11-25-2023

Efficient Navigation and Motion Control for Autonomous Forklifts in Smart Warehouses: LSPB Trajectory Planning and MPC Implementation

Konchanok Vorasawad
Pukyong National University

Myoungkuk Park
Michigan Technological University, mpark2@mtu.edu

Changwon Kim
Pukyong National University

Recommended Citation

Vorasawad, K., Park, M., & Kim, C. (2023). Efficient Navigation and Motion Control for Autonomous Forklifts in Smart Warehouses: LSPB Trajectory Planning and MPC Implementation. *Machines, 11*(12). http://doi.org/10.3390/machines11121050
Retrieved from: https://digitalcommons.mtu.edu/michigantech-p2/303

Article

Efficient Navigation and Motion Control for Autonomous Forklifts in Smart Warehouses: LSPB Trajectory Planning and MPC Implementation

Konchanok Vorasawad [1], Myoungkuk Park [2] and Changwon Kim [3,*]

1 School of Mechanical Design Engineering, Pukyong National University, Busan 48513, Republic of Korea; konchanok@pukyong.ac.kr
2 Department of Mechanical Engineering-Engineering Mechanics, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931-1295, USA; mkpark@mtu.edu
3 School of Mechanical Engineering, Pukyong National University, Busan 48513, Republic of Korea
* Correspondence: ckim@pknu.ac.kr

Abstract: The rise of smart factories and warehouses has ushered in an era of intelligent manufacturing, with autonomous robots playing a pivotal role. This study focuses on improving the navigation and control of autonomous forklifts in warehouse environments. It introduces an innovative approach that combines a modified Linear Segment with Parabolic Blends (LSPB) trajectory planning with Model Predictive Control (MPC) to ensure efficient and secure robot movement. To validate the performance of our proposed path-planning method, MATLAB-based simulations were conducted in various scenarios, including rectangular and warehouse-like environments, to demonstrate the feasibility and effectiveness of the proposed method. The results demonstrated the feasibility of employing Mecanum wheel-based robots in automated warehouses. Also, to show the superiority of the proposed control algorithm performance, the navigation results were compared with the performance of a system using the PID control as a lower-level controller. By offering an optimized path-planning approach, our study enhances the operational efficiency and effectiveness of Mecanum wheel robots in real-world applications such as automated warehousing systems.

1. Introduction

Recently, production and manufacturing sites are becoming increasingly intelligent, starting with the 4th Industrial Revolution. These industrial sites are called smart factories and are gradually being developed based on various components such as 5G communications, cloud computing, artificial intelligence, cyber physics systems, and intelligent robots [1,2]. Among various robotic systems, mobile robots play an important role to satisfy the demand of saving time, reducing cost, and the risk of harm that may affect humans. In particular, in many warehouses, mobile robots are used as both manipulators and autonomous mobile robots, since they are convenient and respond to the needs of users or operators in the working area.

In this research, we are focusing on an autonomous forklift. The market analysis report shows that the current application of autonomous forklifts has increased since 2018 and is expected to increase until 2030, gradually [3]. In order to improve the operating efficiency of autonomous forklifts and increase their usability in narrow spaces, many products have a driving part that can move in all directions [4]. The Mecanum wheel mobile robot (MWR) platform has a type of wheel that allows the robot to move in omni-directions through four independently rotating Mecanum wheels [5–7].

The purpose of navigation is to enable the mobile platform to safely and efficiently move to a given destination. Path planning and motion control are essential for navigation of the MWR. First of all, in the robotics research field, various path-planning algorithms were developed and used for mobile robots, such as probabilistic Roadmaps (PRM) (RRT), (RRT*). The (A*) path planning algorithm was chosen to experiment on the mechatronic design, low-level control and high-level control of an indoor Four Omni-Wheeled Mobile Robot (4OWMR) in [8]; the performance and application of many path-planning algorithms such as the Dijkstra algorithm, A*, D*, LPA*, and D* lite algorithm were compared and the advantages and disadvantages of different algorithms were also described in [9], and choosing an algorithm suitable for the usage and workspace was discussed in [10].

Once the path planning of the mobile robot is completed, the waypoints for the robot to pass through are determined. In this study, the focus is on constructing a trajectory for the target movement while passing through waypoints determined by considering the maximum velocity and acceleration performance of the robot's driving part. The problem of reducing the final operating time by optimizing the movement of the robot is directly related to the performance of the robot. Therefore, the Linear Segment with Parabolic Blends (LSPB) was applied as a method to optimize the robot's movement. The LSPB algorithm was used in path planning for an industrial robot, such as a method of the typical on-line minimum time trajectory planning for robot manipulators [11], through industrial robot manipulators, and applying LSPB can maximize the time trajectory and generation [12].

After the robot's movement path is planned, motion control is essential to execute it. For this purpose, model predictive control is applied. Through exploring trends in research, the application of Model Predictive Control (MPC) in various domains becomes evident, as indicated in [13], which analyses system control engagement trends in the industry and in the research field, and in [14], which shows the survey of available MPC technology, both linear and nonlinear. The MPC is also a control technique used in many systems, such as heating, ventilation, and air conditioning (HVAC) systems [15], power electronic systems [16], and is extensively employed in the field of robotics, as in [17] which used MPC to control the MWR for path-following problems. MPC was used in robot navigation among obstacles through the simulation of MWR [18], and Ref. [19] simulated the robot trajectory tracking on a robot operating system based on an MPC controller to calculate the future speed information.

This paper proposes a path-planning method for the MWR for efficient operation in a warehouse. The advantage of the MWR is that it can move in various directions in an uncomplicated manner. We are adopting the part of robot path planning by LSPB to develop the automated warehouse application. The LSPB algorithm defines the robot's velocity and acceleration. When moving from the start point to the endpoint, which depends on the customer's order, the robot moves at a defined velocity that considers the operating environment and the moving distance of the robot, in that order. Mecanum wheel kinematics was derived to apply velocity and acceleration to control the robot's movement. The proposed trajectory planning algorithm on the MWR is demonstrated by a MATLAB simulation in a warehouse-like environment.

This paper is structured as follows: Section 2 outlines the navigation algorithm proposed in this paper, including trajectory planning, robot kinematics, and motion control. Section 3 discusses the inverse kinematics and forward kinematics of MWR, and Section 4 designs the LSPB for MWR application. The final step of the navigation algorithm, MPC-based motion control, is discussed in Section 5. The experimental results will be discussed in Section 6. Finally, the findings will be summarized in the conclusion section.

2. Problem Statement

In order to accomplish smart warehouses, a range of core technologies to automate and optimize their operations is needed. These technologies include: IoT (Internet of Things), automation and robotics, warehouse management systems, machine learning and AI, augmented reality (AR), Big Data and Analytics, cloud computing, 5G and edge computing, blockchain, security and access control [20].

Among various components of smart factories [21], this study focuses on logistics systems based on autonomous mobile robots (AMRs). In particular, once the work schedule is determined in the production management system, the AMR control system (ACS) allocates tasks and plans paths to carry out the schedule. This is the process of performing the corresponding mission when it is delivered to each AMR. In this study, we developed a velocity–trajectory plan for controlling robot actuators to efficiently travel the waypoint-based path transmitted from ACS and a lower-level controller to implement it.

This paper proposes a trajectory planning and control method for unmanned fork-lifts for application in smart automated warehouses. The purpose of smart automated warehouses is to increase productivity through the application of automation systems and efficient use of space. Therefore, in an automated warehouse, the spacing between racks storing materials must be optimized so that forklifts must move in a relatively narrow space. To overcome these difficulties at work sites, this paper proposes a driving trajectory generation method suitable for moving in narrow spaces considering the omnidirectional movement characteristics of a Mecanum wheel-based forklift. Therefore, Mecanum wheels, which can move in all directions, were selected as the driving type so that the forklift can move smoothly even in narrow spaces. However, in existing studies, the driving velocity and acceleration performance of the robot were not considered in the driving control of the Mecanum wheel robot, so there was a limit to efficient driving control. Therefore, in this study, the modified LSPB method was applied to plan the robot's trajectory, considering mobile robot specifications such as the robot's maximum velocity and acceleration performance while utilizing the characteristics of the Mecanum wheel.

LSPB is a good trajectory method for MWR due to the following advantages. Firstly, LSPB provides a smooth transition between different segments of the trajectory, minimizing abrupt changes in velocity or direction. This is crucial for Mecanum wheels, as their omni-directional movement capabilities rely on precise and continuous control. Secondly, the smooth acceleration and deceleration profile contributes to energy efficiency in the robot's movement, because the control inputs added for robot control are also calculated gradually. Finally, most importantly, LSPB offers the advantage of simplicity and ease of implementation. The mathematical models for LSPB are well understood and computationally efficient, making real-time trajectory adjustments feasible. This simplicity is especially valuable in dynamic environments where quick responses to changing conditions are essential, such as in smart warehouses [22,23].

Through LSPB path planning, the robot gradually accelerates its velocity when starting and decelerates when it reaches the destination, eliminating sudden acceleration and deceleration situations, enabling stable driving.

