the areas are feasible? In this case, the smallest number of wheels is used, thus reducing costs. In this section, a comprehensive study is conducted to find the feasible areas for different box sizes.

If we draw a boundary circle on the conveyor with the circle center located at one of the wheel centers and its radius being the same as the radius of the inscribed circle of the box, then if the box (represented by the inscribed circle) is located within the boundary circle, the wheel where the boundary circle is located will be covered by the box. The wheel will not be covered by the box if the box moves outside the boundary circle. If two boundary circles corresponding to two wheels have an intersection, the related two wheels will both be covered by the inscribed circle of a box if the box is located within the intersection area. With these, we can draw all the boundary circles for every wheel, as shown in Figures 10–18. These boundary circles divide the conveyor into many small areas. Within different small areas, the number of covered wheels and their pattern may vary.

It is clear that a four-wheeled pattern makes a square that is repeated over the conveyor. And the small intersection areas for each four-wheel square are the same. As a result, we only have to investigate the feasible areas within a four-wheel square. For a small radius of the inscribed circle, the number of covered wheels is generally less than the one with a larger radius. We drew all the possible wheel patterns with the inscribed circle radius ranging from smaller than $d/2$ to greater than $\left(\sqrt{10}d\right)/2$, as shown in Figures 10–12. The shaded intersection areas refer to the location of the center of the inscribed circle, and the highlighted wheels are corresponding covered wheels. Due to symmetry, only one of the identical areas is shaded for simplicity, and one can deduce the other corresponding symmetric areas that share identical wheel patterns.

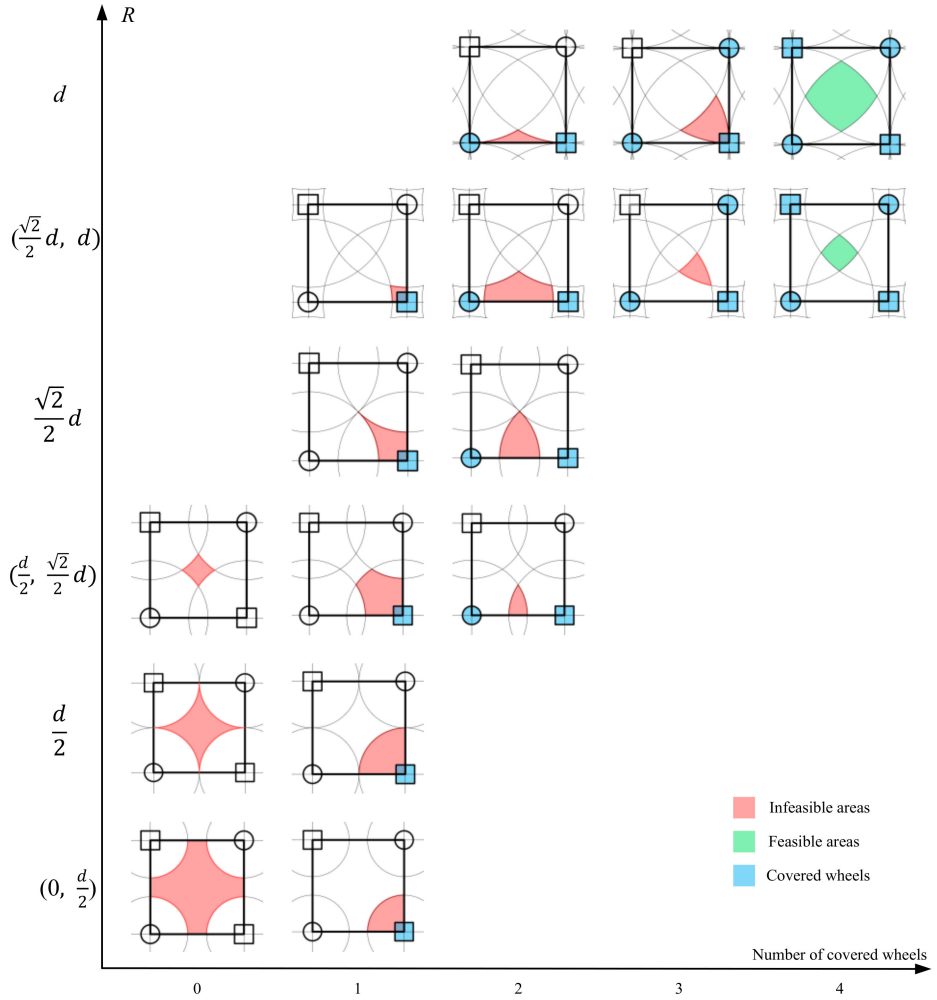


Figure 10. Workspace and wheel patterns: part 1.

It is seen from Figure 10 that when the radius of the inscribed circle is less or equal to the wheel distance d, the number of covered wheels is never above four. Obviously, wheel patterns with 0, 1, 2, and 3 wheels are incapable of driving the box to any translational and rotational velocities and hence infeasible. When the radius of the inscribed circle of the box is equal to d, only the central area is feasible as it has four wheels. However, since its feasible area is surrounded by infeasible areas, the overall conveyor is infeasible for generating an actual trajectory. As a result, when the radius of the maximum inscribed circle is no larger than d, the conveyor cannot be used. In other words, if the horizontal or vertical wheel distance d exceeds the radius of the inscribed circle R, then the conveyor is unusable. The red and green colors refer to the infeasible and feasible areas, respectively.

Similarly, as shown in Figure 11, when the radius of the inscribed circle is smaller than $\sqrt{5}d/2$, new wheel patterns appear with four and five wheels. However, the newly emerged pattern is also infeasible, for there is only one horizontal wheel with three or four vertical wheels (or only one vertical wheel with three or four horizontal wheels). Only feasible areas occur around the center of the four-wheel square, but these feasible areas are isolated by infeasible areas, which is the same as in Figure 10, making the whole conveyor infeasible. When the radius of the inscribed circle is equal to $\sqrt{5}d/2$, the central feasible areas are connected to each other via a point. It is critically feasible to use the conveyor. When the radius of the inscribed circle increased to be in the range of $(\sqrt{5}d/2, \sqrt{2}d)$, new wheel patterns of five and six wheels emerge. The newly emerged patterns turn out to be feasible and adjacent to the feasible area of the four-wheel pattern. In this case, the conveyor has a whole connected feasible area, and the box can be transferred to this

combined area. However, the connecting areas are narrow compared to the wide feasible area of the four-wheel pattern. The box has to move precisely in those narrow areas, and a possible deviation can cause it to move to the infeasible areas.

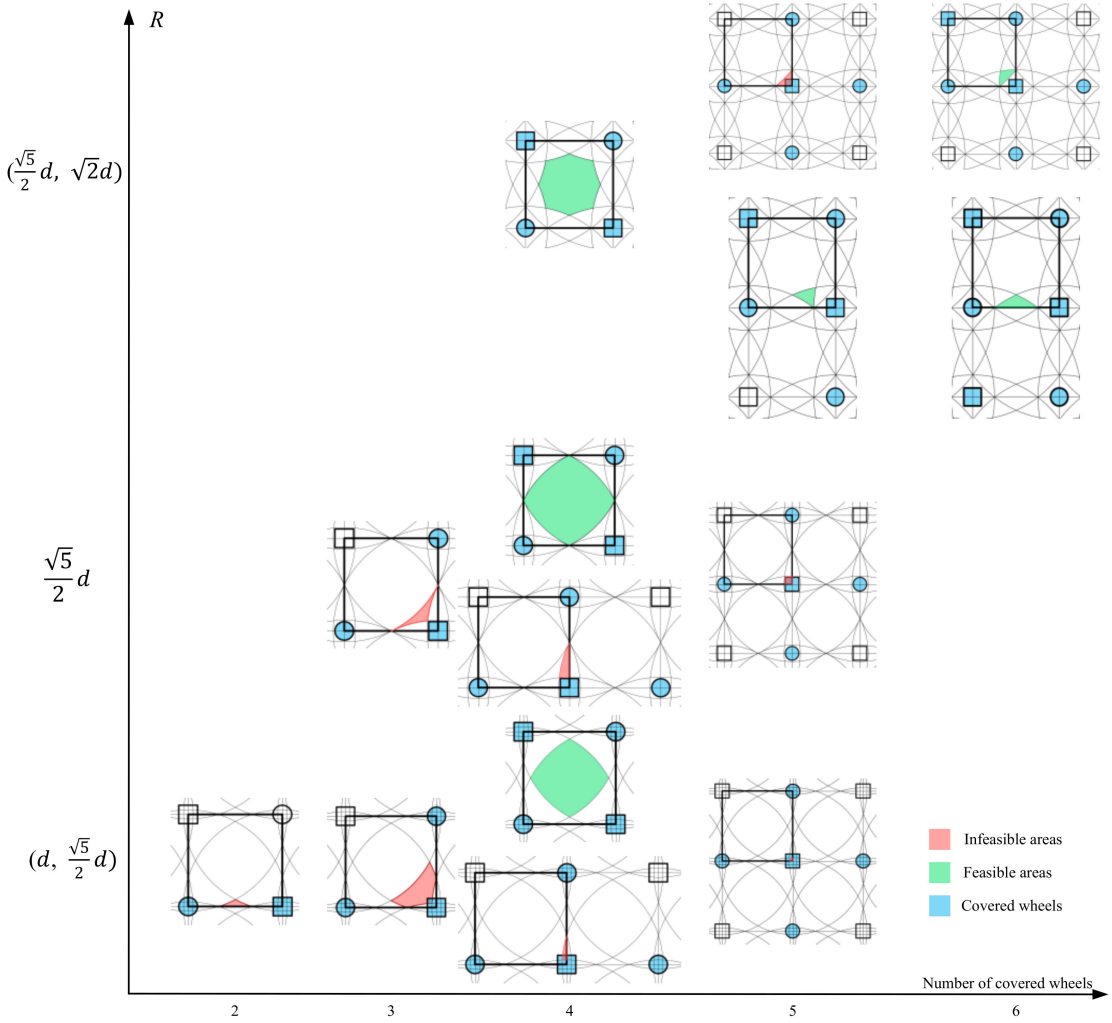


Figure 11. Workspace and wheel patterns: part 2.

When the radius of the maximum inscribed circle is equal to or greater than $\sqrt{2}d$, there are no infeasible areas, and the whole conveyor is everywhere feasible, as illustrated in Figure 12. As the radius of the representation circle increases, new wheel patterns with more wheels emerge. When the radius increased to be equal to $\sqrt{10}d/2$, at least seven wheels are covered by the box. Clearly, more covered wheels make the transfer and rotation more stable and can bear heavier boxes with greater wheel fault tolerance capability. In contrast, a smaller number of wheels is less cost-effective, and the system is simpler. At this point, the radius of $\sqrt{2}d$ is optimal in minimizing the number of wheels and cost of the conveyor. In other words, the maximum horizontal and vertical distance between two adjacent wheels is $\frac{R}{\sqrt{2}}$ to ensure feasibility everywhere on the conveyor, where R is the radius of the inscribed circle of the smallest box. In fact, the size of an actual box is greater than the inscribed circle, which means potentially more wheels may be covered, leading to the transfer of a box being more reliable.

To further compare the proposed omnidirectional conveyor with the existing cellular conveyor, the feasible areas and wheel patterns for the cellular conveyor are illustrated as shown in Figure 13. In Figure 13, the small circles refer to the wheels. It is seen that until the box size was increased to be at least equal to $2d/\sqrt{3}$, the whole areas are feasible.

Similar to the proposed conveyor, further increasing the box size will result in more complex wheel patterns with more covered wheels. The box size less than $2d/\sqrt{3}$ results in either infeasible areas with 0, 1, 2, and 3 wheels or discontinuous areas with triangle patterns. Since any four-wheeled patterns that emerge are comprised of one feasible triangle pattern, the four-wheeled pattern is feasible. In contrast to the diamond shape of the four-wheeled pattern for the cellular conveyor, the four-wheeled pattern of our proposed conveyor is square, thus allowing the isotropic supporting of the box. More critically, the diamond shape of the four-wheeled pattern is oriented rather randomly, making actual controlling difficult. As a result, the actual control of the cellular conveyor is based on the hexagon modules.