The main contribution of this study is to plan the robot's trajectory using LSPB for efficient operation of MWR, as shown in Figure 1, and to design an MPC-based motion controller through inverse kinematics and forward kinematics of MWR to implement the trajectory.
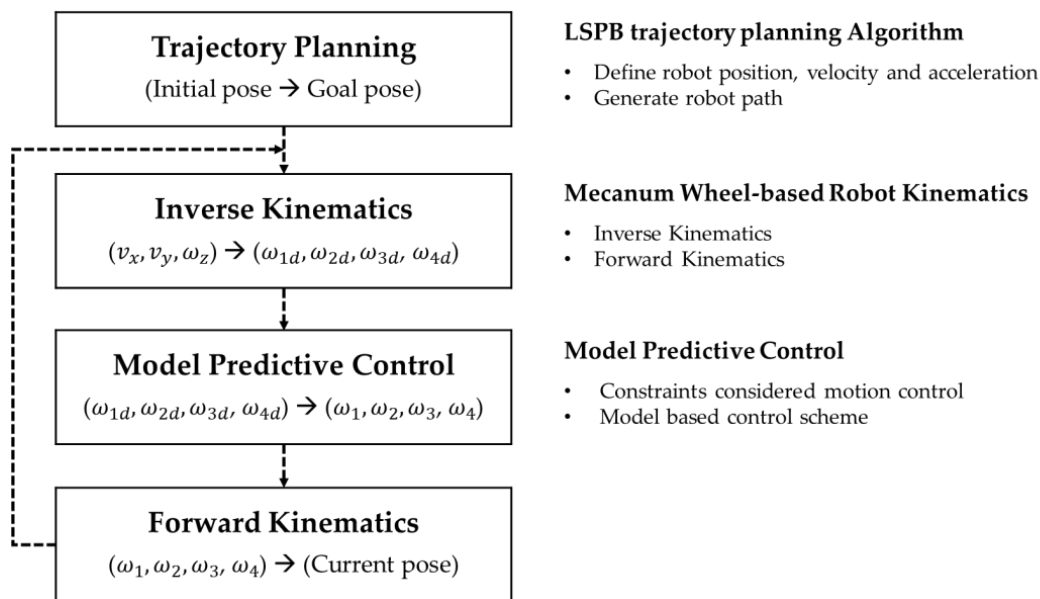
Figure 1. Mecanum wheel-based robot navigation structure.

3. Mecanum Wheel-Based Robot Kinematics

In this section, we will first discuss the kinematics of MWR among the three modules of the MWR navigation algorithm. This section includes the Mecanum wheel-base mobile robot's configuration and kinematics.

3.1. Robot Configuration

Figure 2 shows the robot configuration. It describes the coordinate system and moment vector of the MWR, where X_R, Y_R are the robot's coordinate system, O_R is the center of the robot, P_i represents the center position of each wheel, and X_i and Y_i are the coordinate system of the ith wheel. Also, the robot's parameters can be specified as follows (Table 1):
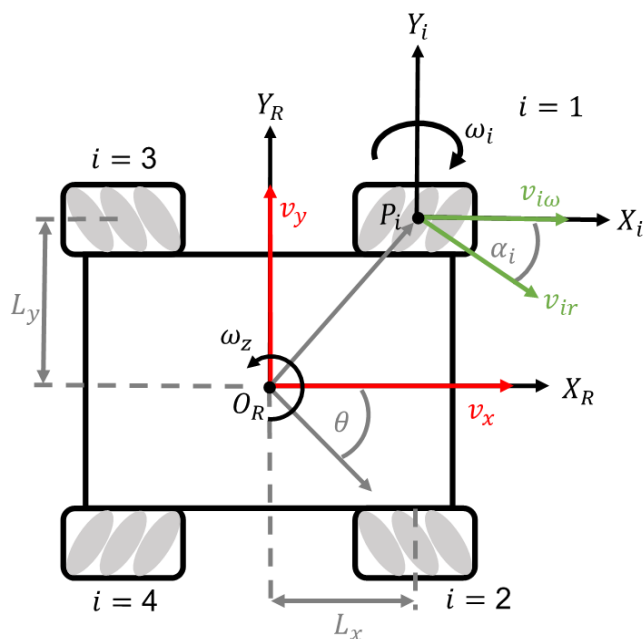

Figure 2. Robot configuration.

Table 1. Mecanum wheel robot's parameters.

Symbol	Parameter	Unit
r	Mecanum wheel radius	m
L_x	The distance between the wheel and the robot's center in the x (longitudinal) axis	m
L_y	The distance between the wheel and the robot's center in the y (lateral) axis	m
v_x	The robot's linear velocity in the x-axis	m/s
v_y	The robot's linear velocity in the y-axis	m/s
ω_z	The robot's angular velocity along the z-axis at the center point O_R	rad/s
$i = (1, 2, 3, 4)$	Wheel numbers	-
v_i	Linear velocity of each wheel	m/s
v_{ir}	Velocity of Mecanum roller direction of each wheel	m/s
α_i	Mecanum roller angle	rad

Utilizing the above configurations of MWR, inverse kinematics and forward kinematics of MWR are derived.

3.2. Inverse Kinematics and Forward Kinematics

Inverse kinematics helps to calculate joint variables as the control target when the target velocity vectors are derived from LSPB. In MWR, the angular velocities of each wheel are calculated from the robot's velocity by means of inverse kinematics. To derive inverse kinematics of MWR, we defined rotational velocity of wheels, $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$, and the velocity of MWR, $V_R = \begin{bmatrix} v_x & v_y & \omega_z \end{bmatrix}^T$ which is the generalized linear velocity of the robot's center in the Cartesian coordinate system. The relation between ω and V_R is defined by the Jacobian matrix J. Then, assuming non-slip condition, the inverse kinematics of MWR is given [24,25]:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & \frac{-(L_x + L_y)}{2} \\ 1 & 1 & \frac{L_x + L_y}{2} \\ 1 & 1 & \frac{-(L_x + L_y)}{2} \\ 1 & -1 & \frac{L_x + L_y}{2} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} = J \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix}, \tag{1}$$

where J represents the inverse kinematic Jacobian matrix of the MWR system. Then, the forward kinematics of MWR can be derived as follows:

$$\begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} = J^+ \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}. \tag{2}$$

Since the form of the inverse kinematics Jacobian matrix in Equation (1) is not square, the pseudo-inverse skill must be applied. Therefore, J^+ satisfies the following: $J^+ J = I$ and $J^+ = (J^+ J)^{-1} J^T$. Through this process, the forward kinematics of MRW can be modified as follows:

$$\begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ \frac{-1}{L_x + L_y} & \frac{1}{L_x + L_y} & \frac{-1}{L_x + L_y} & \frac{1}{L_x + L_y} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}. \tag{3}$$

Forward kinematics helps to convert joint variables to Cartesian variables. From this, the robot's velocity can be derived from the angular velocities of each wheel. Figure 3 shows the robot's motion according to the control of each MWR wheel. The red arrow

indicates the robot's motion [26]. As shown in Figure 3, independent control of each wheel of the MWR enables omnidirectional movement of the robot.
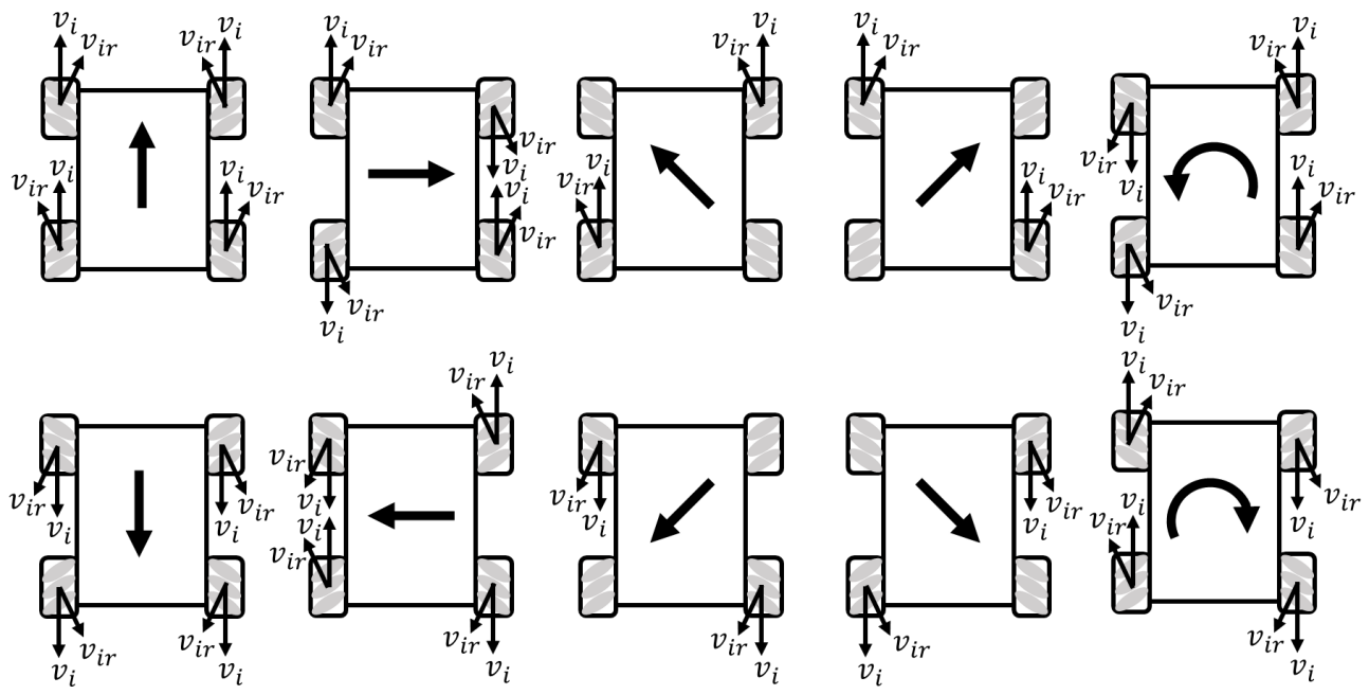
Figure 3. Motion of robotic platform equipped with four Mecanum wheels.

4. Trajectory Planning

This section introduces an algorithm for creating a trajectory that can efficiently move MWR without collisions between given waypoints within a smart warehouse. Proposing a trajectory planning method involves establishing a seamless path connecting its initial state to the goal state. This trajectory is determined while adhering to various limitations and circumstances. For instance, it may entail selecting the quickest path between two endpoints, minimizing the time required for the navigation, or ensuring safe traversal to the destination while avoiding collisions [27,28]. For decades, numerous path-planning algorithms have been developed and employed to address various environments. Each algorithm having distinct advantages allows the algorithms to be applied in different problem-solving scenarios [10,29].

While trajectory-planning a path, specifying how to traverse that path with respect to time is necessary [30]. In this study, an efficient trajectory planning method was developed after path planning was completed and waypoints were defined. For that, the LSPB (linear segments with parabolic blends)-based trajectory generation skill was employed. LSPB is a trajectory planning method formulated to increase velocity until it reaches the intended velocity gradually. Subsequently, as it approaches the designated goal position, the velocity gradually decreases. For this reason, the LSPB path-planning algorithm makes smoother and less intimidating movements, improving safety and user acceptance [31]. Therefore, applying LSPB to autonomous forklift robots can achieve efficient movement in the given workspace.

4.1. Linear Segments with Parabolic Blends (LSPB) Theory

LSPB is a method that uses velocity and acceleration to calculate the trajectory connecting the given start and end points. This feature is that it plans a movement path that applies the maximum velocity and acceleration/deceleration performance of the robot used for a specific task. Through this, a path that can actually be implemented through a robot and a velocity profile to achieve it are created, which in turn contributes to the implementation of

efficient navigation. Therefore, the LSPB can define the trajectory that satisfies the desired velocity (V_c) and acceleration (a_c). In LSPB, the whole trajectory is divided into three main segments. The first segment is a quadratic polynomial that starts with an initial velocity of zero and reaches the desired velocity (V_c) at blend time (t_b). The second segment is a constant-velocity zone. The trajectory will be defined along with the desired velocity (V_c). The last segment is also a quadratic polynomial where the velocity falls to zero when reaching the endpoint within traveling time (t_f) [32], as shown in Figure 4. By means of LSPB, the trajectory of position, velocity, and acceleration are derived, as shown in Figure 4.
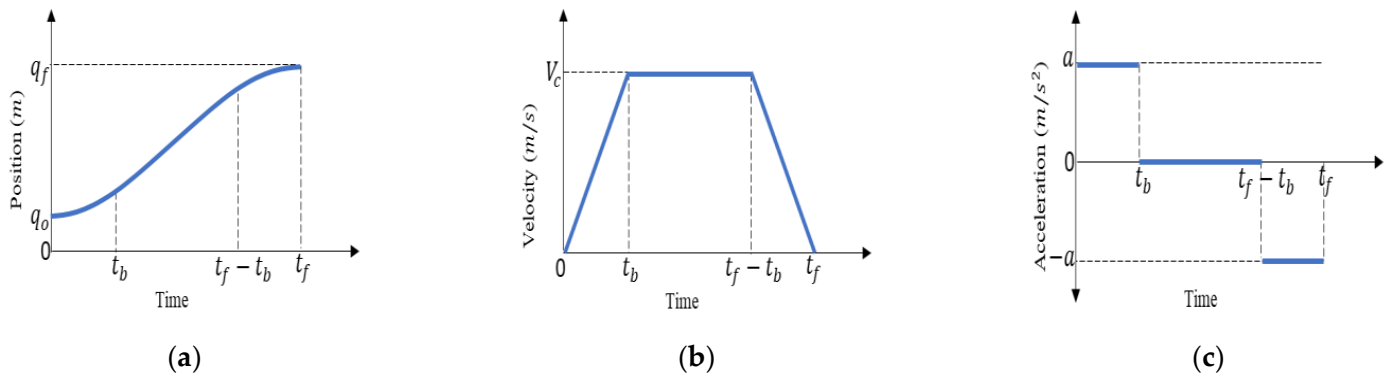
Figure 4. LSPB trajectory. (**a**) Position profile of trapezoidal velocity profile. (**b**) Velocity profile of trapezoidal velocity profile. (**c**) Acceleration profile of trapezoidal velocity profile.

The following equations calculate the position, velocity, and acceleration trajectory connecting the start and end points by considering the maximum velocity, required driving velocity, and deceleration performance.

$$s(t) = \begin{cases} s_i + \frac{1}{2}a_c t^2 & (0 \le t \le t_b) \\ s_i - \frac{V_c t_b}{2} + V_c t & \left(t_b \le t \le t_f - t_b\right) \\ s_f - \frac{a_c}{2}t_f^2 + a_c t_f t - \frac{a_c}{2}t^2 & \left(t_f - t_b \le t \le t_f\right) \end{cases} \tag{4}$$

$$v(t) = \begin{cases} a_c t & (0 \le t \le t_b) \\ V_c & \left(t_b \le t \le t_f - t_b\right) \\ a_c t_f t - at & \left(t_f - t_b \le t \le t_f\right) \end{cases} \tag{5}$$

$$a(t) = \begin{cases} a_c & (0 \le t \le t_b) \\ 0 & \left(t_b \le t \le t_f - t_b\right), \\ -a_c & \left(t_f - t_b \le t \le t_f\right) \end{cases} \tag{6}$$

where q_i is the initial position, q_f is final position, V_c is velocity, a_c is acceleration, t_b is blend time, and t_f is final time [33,34]. However, there may occur a 'jerk' phenomenon that occurs in the trapezoidal speed profile. The s-curved spline was applied at the vertex section of the speed profile to eliminate the jerk phenomenon [35,36]. We used the s-curved spline method by naming the modified LSPB (M-LSPB) and suggest a way to implement it in the autonomous forklift. The corresponding trajectories are shown in Figure 5.
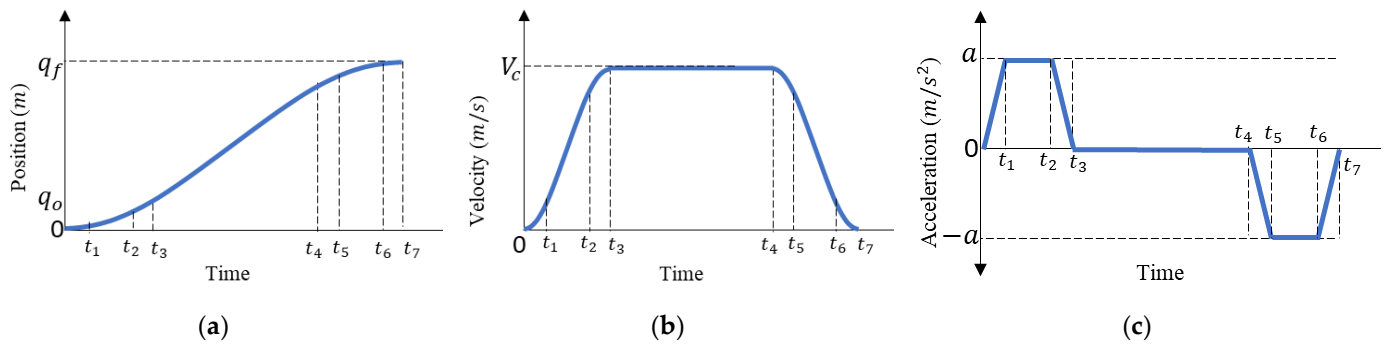
Figure 5. M-LSPB trajectory. (**a**) Position profile of s-curve velocity profile. (**b**) Velocity profile of s-curve velocity profile. (**c**) Acceleration profile of s-curve velocity profile.