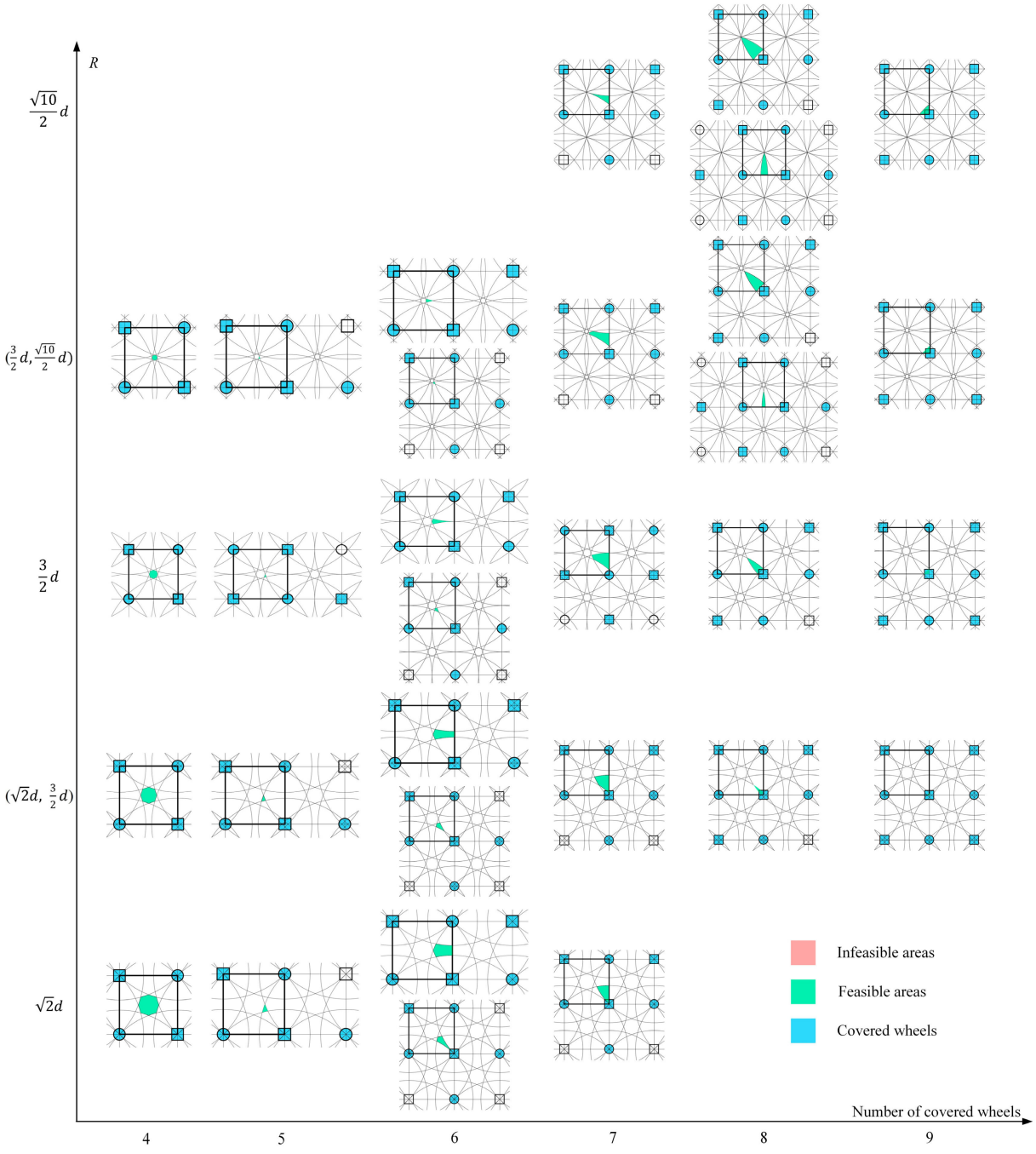


Figure 12. Workspace and wheel patterns: part 3.

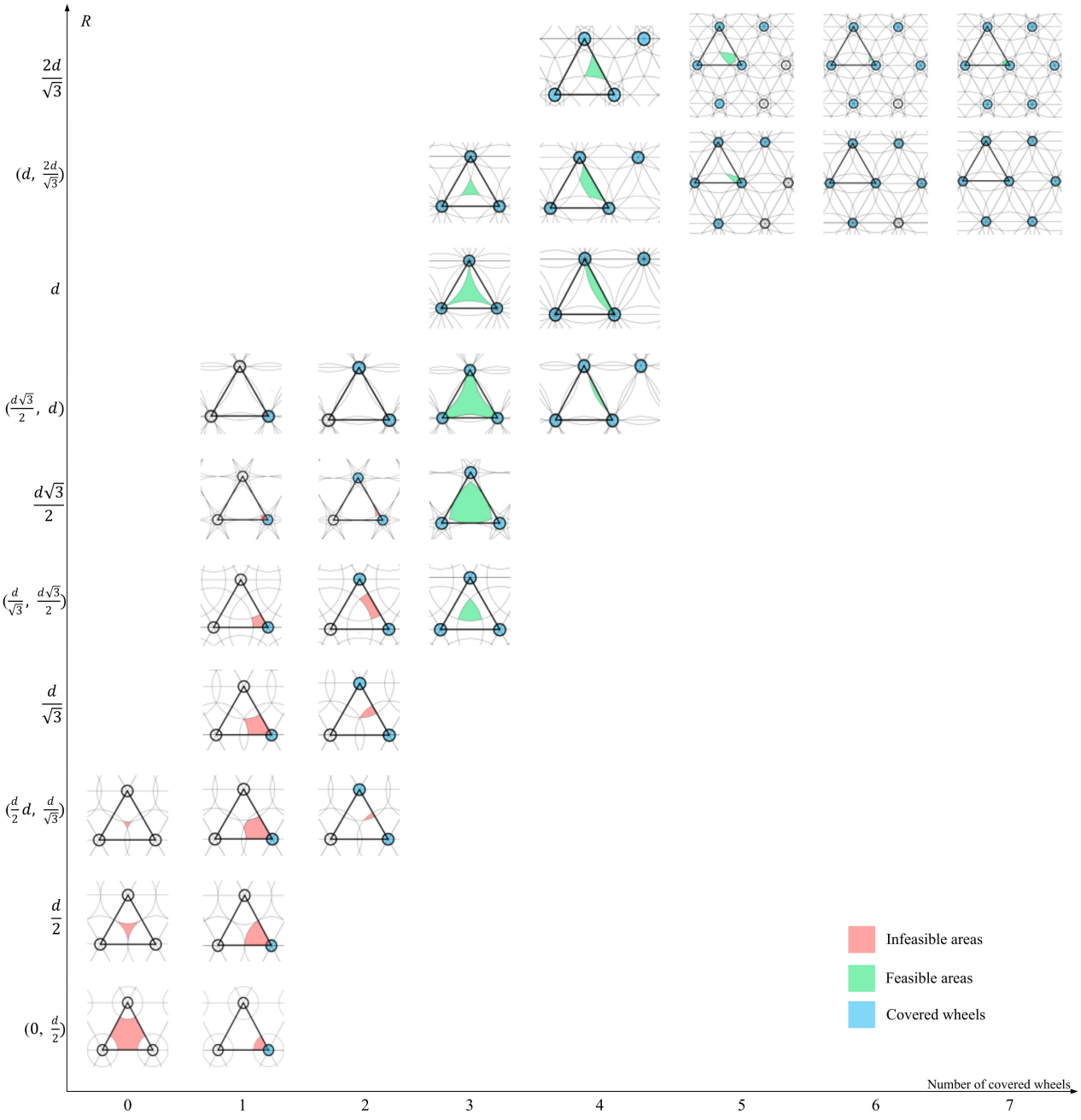


Figure 13. Workspace and wheel patterns for the cellular conveyor.

It is also interesting to find that if we draw a circumscribed circle with radius R_0 for the three-wheel pattern of the cellular conveyor and the four-wheel pattern of the proposed conveyor, the minimum radius of the inscribed circle of the box for ensuring the whole conveyor feasible are both equal to $2R_0$ (which is $\sqrt{2}d$ for the proposed conveyor and $2d/\sqrt{3}$ for the cellular conveyor). This makes the cellular conveyor better than the proposed conveyor in reducing the number of wheels. However, because of the difficulty of using the other wheel patterns, the conveyor is utilized based on the hexagon module [1,5,20]. And a greater number of wheels are thus needed compared to our proposed conveyor to ensure that at every time, there is at least one of the modules covered.