The detailed equation are given as follows:

$$a(t) = \begin{cases} j_c t & (0 \le t \le t_1) \\ j_c T_1 & (t_1 \le t \le t_2) \\ j_c T_1 - j_c(t - t_1) & (t_2 \le t \le t_3) \\ 0 & (t_3 \le t \le t_4) \\ -j_c(t - t_4) & (t_4 \le t \le t_5) \\ -j_c T_4 & (t_5 \le t \le t_6) \\ -j_c T_4 + j_c(t - t_6) & (t_6 \le t \le t_7) \end{cases} \tag{7}$$

$$v(t) = \begin{cases} \frac{1}{2} j_c t^2 & (0 \le t \le t_1) \\ v_1 + j_c T_1(t - t_1) & (t_1 \le t \le t_2) \\ v_2 + j_c T_1(t - t_2) - \frac{1}{2} j_c(t - t_2)^2 & (t_2 \le t \le t_3) \\ v_3 & (t_3 \le t \le t_4) \\ v_4 - \frac{1}{2} j_c(t - t_4)^2 & (t_4 \le t \le t_5) \\ v_5 - j_c T_5(t - t_5) & (t_5 \le t \le t_6) \\ v_6 - j_c T_5(t - t_6) + \frac{1}{2} j_c(t - t_6)^2 & (t_6 \le t \le t_7) \end{cases} \tag{8}$$

$$s(t) = \begin{cases} v_0 t + \frac{1}{6} j_{cx} t^3 & (0 \le t \le t_1) \\ s_1 + v_1(t - t_1) + \frac{1}{2} j_c T_1(t - t_1)^2 & (t_1 \le t \le t_2) \\ s_2 + v_2(t - t_2) + \frac{1}{2} j_c T_1(t - t_2)^2 - \frac{1}{6} j_c(t - t_2)^3 & (t_2 \le t \le t_3) \\ s_3 + v_3(t - t_3) & (t_3 \le t \le t_4) \\ s_4 + v_4(t - t_4) - \frac{1}{6} j_c(t - t_4)^3 & (t_4 \le t \le t_5) \\ s_5 + v_5(t - t_5) - \frac{1}{2} j_c T_5(t - t_5)^2 & (t_5 \le t \le t_6) \\ s_6 + v_6(t - t_6) - \frac{1}{2} j_c T_5(t - t_6)^2 + \frac{1}{6} j_c(t - t_6)^3 & (t_6 \le t \le t_7) \end{cases} \tag{9}$$

where j_c is defined jerk, $s(t)$ is the initial position, s_i is the position at time t when $t_{i-1} \le t \le t_i$, $v(t)$ is velocity, v_i is the velocity at time t when $t_{i-1} \le t \le t_i$, $a(t)$ is acceleration, and T_i is time phase when $t_{i-1} \le t \le t_i$ [35,36]. Time $t_3 = t_b$ when $t_b = \frac{v}{a(1-j_c)}$ and $t_7 = t_f$ when $t_f = \left| \frac{vt_b - q_0 + q_f}{v} \right|$.

4.2. Modified LSPB-Based Trajectory Planning

Once the waypoints are given, modified LSPB applies the characteristics of MWR to plan a seamless trajectory. If the angle formed by the path is vertical or has a sharper angle, it is inevitable that a typical mobile robot will deviate from the given reference path during the path-following process. However, by using the characteristics of MWR and the

advantages of modified LSPB, the reference path can be successfully followed even in the same situation. Figure 6 shows the modified LSPB-based trajectory planning algorithm.
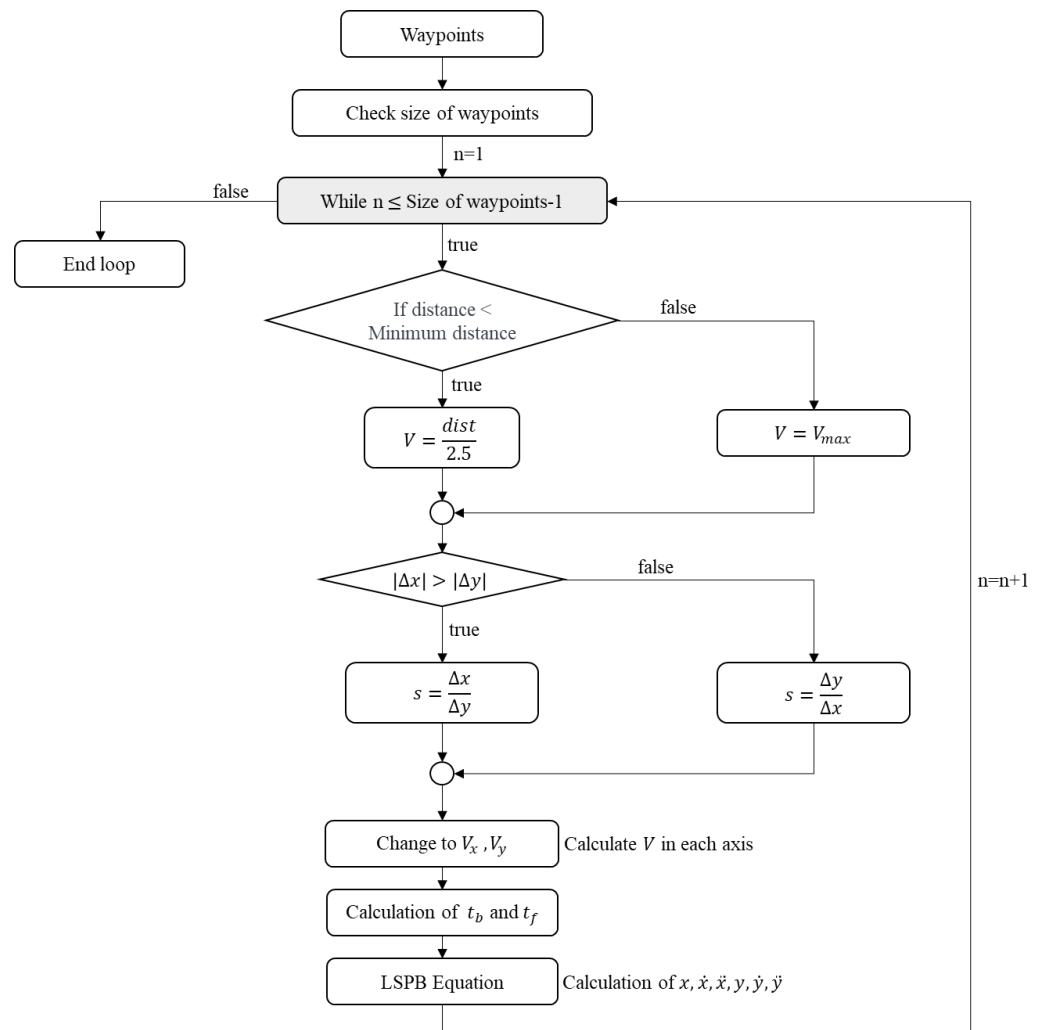

Figure 6. Flowchart of modified LSPB path planning.

First, the given waypoints are analyzed to calculate the number of points (n) that the robot must pass through. A trajectory between two points is planned by modified LSPB, and the same process is repeated until it reaches the final destination. In this research, the maximum velocity and acceleration of robots used in actual logistics sites [37] are taken into consideration and assumed to be $V_{max} = 1.8$ m/s and $a_c = 0.9$ m/s^2, respectively. Considering the robot's maximum velocity and acceleration performance, if the distance between two points is shorter than the minimum distance criterion, the upper limit value of the robot's movement velocity is adjusted to ensure that a reasonable trajectory is calculated. The minimum-distance criterion to which the maximum velocity can be applied is obtained through the condition that there is no constant-speed section and only acceleration and deceleration sections exist. Using the given maximum velocity, acceleration, jerk (j_c), t_b and $dist_{min}$ can be obtained, as in Equations (10) and (11),

$$t_b = \left| \frac{V_c}{(1 - j_c)a_c} \right| \tag{10}$$

$$
\begin{aligned}
dist_{min} &= v_0 + \tfrac{1}{2}a_c t_b^2 + v_0 + a_c t_b - \tfrac{1}{2}a_c \left(t_f - t_b \right)^2 \\
&= 2v_0 + a_c t_b,
\end{aligned}
\tag{11}
$$

where $t_f = 2t_b$. Therefore, the maximum velocity is determined according to the distance between the two points P_1 and P_2, $dist_{P_1 P_2}$, for calculating the trajectory.

$$V_{max} = \begin{cases} v_0 + a_c t_b, & if \ dist_{p_1 p_2} \leq dist_{min} \\ V_{max}, & if \ dist_{p_1 p_2} > dist_{min} \end{cases} \tag{12}$$

Since the pose of a mobile robot is (x, y, θ), in trajectory planning, the path in the direction of the relatively longer coordinate axis is first planned among the directions of coordinates to be moved, and then the path in the other direction is planned.