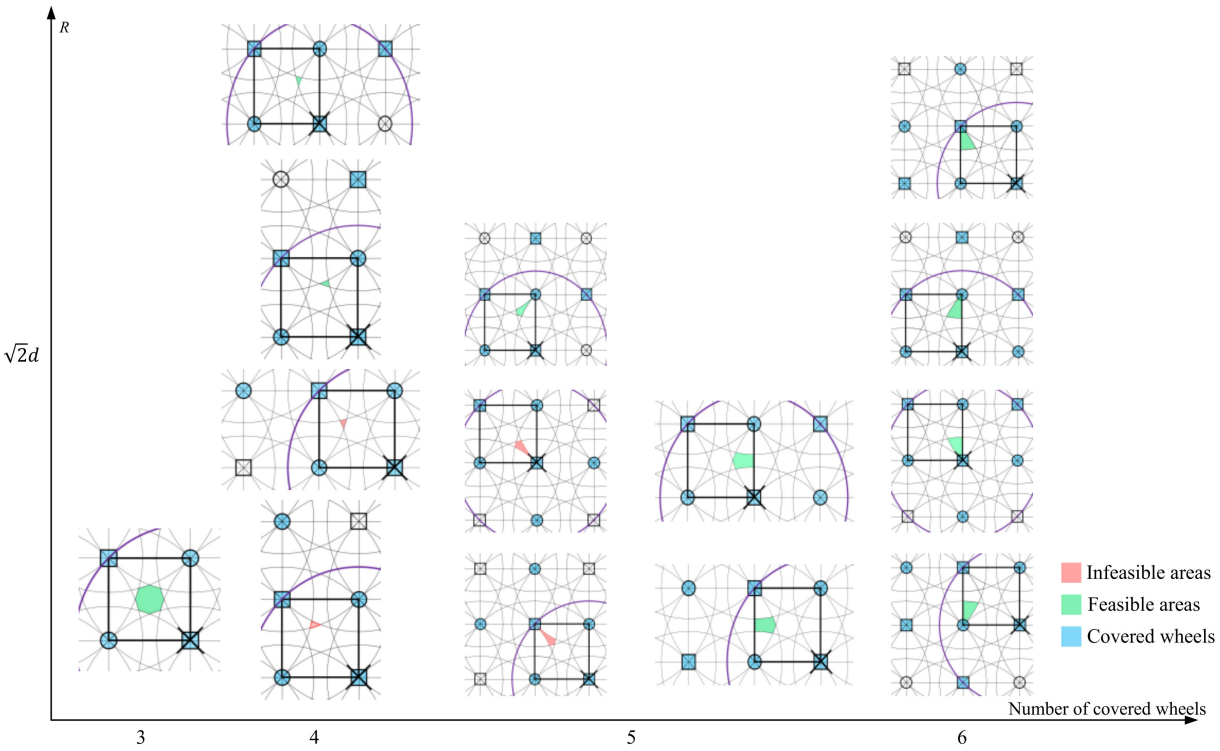


Figure 14. Workspace of the conveyor with one failed wheel.

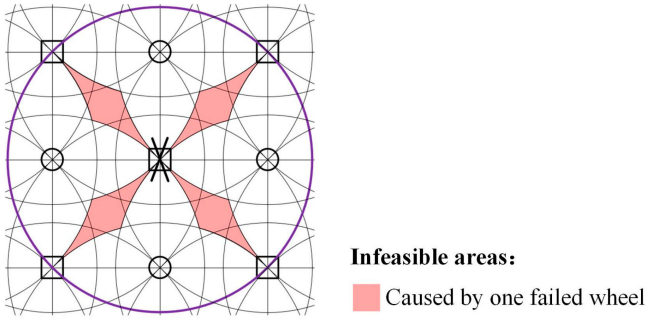


Figure 15. Infeasible areas with one wheel failed.

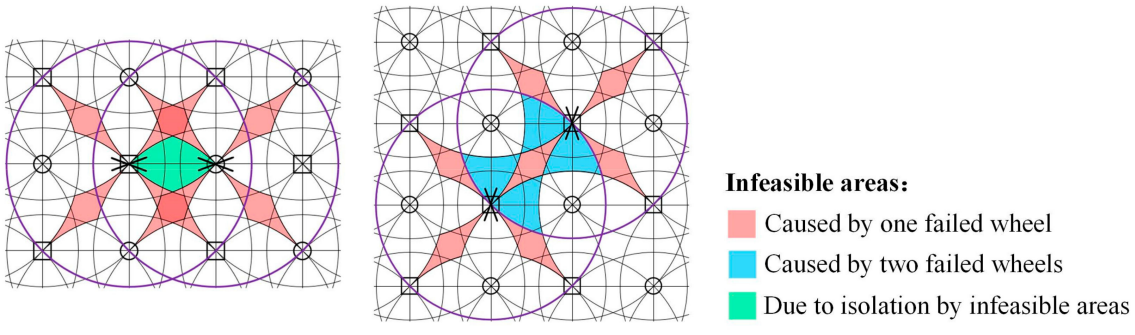


Figure 16. Infeasible areas with two adjacent wheels failed: (**a**) horizontal or vertical adjacent failed wheels; (**b**) diagonal adjacent failed wheels.

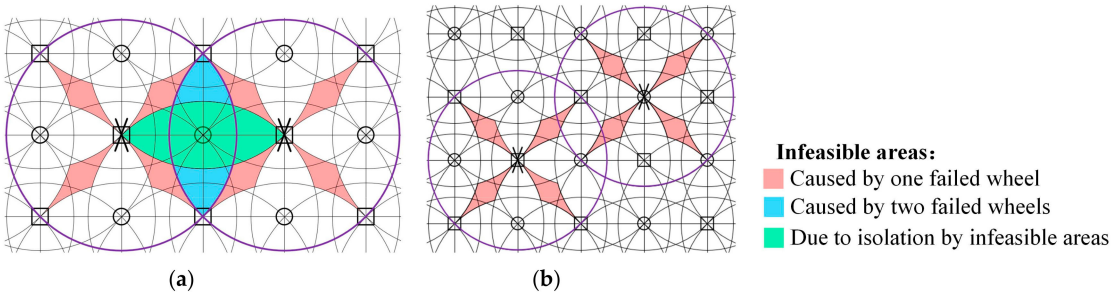


Figure 17. Infeasible areas with two wheels failed (with one normal wheel in between): (**a**) horizontal or vertical failed wheels; (**b**) diagonal failed wheels.

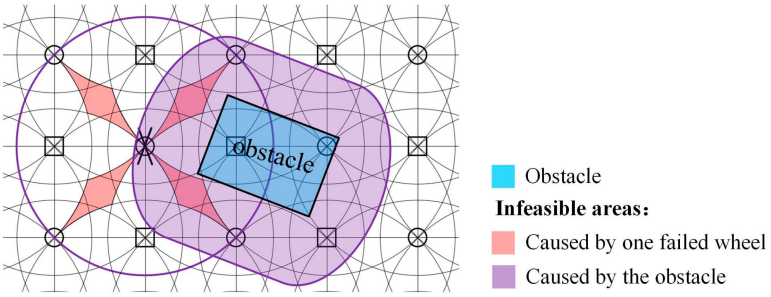


Figure 18. Infeasible areas for collision avoidance.

6. Fault Tolerance

Section 5 discusses the feasible areas of the conveyor for different box sizes under the assumption that all the omni wheels are well-conditioned. In practice, the conveyor should have the ability of fault tolerance of the wheels to increase the productivity of the conveyor. Obviously, failed wheels will change the wheel patterns and, therefore, may cause a feasible area to be infeasible. This section studies this issue and redraws the workspace to provide insights and guidance on trajectory planning.

From Section 5, it is found that $\sqrt{2}d$ is the minimum radius of the inscribed circle that is feasible everywhere on the conveyor. In other words, the maximum wheel distance d should be $R/\sqrt{2}$ to ensure that the overall conveyor is feasible for a box with the radius R of its inscribed circle. In this case, the conveyor can have a minimum number of wheels, and the whole area is feasible. So, in this section, we discuss the fault tolerance issue with the radius of the represented circle to be equal to $\sqrt{2}d$. Other sizes can be derived using the same logic.

The distinct wheel patterns for different areas are depicted only once in Figure 14. Different areas sharing the same wheel pattern can be easily deduced. As compared to Figure 12, decreasing the number of wheels leads to pattern changes. Originally, all the areas were feasible, and all the patterns were feasible patterns. Due to a loss of one of the covered wheels, the feasible pattern may become infeasible, and the corresponding area is invalid. To find the feasible area, first, a boundary circle (the purple circle in Figures 14–18) is drawn with its center located at the failed wheel, and the radius is the same as the inscribed circle of the box. Clearly, when the box center moves outside the boundary circle, the failed wheel is not covered by the box, and therefore, the failed wheel does not affect the box. The areas inside the boundary circle may be feasible or infeasible depending on the wheel patterns.

It can be seen from Figure 14 that new wheel patterns emerged due to the loss of the failed wheel. The feasible and infeasible areas are shaded with green and red, respectively. The areas with identical wheel patterns are not presented in Figure 14 to avoid complexity. Combined with the infeasible areas, the whole infeasible area in red is depicted as shown in Figure 15. Clearly, all the areas within the boundary circle are affected by the failed patterns,

but not all of them turn out to be infeasible. The infeasible areas consist of four independent parts with a symmetric layout.

If two omni wheels fail at the same time, the feasible and infeasible areas are depicted as shown in Figures 16 and 17, where the area identified by blue refers to the new infeasible areas caused by combined two failed wheels. The green area is theoretically feasible, but because it is isolated by infeasible areas, it also becomes infeasible. If two failed wheels have two wheels in between, the boundary circles do not intersect with each other, and all the infeasible areas are caused solely by the two failed wheels independently.

Based on the above discussions, trajectory planning must be conducted in the feasible areas. Commonly, the conveyor is used in situations where many boxes are presented. Trajectory planning must ensure that no collision occurs between the boxes. Figure 18 illustrates the infeasible areas for collision avoidance with one-wheel fails. The infeasible areas consist of the red areas caused by one wheel failure and the purple area inside the surrounding line (the purple lines on the right, in Figure 18) caused by one obstacle. The distance between the surrounding line and the obstacle boundary is the same as the radius of the inscribed circle of the box. When the box center is outside of the surrounding line, no collision occurs. It is seen that the infeasible areas consist of the union areas of both the infeasible area due to wheel failures and the infeasible areas generated by the obstacles.

7. Feasible Motions under Constrained Wheel Velocity

In Section 5, it is seen that the wheel patterns depend on the size and location of the box. It was found that when the radius of the inscribed circle of the box is greater than or at least equal to $\sqrt{2}d$, all the areas are feasible. However, the feasibility is based on the assumption that the wheel rotational velocity is unbounded. When the wheel rotational velocity has a limit, parts of the feasible areas may become infeasible. Here, each wheel is assumed to be identical, and the maximum rotational velocity cannot exceed a certain value, denoted as $u_{max}(> 0)$. The maximum translational wheel velocity is thus $V_{max} = u_{max}r$.

Based on (1), the translational velocity of the horizontal wheel above and below the box center can be written as in (20) and (21), respectively.

$$V_{1x} = \dot{x} - \left[(n-1)d + h_{1y}\right]\dot{\theta} \tag{20}$$

$$V_{2x} = \dot{x} + \left[(n-1)d + h_{2y}\right]\dot{\theta} \tag{21}$$

With constrained wheel velocity, it is obvious to have $|V_{1x}| \leq V_{max}$ and $|V_{2x}| \leq V_{max}$, which can be rewritten as

$$-V_{max} \leq \dot{x} - \left[(n-1)d + h_{1y}\right]\dot{\theta} \leq V_{max} \tag{22}$$

$$-V_{max} \leq \dot{x} + \left[(n-1)d + h_{2y}\right]\dot{\theta} \leq V_{max} \tag{23}$$

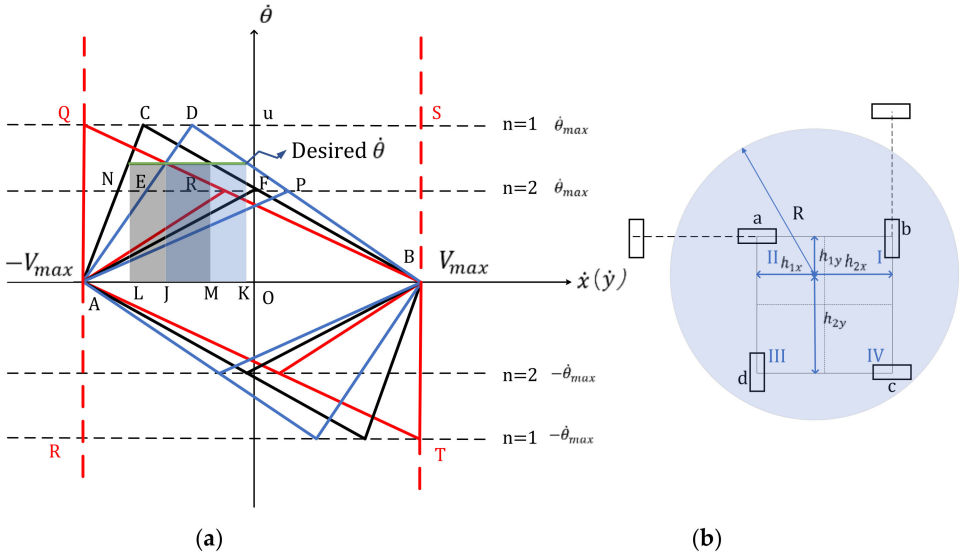
The same results can be derived for the vertical wheels, as in (24) and (25), respectively.