The distinctive feature of the Mecanum wheel, which enables it to move freely in all directions without rotating, allows for calculating position, velocity, and acceleration, as depicted in Figure 7. Infinity or zero slope value occurs when there is distance in only one axis. For example, in Figure 6, assume the waypoints are P_1 as an initial point and P_2 as endpoint. When P_1 is posed at (x_0, y_0) and P_2 is posed at (x_f, y_f), the slope value condition can be defined as in Equation (12),

$$s = \begin{cases} \frac{\Delta y}{\Delta x}, & if \ \Delta y \geq \Delta x \\ \frac{\Delta x}{\Delta y}, & if \ \Delta x > \Delta y \end{cases} \tag{13}$$

where s is the slope of a linear line, $\Delta x = x_f - x_0$, Δx is the distance in the x-axis, $\Delta y = y_f - y_0$, Δy is the distance in the y-axis.
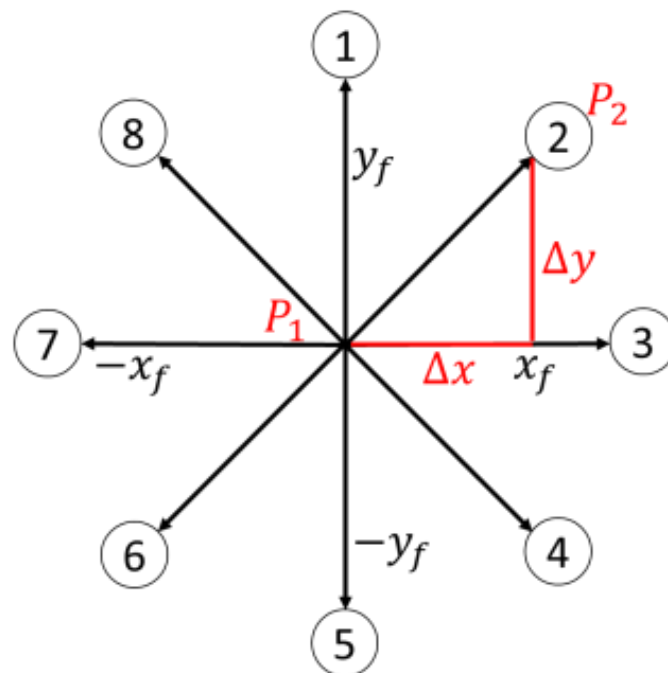
Figure 7. Possible movement of a MWR.

Finally, the displacement, velocity, and acceleration in each coordinate axis are calculated using Equations (4)–(6).

5. Robot Motion Controller

This section presents a motion control method of MWR. Among various control methods, we utilized the model predictive control to generate mobile robot movement. Model predictive control (MPC) is an optimal control method in which the derived control inputs minimize the cost function for a constrained dynamic system [38]. In MPC, a model is used to predict the future behavior of the system over a finite time window. Based on the predictions and the current measured state of the system, the optimal control inputs

are calculated according to the cost function. This method is popular due to the following advantages [39]: (a) proactive control action: the controller is predicting future disturbances, references, etc.; (b) non-linear control: MPC can explicitly consider non-linear systems without linearization; (c) constrained formulation: explicitly consider physical, safety or operational system constraints.

In this research, MWR is controlled to follow the predefined modified LSPB trajectory, especially the velocity profiles such as V_x and V_y. The kinematic model of MWR is utilized as the system model. In the framework of an MPC, the cost function to be minimized by the control input and constraints must be designed. The purpose of the motion controller is to track the LSPB-based path and to minimize energy consumption. Therefore, we utilized three manipulation variables as the manipulation variable constraints, defined as follows:

- Maximum velocity (forward and backward): ± 1.8 m/s
- Maximum acceleration (acceleration and deceleration): ± 0.9 m/s^2
- Maximum rotational velocity: $\pm \frac{\pi}{3}$ rad/s.

The speed error terms for defining the cost function were defined as follows: $e_{v_x} = v_{x_{ref}} - v_x$, $e_{v_y} = v_{y_{ref}} - v_y$, $e_\omega = \omega_{ref} - \omega$. Considering the defined constraints, the cost function of the MPC controller for MWR trajectory tracking was defined as follows:

$$J = w_1 e_{v_x}^2 + w_2 e_{v_y}^2 + w_3 e_\omega^2 + r_1 \Delta u_1^2 + r_2 \Delta u_2^2 + r_3 \Delta u_3^2, \tag{14}$$

where w_1, w_2, w_3, r_1, r_2, and r_3 are the weights, and $\Delta u_1, \Delta u_2$, and Δu_3 are the difference of control variables u_1, u_2, and u_3 when the individual control inputs are $u_1 = V_x$, $u_2 = V_y$, and $u_3 = \omega$.

The proposed controller uses the reference from the LSPB path planning result as the robot's position and provides the robot velocity for kinematics, ensuring it tracks the LSPB-based path. The main parameters of the MPC model applied in this study are summarized in Table 2.

Table 2. MPC controller parameters.

Parameter	Symbol	Value
Prediction horizon	P	10
Control horizon	m	5
Sampling time	Ts	0.01 (s)
Maximum speed	V_{max}	1.8 (m/s)
Speed constraint	$\pm V_{max}$	± 1.8 (m/s)
Maximum acceleration	a_{max}	0.9 (m/s^2)
Acceleration constraint	$\pm a_{max}$	± 0.9 (m/s^2)
Maximum angular speed	ω	$\frac{\pi}{3}$ (m/s)
Angular speed constraint	$\pm \omega$	$\pm \frac{\pi}{3}$ (m/s)
Wheel radius [1]	r	0.133 (m)
Wheel base [1]	$2L_x$	0.762 (m)
Wheel track [1]	$2L_y$	0.610 (m)

[1] The vehicle specification follows RoboMate 10 [40].

6. Simulation and Results

6.1. Simulink Model

We simulated the robot's movement with the MPC control system using references from the calculation of the robot position by the modified LSPB path-planning algorithm via MATLAB Simulink, as shown in Figure 8.
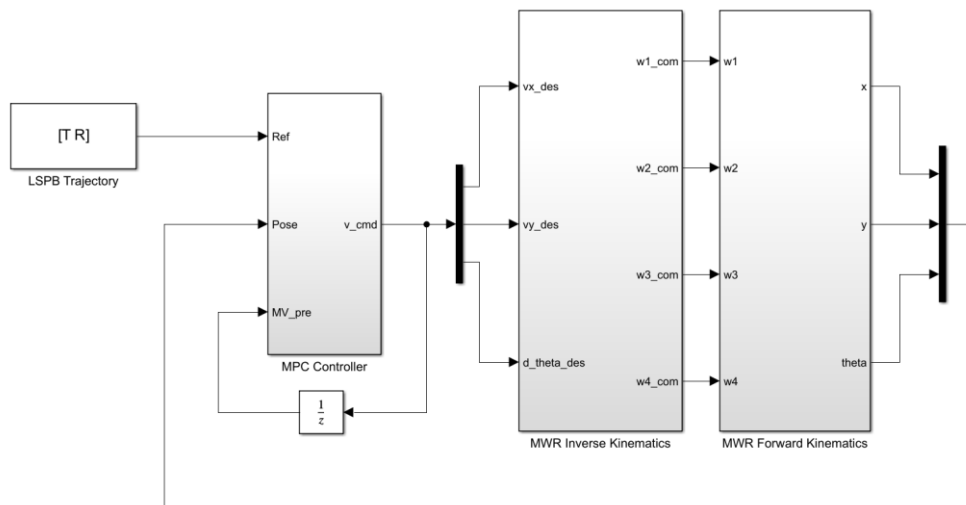
Figure 8. MPC robot control Simulink model.

This Simulink model facilitates the simulation of robot Model Predictive Control (MPC). The process commences with inputs derived from the modified LSPB path-planning outcome, encompassing a time vector and the robot's positional reference. These inputs are directed to the MPC control module. Within this module, velocities are computed for the robot's kinematic system. These calculated velocities are subsequently visualized by pose scope, which displays the robot's positions. To show the superiority of the proposed control algorithm performance, the navigation results are compared with the navigation performance of a system using the PID control as a lower-level controller.

Figure 9 shows the structure of the PID controller. The interior of the first block calculates the robot's longitudinal, lateral, and rotational speeds through a PID control structure and was implemented using the equation below.

$$v_{cmd} = \begin{cases} v_x \\ v_y, \\ \omega \end{cases} \tag{15}$$

$$v_x = k_p e_x + k_d \dot{e}_x + \int e_x dt, \tag{16}$$

$$v_y = k_p e_y + k_d \dot{e}_y + \int e_y dt, \tag{17}$$

$$\omega = k_p e_\theta + k_d \dot{e}_\theta + \int e_\theta dt. \tag{18}$$

The advantage of MPC is that constraints on control variables can be applied. We have also added a constraint on control input to compare performance under the same conditions. To this end, to apply the speed limit, the speed was recalculated through the ratio of longitudinal and lateral speeds as follows:

$$\theta_v = \text{atan2}\left(\frac{v_y}{v_x}\right), \tag{19}$$

$$v_x = v_{lim}\cos(\theta_v), \tag{20}$$

$$v_y = v_{lim}\sin(\theta_v), \tag{21}$$

where θ_v represents the direction of resultant velocity direction of the MWR. In addition, this study presented the performance of the proposed algorithm through simulations in

various scenarios reflecting the characteristics of the work site. To improve the reliability of the simulation, the specifications of an actual unmanned forklift were used in the simulation [40]. Since the Mecanum wheel robot moves by the slipping of the four wheels, it is appropriate to consider the slipping of the wheels in actual environment. However, it is impossible to consider this quantitatively; therefore, simulations were conducted by adding ±5% random noise to the speed of each wheel.
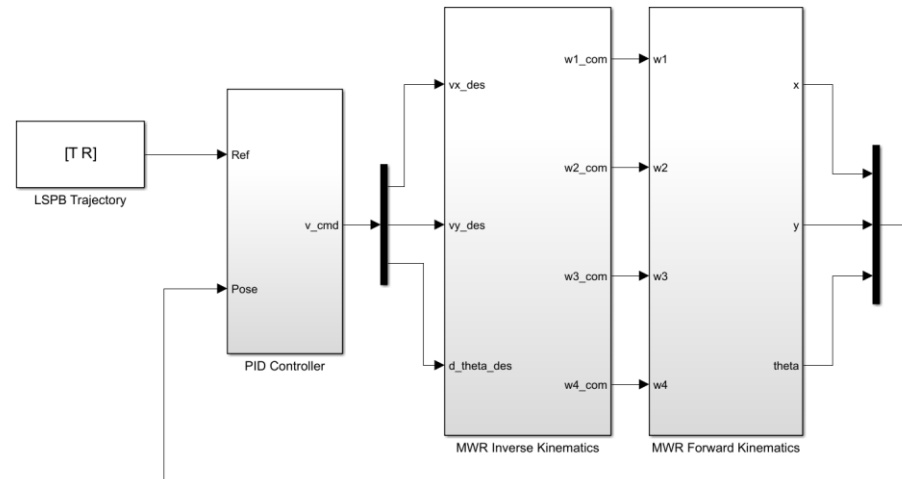
Figure 9. PID robot control Simulink model.

6.2. Simulation Results

We used modified LSPB as the path−planning method and MPC as the motion control strategy. In this section, we present the path-planning and motion control results of three scenarios: (1) Demonstrate key parameters in a rectangular−shaped path, (2) LSPB−PID and MLSPB−MPC comparison in a rectangular-shaped path, and (3) LSPB−PID and MLSPB−MPC comparison in a warehouse-like environment. The characteristic of the proposed algorithm is that it can plan a trajectory that reflects the speed and acceleration of the forklift suitable for the transportation work environment and work object. Therefore, the key parameters of modified LSPB are speed, acceleration, and jerk. For this reason, in the simulations, the position error, speed error, jerk, and work time were compared when the jerk is fixed as a constant and when work was performed at different speeds and accelerations. In addition, the results when applying the proposed modified LSPB compared to the existing LSPB were compared using the same criteria.
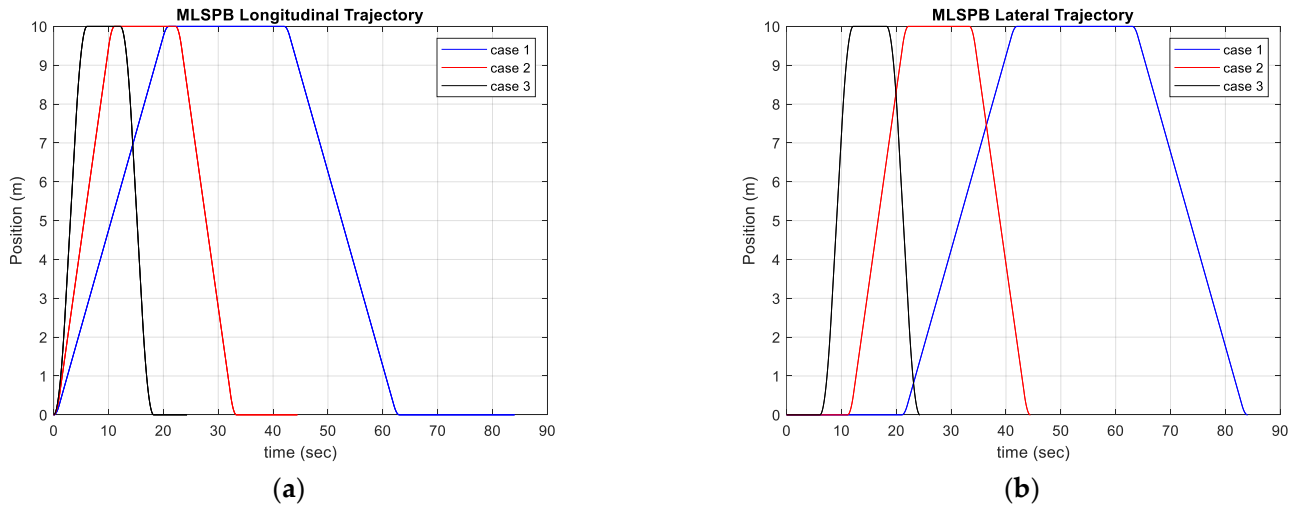
6.2.1. The Key Parameters (Rectangular Shape Path)

The first simulation was conducted to check the performance when changing the speed and acceleration parameters in the proposed algorithm. In the simulation, the jerk was fixed as a constant and the characteristics when changing the maximum speed and maximum acceleration according to the work environment were confirmed. As shown in Figure 10, it can be confirmed that the given path is successfully traversed by the proposed method. We assumed three cases. The first is the case of transporting materials that may explode due to rapid changes in speed. The maximum moving speed is 0.5 m/s, and the maximum acceleration is 0.5 m/s^2. The second is a case where a forklift moves at the speed that a person moves in a typical workplace, and the maximum speed is defined as 1.0 m/s and the maximum acceleration is 0.9 m/s^2. The last case is a case where work is performed in a relatively large space, the risk of collision is low, and the transported material is not affected by speed; here, the speed was defined as 2.5 m/s and the acceleration was 1.2 m/s^2.
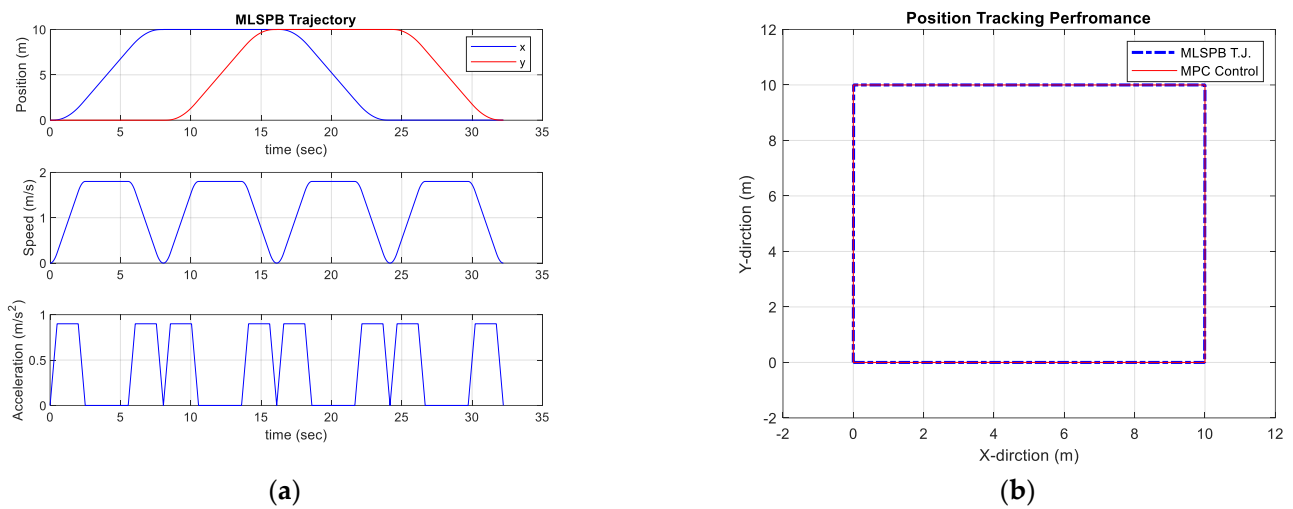
Figure 10. MLSPB path−planning results in a rectangular shape path. (**a**) Path trajectory. (**b**) Planned waypoint for the robot to visit.

Figure 11 shows the movement trajectory of a forklift according to each speed and acceleration condition. As shown in Table 3, it can be seen that not only the work time but also the position error and jerk are affected by changes in speed and acceleration parameters considering the work environment. In reality, these results are self−evident, but the advantage of this research result is that it is possible to plan work that takes into account speed and acceleration according to the work environment.


Figure 11. MLSPB path planning results in a rectangular shape path. (**a**) Path trajectory. (**b**) Planned waypoint for the robot to visit.

Table 3. Work performance according to M−LSPB key parameters.

Speed (m/s)	Acceleration (m/s^2)	Position RSME (m)	Working Time (s)	Average Jerk (m/s^3)
0.5	0.5	0.0210	84.03	0.5837
1.0	0.9	0.0409	44.43	1.1830
2.5	1.2	0.0835	24.31	2.5928