$$-V_{max} \leq \dot{y} - [(n-1)d + h_{1x}]\dot{\theta} \leq V_{max} \tag{24}$$

$$-V_{max} \leq \dot{y} + [(n-1)d + h_{2x}]\dot{\theta} \leq V_{max} \tag{25}$$

Equations (22)–(25) can be conveniently represented graphically, as in Figure 19. Figure 19b shows the inscribed circle locating its center at the II quadrant of the covered four-wheel square area, and Figure 19a depicts the relevant feasible areas and boundaries generated by (22)–(25). The bold, solid black line indicates the feasible area for the horizontal velocity of the box $\dot{x}$ and rotational velocity of the box $\dot{\theta}$. The bold solid blue line represents the feasible area for vertical velocity $\dot{y}$ and rotational velocity $\dot{\theta}$ of the box. Above the horizontal axis, the areas are feasible for the clockwise rotation of the box, and below

the horizontal axis, the areas relate to the counterclockwise rotation of the box. Due to symmetry, only the upper part of the horizontal axis is considered here.



(**a**) (**b**)

Figure 19. Feasible areas with constrained wheel velocity: (**a**) feasible areas for desired movements; (**b**) box location and wheel patterns.

If the inscribed circle of a box only covers the four wheels that form a square, as shown in Figure 19b, the feasible areas for the horizontal and vertical wheels form a parallelogram, respectively. The upper parts form a triangle. For example, if the desired horizontal and rotational velocities of the box are within the triangle ABC, it is achievable with the constrained wheel velocity. The triangle ABD indicates the feasible desired vertical and rotational velocities of a box. If we draw a horizontal line indicating the desired rotational velocity of a box: $\dot{\theta}$, then the corresponding feasible horizontal and vertical velocities are within the ranges $\overline{LM}$ and $\overline{JK}$, respectively.

It is easy to find that the area of the feasible triangle areas is irrelevant to the location of the box and calculated to be a constant value: $\frac{2V_{max}^2}{(n-1)d}$. Obviously, the greater the value of n, that is, the farther the wheel is covered, the smaller the feasible areas the desired trajectory can be achieved. Here, in this work, we only consider $n = 1, 2$. Because the area is constant, the height of the triangle is also constant. As a result, the maximum feasible desired rotational velocity for the box is $\dot{\theta}_{max} = \frac{2V_{max}}{(n-1)d}$. The corresponding feasible translational velocity of the box is $\dot{x} = \frac{V_{max}}{(n-1)d}(h_{1y} - h_{2y})$ and $\dot{y} = \frac{V_{max}}{(n-1)d}(h_{1x} - h_{2x})$. The feasible triangle is located with its upper vertex on the horizontal line: $\dot{\theta}_{max} = \frac{2V_{max}}{(n-1)d}$.

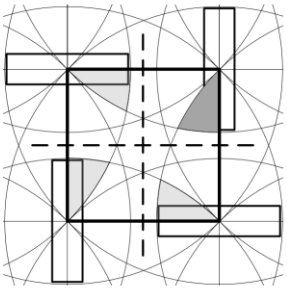
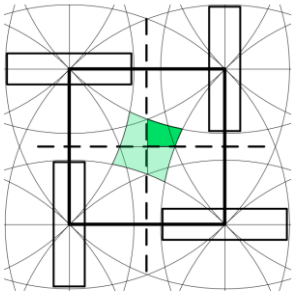
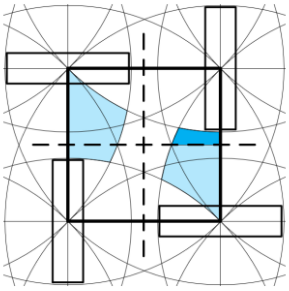
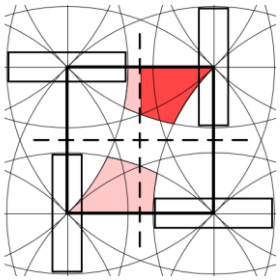
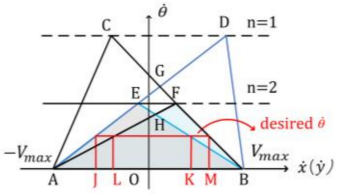
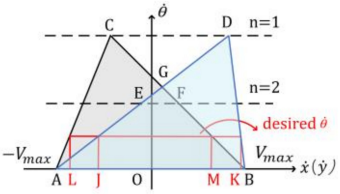
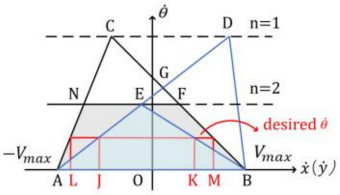
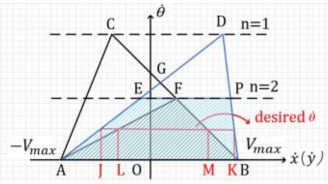
Based on Equations (22)–(25), it is seen that if the upper vertex of the feasible triangle is approaching left, the location of the box center is approaching up to the horizontal wheel and left to the vertical wheel. On the contrary, if the upper vertex of the feasible triangle is approaching right, the location of the box center is approaching down to the horizontal wheel and right to the vertical wheel. If the feasible triangle is right located in the middle, i.e., the upper vertex is located on the vertical axis, the box center is right located in the center of the four-wheel square. To extremes, if the box center is moved to the same location as the vertical or horizontal wheels, i.e., $h_{ix} = h_{iy} = 0$ $(i = 1, 2)$, the upper vertex of the feasible triangle is located on the point Q or S, as indicated in red in Figure 19a.

From the discussion of previous sections, the four-wheel pattern only occurs in particular areas of the overall workspace. If there is an upper horizontal wheel being covered, the feasible area for horizontal velocity $\dot{x}$ of the box ΔABC reduces to ΔABF. The added horizontal wheel has no effect on the feasible area for the vertical velocity $\dot{y}$, which still is ΔABD. If the box center is moving across the boundary, which causes the wheel pattern to

turn from the four-wheel pattern to the four-wheel with one more upper horizontal wheel pattern, the desired velocity within the area ΔAFC will no longer be feasible.

Clearly, different locations of the box center result in different wheel patterns. From the above findings, the feasible area for different patterns may also become different. In this work, the feasible area is summarized in Table 1. Here, we only present the case when the box radius is equal to $\sqrt{2}d$. Row two of Table 1 refers to the feasible area for desired velocities for the corresponding area in row one in dark color, and other identical areas are indicated in light color.

Table 1. Feasible areas for desired velocities.

							
							
$\dot{x}$	$\dot{y}$	$\dot{x}$	$\dot{y}$	$\dot{x}$	$\dot{y}$	$\dot{x}$	$\dot{y}$
ABF	ABE	ABC	ABD	ABFN	ABE	ABF	ABPE
$\overline{LM}$	$\overline{JK}$	$\overline{LM}$	$\overline{JK}$	$\overline{LM}$	$\overline{JK}$	$\overline{LM}$	$\overline{JK}$

It is seen from Table 1 that the feasible areas in the second row correspond to box locations within quadrant I (In Figure 19b, the box is located within quadrant II) such that the horizontal and vertical velocity feasible areas are located to the left and right of the vertical axis, respectively. The feasible areas of desired trajectory velocities for counterclockwise rotational motions are symmetric about the origin of the coordinate frame and thus omitted for simplicity. For the same desired rotational velocity $\dot{\theta}$, the desired horizontal and vertical velocities are within the ranges $\overline{LM}$ and $\overline{JK}$, respectively, as illustrated in the second row of Table 1.

In summary, overall feasible areas for the desired velocities are illustrated in Figure 20, where the meanings of the colors are the same as in Table 1. It is found that if the box is moved along the blue areas, the desired horizontal velocity of the box can be larger than the desired vertical velocity. The desired vertical velocity of the box can be larger than the desired horizontal velocity if the box is moved to the red areas. The desired velocity ranges for the horizontal and vertical directions are the same when the box is moved to areas in green and grey areas. The desired velocity for the green areas can be larger than the grey areas. From Figure 20, it is seen that a time-optimal path exists if the box moves horizontally on the blue and green areas and vertically on the red and green areas. It is noted that because the kinematics of the cellular conveyor are coupled, the feasible areas for the desired velocities are not likely to be depicted in the plane. Rather, it can only be illustrated in the space, making analyses difficult. Due to the decoupled characteristics of the proposed conveyor, the kinematic analyses can be carried out concisely.

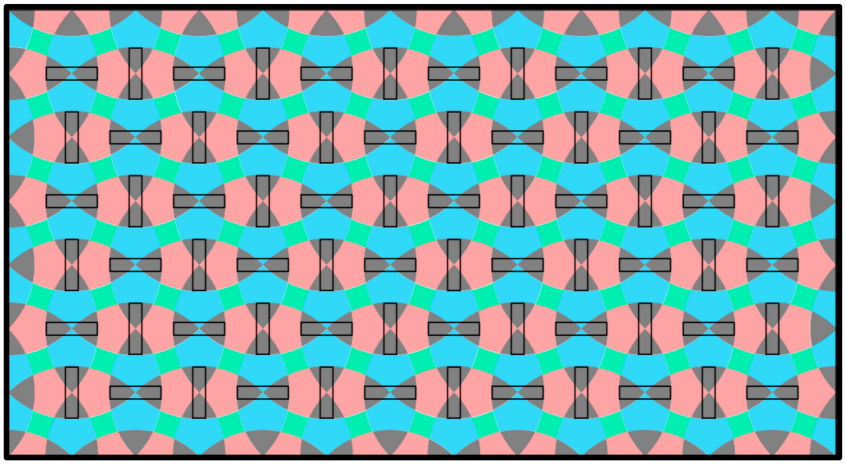


Figure 20. Overall feasible areas under constrained wheel velocities (The green areas refers to the largest range for the desired velocity in both horizontal and vertical directions. The grey areas refers to the smallest range for the desired velocity in both horizontal and vertical directions. In the blue areas, the range of the desired velocity in the horizontal direction is larger than the vertical direction. In the red areas, the range of the desired velocity in the vertical direction is larger than the horizontal direction).

8. Conclusions

In this work, we proposed a new omnidirectional conveyor. A kinematic model was established, and trajectory planning of four example typical trajectories was achieved and verified via simulations. The results are provided to show that the actual trajectories with PID control are well in line with the desired trajectories. In addition, the workspace and wheel patterns for boxes with different sizes were depicted and analyzed. Furthermore, the workspace and wheel patterns for the conveyor with failed wheels were investigated in order to provide guidance for trajectory planning. On the velocity level, the feasible areas for trajectory planning were comprehensively studied, showing that on the velocity level, some areas within the feasible workspace may become infeasible under the constrained wheel rotational velocity. In comparison to the existing cellular counterpart, our proposed conveyor ensures consistent and decoupled kinematics and that fewer wheels are needed, thus saving the cost.

In summary, this paper has three major contributions. Firstly, this paper proposed a new omnidirectional conveyor. Secondly, trajectory planning was achieved and verified using PID control. Thirdly, the paper presents the workspace of the conveyor for boxes with different sizes and feasible areas in the case of wheel failures. Finally, with the assumption of constrained wheel velocity, this work further depicted the feasible areas for the desired box velocities, thus paving the way for trajectory planning and optimal control. Based on this work, our ongoing work will focus on dynamic issues, obstacle avoidance, optimal control, and actual experiments with our proposed omnidirectional conveyor.

Author Contributions: Conceptualization, Z.Z.; methodology, Z.Z. and X.Z.; software, Z.Z., T.S. and Z.W.; validation, Z.Z.; formal analysis, Z.Z.; investigation, T.S.; resources, Z.Z. and T.S.; data curation, Z.Z. and Z.W.; writing—original draft preparation, Z.Z.; writing—review and editing, X.Z. and Z.Z.; visualization, T.S., Z.W. and Z.Z.; supervision, X.Z.; project administration, Z.Z.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the National Natural Science Foundation of China, grant number 52005407.

Data Availability Statement: CoppeliaSim simulation model files and codes can be found at https://github.com/BeginPy/A-new-omnidirectional-conveyor.git (accessed on 25 September 2024).