6.2.2. Rectangular Shape Path: LSPB vs. MLSPB

In the following simulation scenario, LSPB−PID and the proposed MLSPB−MPC are compared for the task of moving along a rectangular path. Figure 12 shows the comparison results of navigation performance when LSPB−PID and MLSPB−MPC are applied, respectively. As can be seen in Table 4, when the proposed method is applied, the work time increases slightly, while the RMS position error and average jerk become relatively small, showing the superiority of the proposed algorithm.
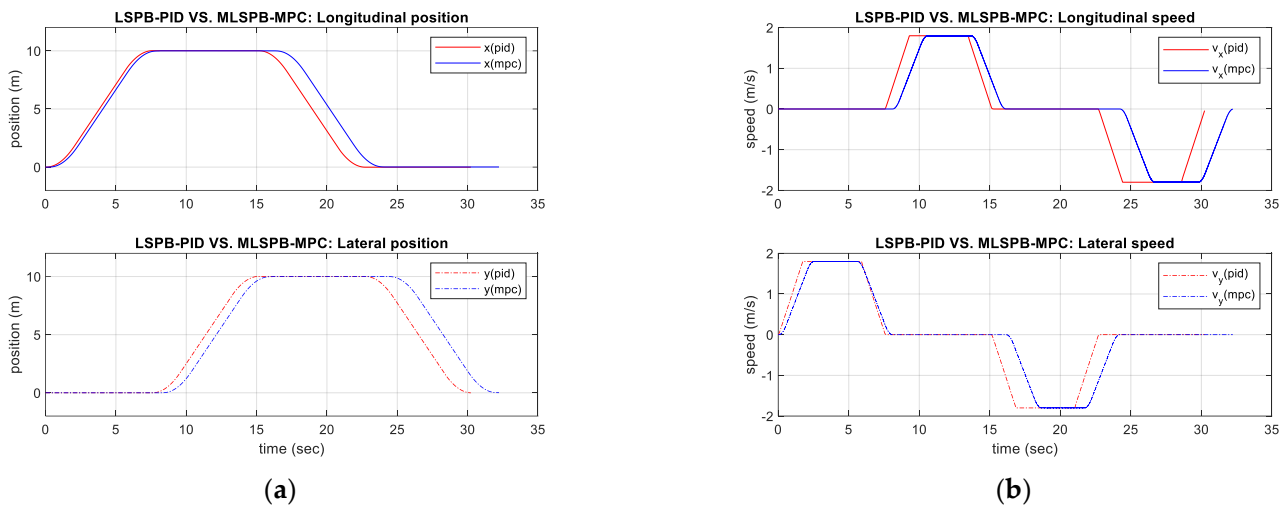
Figure 12. Robot navigation performance comparison between LSPB−PID and MLSPB−MPC in a rectangular shape path. (**a**) Position of the robot. (**b**) Speed of the robot.

Table 4. Navigation performance comparison: rectangular path.

Method	Position RSME (m)	Working Time (s)	Average Jerk (m/s^3)
LSPB−PID	0.0751	30.23	30,424
MLSPB−MPC	0.0604	32.23	1.8115

6.2.3. Warehouse-like Environment: LSPB vs. MLSPB

In the final simulation, an unmanned forklift operation scenario was designed in an automated warehouse environment to apply the proposed algorithm. The use of Mecanum wheel-based forklifts in smart warehouses is due to the space utilization efficiency resulting from the omnidirectional movement of the MWR. At the beginning of this study, we visited the materials warehouse at Samsung Shipyard, which builds ships, and confirmed that the distance between racks storing materials was a little over 2 m. The warehouse environment applied to the simulation was designed as shown in Figure 13, similar to the Samsung Heavy Industries material warehouse we visited. In other words, the effectiveness of the proposed algorithm was confirmed in an environment similar to an actual work site. Black boxes represent shelves in the warehouse. The red 'x' marks represent the waypoint that the robot must pass through, and the blue line represents the LSPB−based robot−driving trajectory. Figure 13 shows the path for moving to a given waypoint in an automated warehouse environment and the LSPB−PID and MLSPB−MPC-based trajectory planned accordingly. Figure 14a shows the waypoint of the warehouse the robot must pass through and Figure 14b shows longitudinal and lateral trajectory of the robot, velocity and acceleration of the robot.
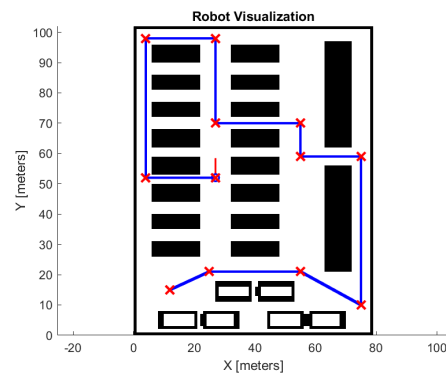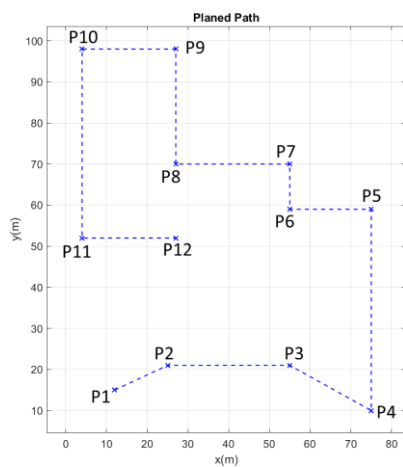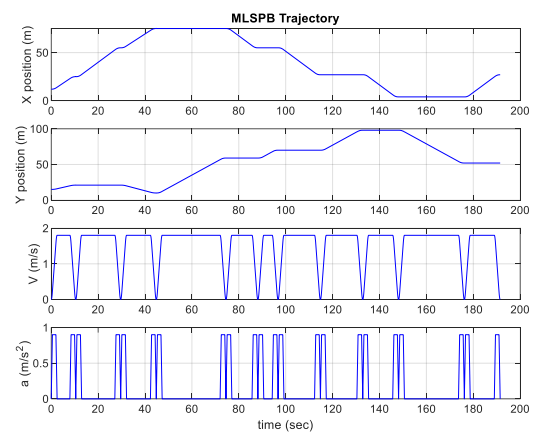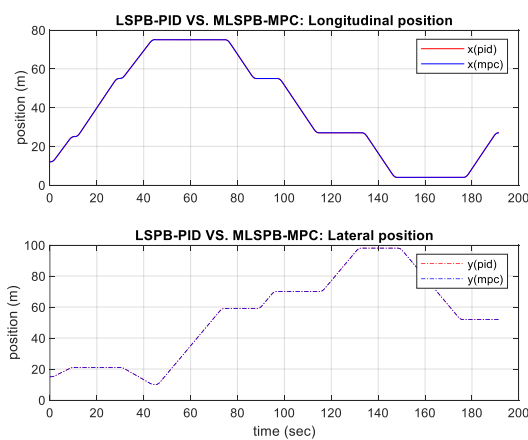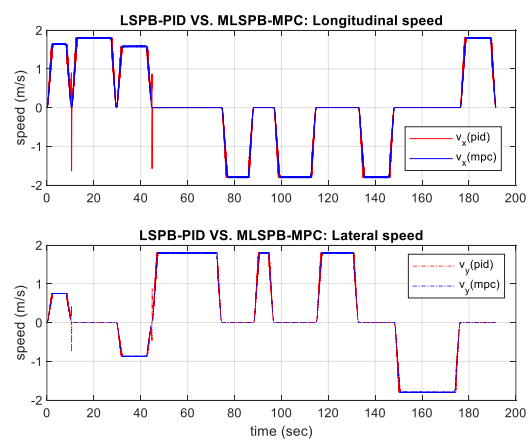
Figure 13. Warehouse environment.

Figure 14. Robot navigation through MPC control in the warehouse−like environment. (**a**) Waypoints of the warehouse. (**b**) Robot longitudinal velocity and lateral velocity.

Figure 15 and Table 5 show the warehouse layout-based navigation performance. Both methods successfully accomplished the task. However, as in the previous simulation, it was confirmed that the M-LSPB-MPC-based method has low RMS error and average jerk value.

Figure 15. Robot navigation through MPC control in the warehouse-like environment. (**a**) Robot visualization. (**b**) Robot longitudinal velocity and lateral velocity.

Table 5. Navigation performance comparison: warehouse path.

Method	Position RSME (m)	Working Time (s)	Average Jerk (m/s^3)
LSPB−PID	0.0795	191.468	14,097
MLSPB−MPC	0.0705	191.520	3.7184

7. Conclusions

Currently, many robot platforms are widely used in industry. This research focuses on developing a mobile robot platform for automated warehouses to save time and reduce costs. We adapted a modified−LSPB algorithm to harness the advantages of the Mecanum wheel and conducted simulations in MATLAB. The Mecanum wheel robot's distinct advantage lies in its capacity for omnidirectional movement without necessitating wheel rotation. This unique capability has been incorporated into the LSPB algorithm. The algorithm's efficacy has been demonstrated through simulations, wherein the robot was tasked with three scenarios: straight-line trajectory, curved trajectories, and a warehouse-like environment. The MLSPB path−planning algorithm meticulously designs the robot's trajectory, adhering to the specified velocity and acceleration parameters, ensuring accurate tracking of predefined waypoints. Furthermore, the robot's movement was controlled using the Model Predictive Control (MPC) method. Simulation results indicate that the robot's movement aligns closely with the M-LSPB path, with minimal error observed across all scenarios. However, it is imperative to note the limitations of this study, primarily its reliance on simulations. To develop a system that can be applied in real-world environments, it is essential to conduct experiments on actual robots to achieve greater stability and efficiency. Future tasks lie in the application of the proposed algorithm to actual unmanned forklifts. Project planning for actual vehicle-based control is in progress, and efforts will be made to secure reliable technology through production of actual vehicles and testing in real environments in the near future.

Author Contributions: Conceptualization, M.P. and C.K.; methodology, K.V. and C.K.; software, K.V. and C.K.; validation, K.V. and C.K.; formal analysis, K.V. and M.P.; writing—original draft preparation, K.V. and C.K.; writing—review and editing, M.P. and C.K.; visualization, K.V.; supervision, M.P. and C.K. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The data is unavailable due to privacy.