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

References

1. Firvida, M.B.; Tharner, H.; Freitag, M. Decentralized omnidirectional route planning and reservation for highly flexible material flow systems with small-scaled conveyor modules. In Proceedings of the IEEE 23rd International Conference on Emerging Technologies and Factory Automation, Turin, Italy, 4–7 September 2018.
2. Zaman, M.Q.; Wu, H.M. Intelligent Motion Control Design for an Omnidirectional Conveyor System. *IEEE Access* **2023**, *11*, 47351–47361. [CrossRef]
3. Shabalina, K.; Sagitov, A.; Magid, E. Comparative Analysis of Mobile Robot Wheels Design. In Proceedings of the International Conference on Developments in eSystems Engineering, DeSE, Cambridge, UK, 2–5 September 2018.
4. Song, J.-B.; Byun, K.-S. Design and control of a four-wheeled omnidirectional mobile robot with steerable omnidirectional wheels. *J. Robot. Syst.* **2004**, *21*, 193–208. [CrossRef]
5. Cofaru, N.F.; Rizescu, C.I.; Pleşea, A.I.; Rizescu, D.; Inţă, M. Modular Transport and Sorting System with Omnidirectional Wheels. *MATEC Web Conf.* **2021**, *343*, 08011.
6. Uriarte, C.; Asphandiar, A.; Thamer, H.; Benggolo, A.; Freitag, M. Control strategies for small-scaled conveyor modules enabling highly flexible material flow systems. *Procedia CIRP* **2019**, *79*, 433–438. [CrossRef]
7. Sint, T.H. Design and Implementation of Automatic Omnidirectional Conveyor Belt System. Master's Thesis, National University of Singapore, Singapore, 2021.
8. Fedorko, G.; Liptai, P.; Molnár, V. Proposal of the methodology for noise sources identification and analysis of continuous transport systems using an acoustic camera. *Eng. Fail. Anal.* **2018**, *83*, 30–46. [CrossRef]
9. Gaiardelli, S.; Carra, D.; Spellini, S.; Fummi, F. Dynamic Job and Conveyor-Based Transport Joint Scheduling in Flexible Manufacturing Systems. *Appl. Sci.* **2024**, *14*, 3026. [CrossRef]
10. Kruhn, T.; Overmeyer, L. Decentralized and dynamic routing for a cognitive conveyor. In Proceedings of the 1st Joint International Symposium on System-Integrated Intelligence, Wollongong, Australia, 9–12 July 2013.
11. Kim, K.; Hong, Y. Industrial general reinforcement learning control framework system based on intelligent edge. In Proceedings of the 2020 22nd International Conference on Advanced Communication Technology (ICACT), Phoenix Park, Republic of Korea, 16–19 February 2020.
12. Abu-Radeh, Y.; Shahateet, N.; Tarwa, S.; Tahbob, R.; Takrouri, W. Omni-Directional Sorting Conveyor. Bachelor's Thesis, Palestine Polytechnic University, Hebron-Palestine, Palestinian Territory, 1 May 2023.
13. Zhou, Z.; Zhang, H.; Liu, K.; Ma, F.; Lu, S.; Zhou, J.; Ma, L. Design of a Two-Dimensional Conveyor Platform with Cargo Pose Recognition and Adjustment Capabilities. *Sensors* **2023**, *23*, 8754. [CrossRef] [PubMed]
14. Salazar, E.B.; Escudero, B.N.; Minango, S.N.R. Omnidirectional transport system for classification and quality control using artificial vision. In Proceedings of the 3rd International Conference on Virtual Augmented Reality Simulations, Perth, Australia, 23–25 February 2019.
15. Gorner, M.; Benedikt, F.; Grimmel, F.; Hulin, T. SwarmRail: A Novel Overhead Robot System for Indoor Transport and Mobile Manipulation. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020.
16. Youssef, A.W.; Elhusseiny, N.M.; Shehata, O.M.; Shihata, L.A.; Azab, E. Kinematic modeling and control of omnidirectional wheeled cellular conveyor. *Mechatronics* **2022**, *87*, 102896. [CrossRef]
17. Keek, J.S.; Loh, S.L.; Hanafi, A.N.; Cheong, T.H. Pre-slippage detection and counter-slippage for e-pattern omniwheeled cellular conveyor. *Bull. Electr. Eng. Inform.* **2024**, *13*, 2298–2309. [CrossRef]
18. Keek, J.S.; Loh, S.L.; Chong, S.H. Design and Control System Setup of an E-Pattern Omniwheeled Cellular Conveyor. *Machines* **2021**, *9*, 43. [CrossRef]
19. Uriarte, C.; Kunaschk, S. Omnidirectional Conveying System Module, Modular Omnidirectional Conveying System and Omnidirectional Conveying System. *BIBA* **2014**, *submitted*.
20. Tao, S.; Youpan, Z.; Hui, Z.; Pei, W.; Yongguo, Z.; Guangliang, L. Three-wheel Driven Omnidirectional Reconfigurable Conveyor Belt Design. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019.
21. Yaseen Ismael, O.; Hedley, J. Analysis, Design, and Implementation of an Omnidirectional Mobile Robot Platform. *Am. Sci. Res. J. Eng.* **2016**, *22*, 195–209.
22. Liu, Y.; Zhu, J.J.; Williams, R.L., II; Wu, J. Omni-directional mobile robot controller based on trajectory linearization. *Robot. Auton. Syst.* **2008**, *56*, 461–479. [CrossRef]
23. Galgamuwa, G.I.R.K.; Liyanage, L.K.G.; Ekanayake, M.P.B.; Samaranayake, B.G.L.T. Simplified controller for three wheeled omni directional mobile robot. In Proceedings of the IEEE 10th International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 18–20 December 2015.
24. Zhang, H.; Zhou, Z.; Lu, S.; Wang, H.; Yan, B.; Si, X. Design of reconfigurable transfer palletizing platform based on omnidirectional wheels. In Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 25–27 November 2022.
25. Zaher, W.; Youssef, A.W.; Shihata, L.A.; Azab, E.; Mashaly, M. Omnidirectional-Wheel Conveyor Path Planning and Sorting Using Reinforcement Learning Algorithms. *IEEE Access* **2022**, *10*, 27945–27959. [CrossRef]
26. Bi, Z.M.; Wang, L. Dynamic control model of a cobot with three omni-wheels. *Robot. Comput.-Integr. Manuf.* **2010**, *26*, 558–563. [CrossRef]

27. Sanghai, R.A.; Saundalkar, P.P.; Mallick, J.A.; Shah, B. Sort X Consignment Sorter using an Omnidirectional Wheel Array for the Logistics Industry. In Proceedings of the 2020 International Conference on Convergence to Digital World—Quo Vadis (ICCDW), Mumbai, India, 18–20 February 2020.
28. Wolfe, K.C.; Kutzer, M.D.M.; Armand, M.; Chirikjian, G.S. Trajectory generation and steering optimization for self-assembly of a modular robotic system. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010.
29. Feng, S.; Liu, Y.; Pressgrove, I.; Ben-Tzvi, P. Autonomous Alignment and Docking Control for a Self-Reconfigurable Modular Mobile Robotic System. *Robotics* **2024**, *13*, 81. [CrossRef]
30. Li, W.; Yang, C.; Jiang, Y.; Liu, X.; Su, C.-Y. Motion planning for omnidirectional wheeled mobile robot by potential field method. *J. Adv. Transp.* **2017**, *2017*, 4961383. [CrossRef]
31. Stampa, M.; Röhrig, C.; Künemund, F.; Heß, D. Estimation of Energy Consumption on Arbitrary Trajectories of an Omnidirectional Automated Guided Vehicle. In Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, Poland, 24–26 September 2015.
32. Cservenak, A. Path and trajectory planning for an automated carrier vehicle equipped with two conveyor belts used in manufacturing supply. *Manuf. Technol.* **2021**, *21*, 163–182. [CrossRef]
33. Renny, S.K.; Uchiyama, N.; Sano, S. Real-time smooth trajectory generation for nonholonomic mobile robots using Bézier curves. *Robot. Comput. Integr. Manuf.* **2016**, *41*, 31–42. [CrossRef]
34. Craig, J. *Introduction to Robotics: Mechanics and Control*, 3rd ed.; Pearson Education International: London, UK, 2006.

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.