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

References

1. Bourne, D.; Choset, H.; Hu, H.; Kantor, G.; Niessl, C.; Rubinstein, Z.; Simmons, R.; Smith, S. Mobile manufacturing of large structures. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, DC, USA, 26–30 May 2015; p. 1565.
2. Shneier, M.; Bostelman, R. *Literature Review of Mobile Robots for Manufacturing*; NIST: Gaithersburg, MD, USA, 2015. [CrossRef]
3. Polaris. Autonomous Forklift Market Share, Size, Trends, Industry Analysis Report. Available online: https://polarismarketresearchscoop.wordpress.com/2022/07/18/autonomous-forklift-market-business-status-growth-industry-trends-and-outlook-2022-to-2030/ (accessed on 24 October 2023).
4. Qian, J.; Zi, B.; Wang, D.; Ma, Y.; Zhang, D. The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System. *Sensors* **2017**, *17*, 2073. [CrossRef] [PubMed]
5. Taher, H.; Zhao, C.X. Omnidirectional mobile robots, mechanisms and navigation approaches. *Mech. Mach. Theory* **2020**, *153*, 103958. [CrossRef]
6. Kanjanawanishkul, K. Omnidirectional wheeled mobile robots: Wheel types and practical applications. *Int. J. Adv. Mechatron. Syst.* **2015**, *6*, 74788. [CrossRef]
7. Adăscăliţei, F.; Doroftei, I. Practical applications for mobile robots based on Mecanum wheels—A systematic survey. *Rom. Rev. Precis. Mech. Opt. Mechatron.* **2011**, *40*, 21–29.

8.	Kumar, N.V.; Kumar, C.S. Development of collision free path planning algorithm for warehouse mobile robot. *Procedia Comput. Sci.* **2018**, *133*, 456–463. [CrossRef]

9.	Massoud, M.M.; Abdellatif, A.; Atia, M.R.A. Different Path Planning Techniques for an Indoor Omni-Wheeled Mobile Robot: Experimental Implementation, Comparison and Optimization. *Appl. Sci.* **2022**, *12*, 12951. [CrossRef]

10.	Karur, K.; Sharma, N.; Dharmatti, C.; Siegel, J.E. A Survey of Path Planning Algorithms for Mobile Robots. *Vehicles* **2021**, *3*, 448–468. [CrossRef]

11.	Kim, J.-Y.; Kim, D.-H.; Kim, S.-R. On-line minimum-time trajectory planning for industrial manipulators. In Proceedings of the 2007 International Conference on Control, Automation and Systems, Seoul, Republic of Korea, 17–20 October 2007; pp. 36–40.

12.	Kim, J.; Kim, S.R.; Kim, S.J.; Kim, D.H. A practical approach for minimum-time trajectory planning for industrial robots. *Ind. Robot Int. J.* **2010**, *37*, 51–61. [CrossRef]

13.	Davies, T.; Jnifene, A. Multiple Waypoint Path Planning for a Mobile Robot using Genetic Algorithms. In Proceedings of the CIMSA 2006 IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, La Coruna, Spain, 12–14 July 2006; pp. 21–26.

14.	Samad, T.; Bauer, M.; Bortoffc, S.; Cairano, S.D.; Fagiano, L.; Odgaard, P.F.; Rhinehart, R.R.; Sánchez-Peña, R.; Serbezov, A.; Ankersen, F.; et al. Industry engagement with control research: Perspective and messages. *Annu. Rev. Control* **2020**, *49*, 1–14. [CrossRef]

15.	Qina, S.J.; Badgwell, T.A. A survey of industrial model predictive control technology. *Control Eng. Pract.* **2003**, *11*, 733–764. [CrossRef]

16.	Afram, A.; Janabi-Sharifi, F. Theory and applications of HVAC control systems—A review of model predictive control (MPC). *Build. Environ.* **2014**, *72*, 343–355. [CrossRef]

17.	Umar, A.A.; Kim, J.-S. Nonlinear Model Predictive Path-Following for Mecanum-Wheeled Omnidirectional Mobile Robot. *Trans. Korean Inst. Electr. Eng.* **2021**, *70*, 1946–1952. [CrossRef]

18.	Moreno-Caireta, I.; Celaya, E.; Ros, L. Model Predictive Control for a Mecanum-wheeled Robot Navigating among Obstacles. *IFAC Pap.* **2021**, *54*, 119–125. [CrossRef]

19.	Tamba, T.A.; Hong, B.; Hong, K.-S. A path following control of an unmanned autonomous forklift. *Int. J. Control Autom. Syst.* **2009**, *7*, 113–122. [CrossRef]

20.	Affia, I.; Aamer, A. An internet of things-based smart warehouse infrastructure: Design and application. *J. Sci. Technol. Policy Manag.* **2021**, *13*, 90–109. [CrossRef]

21.	Hao, J.; Shi, H.; Shi, V.; Yang, C. Adoption of automatic warehousing systems in logistics firms: A technology–organization–environment framework. *Sustainability* **2020**, *12*, 5185. [CrossRef]

22.	Macfarlane, S.; Croft, E.A. Jerk-bounded manipulator trajectory planning: Design for real-time applications. *IEEE Trans. Robot. Autom.* **2003**, *19*, 42–52. [CrossRef]

23.	Walch, A.; Eitzinger, C.; Zambal, S.; Palfinger, W. LSPB Trajectory Planning Using Quadratic Splines. In Proceedings of the 3rd International Conference on Mechatronics and Robotics Engineering, Paris, France, 8–13 February 2017; pp. 81–87. [CrossRef]

24.	Nurallah, G.; Bing, Q.; Hamid, T. Kinematic Model of a Four Mecanum Wheeled Mobile Robot. *Int. J. Comput. Appl.* **2015**, *113*, 6–9. [CrossRef]

25.	Li, Y.; Dai, S.; Zheng, Y.; Tian, F.; Yan, X. Modeling and Kinematics Simulation of a Mecanum Wheel Platform in RecurDyn. *J. Robot.* **2018**, *2018*, 9373580. [CrossRef]

26.	Jia, Y.; Song, X.; Xu, S.S.-D. Modeling and Motion Analysis of Four-Mecanum Wheel Omni-directional Mobile Platform. In Proceedings of the International Automatic Control Conference (CACS), Sun Moon Lake, Taiwan, 2–4 December 2013.

27.	Al-Dahhan MR, H.; Schmidt, K.W. Safe and Efficient Path Planning for Omni-directional Robots using an Inflated Voronoi Boundary. *Ç Ankaya Univ. J. Sci. Eng.* **2019**, *16*, 46–69.

28.	Kim, J.-C.; Pae, D.-S.; Lim, M.-T. Obstacle Avoidance Path Planning based on Output Constrained Model Predictive Control. *Int. J. Control Autom. Syst.* **2019**, *17*, 2850–2861. [CrossRef]

29.	Klančar, G.; Zdešar, A.; Blazic, S.; Škrjanc, I. Path Planning. In *Wheeled Mobile Robotics*; Butterworth-Heinemann: Cambridge, MA, USA, 2017; pp. 161–206.

30.	Siegwart, R.; Nourbakhsh, I.R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2011; p. 369.

31.	Craig, J.J. *Introduction to Robotics Mechanics and Control*, 3rd ed.; Dworkin, A., Ed.; Pearson Education International: Upper Saddle River, NJ, USA, 2005; p. 385.

32.	Khatamianfar, A. Advanced Discrete-Time Control Methods for Industrial Applications. Ph.D. Thesis, University of New South Wales, Sydney, NSW, Australia, 2015.

33.	Abdul-Lateef, W.E.; Huayier, A.F.; Farhood, N.H. Design of planning Trajectory for the planar Robot Manipulator using Linear Segments method with Parabolic Blends (LSPB). *J. Mech. Eng. Res. Dev.* **2021**, *44*, 159–171.

34.	Li, X.; Tan, S.; Feng, X.; Rong, H. LSPB Trajectory Planning: Design for the Modular Robot Arm Applications. In Proceedings of the 2009 International Conference on Information Engineering and Computer Science, Wuhan, China, 19–20 December 2009; pp. 1–4.

35.	Lei, J.; Luo, Z.; Li, C. *Multi-Robot Systems—New Advances [Working Title]*; IntechOpen: London, UK, 2023. [CrossRef]

36. Ioana, I.; Blejan, M.; Blejan, R. S-curve motion profiles generator for hydraulic actuators. In Proceedings of the 2019 International Conference on Hydraulics and Pneumatics—HERVEX, Baile Govora, Romania, 13–15 November 2019; pp. 124–127.

37. OMRON. HD-1500 Series Autonomous Mobile Robot. Available online: https://automation.omron.com/en/us/products/family/hd (accessed on 24 September 2023).

38. Wang, L. *Model Predictive Control System Design and Implementation Using MATLAB*; Springer: Berlin/Heidelberg, Germany, 2009.

39. Fiedler, F.; Karg, B.; Lüken, L.; Brandner, D.; Heinlein, M.; Brabender, F.; Lucia, S. do-mpc: Towards FAIR nonlinear and robust model predictive control. *Control Eng. Pract.* **2023**, *140*, 105676. [CrossRef]

40. RoboMate. Omni-Directional Mobility Platform. Available online: http://www.vetexinc.com/vehicles/robomate.html (accessed on 14 November 2023).