IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 March 2023, accepted 6 May 2023, date of publication 16 May 2023, date of current version 25 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276858

RESEARCH ARTICLE

An Automatic Parking Algorithm Design Using Multi-Objective Particle Swarm Optimization

SAEEDE MOHAMMADI DANIALI[1], ALIREZA KHOSRAVI[1], POURIA SARHADI[2], AND FATEMEH TAVAKKOLI[1]

[1]Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol 47148-71167, Iran
[2]School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield, AL10 9AB Hertfordshire, U.K.

Corresponding author: Pouria Sarhadi (p.sarhadi@herts.ac.uk)

ABSTRACT In this article, an automatic vehicle parallel parking algorithm, consisting of path planning, controller design, and state estimation is developed. The path is planned using clothoid sequences and a straight line, which avoids stopping the car to reorient the wheels. The control inputs, including speed and steering angle, are a function of traveled distance. This method enables the car to park from different initial poses, achieving reduced parking time and the ability to park in one or two maneuvers, in smaller than standard places. An evolutionary optimization algorithm is used to calculate the best speed parameter according to the defined criteria. The proposed technique utilizes the Unscented Kalman Filter (UKF) to estimate the traveled distance, resulting in a smaller error compared to the conventional Extended Kalman Filter (EKF). The research aims to introduce an optimal automatic parking algorithm to improve the existing methods in terms of parking duration, the required space size for parking in the maximum of two maneuvers, and path continuity. Finally, the fidelity and improved performance of the proposed method are assessed in various probable conditions using the powerful Monte Carlo simulations.

INDEX TERMS Automatic parking, Kalman filter, Monte Carlo method, optimization, path planning.

I. INTRODUCTION

The use of advanced control methods has seen extensive growth in modern vehicles over the past decade. Among the existing control methods, automatic parking algorithms, known as park assist, have attracted considerable attention. Parallel parking, in particular, can be an arduous task due to blind spots and the challenge of reversing, especially for drivers with less experience.

One of the main goals of vehicle control is to assist the car driver in driving the car safer in a comfortable way, which can be achieved through automatic control of parking the vehicle. Depending on the form of a parking spot, there are generally three types of parking maneuvers: vertical, diagonal, and parallel, with the latter being the most difficult for drivers. Most automakers have recently implemented the automatic parallel parking system on their cars. In early solutions, some automakers, such as Toyota and Volvo, have employed semi-automatic systems that only control the steering wheel, leaving the driver to control the gas, brake, and gear themselves [1], [2]. Automated parking assistance systems have continued to evolve, with newer solutions now being offered, such as fully automatic parallel parking that eliminates the need for any driver involvement [3]. The efficiency of the automatic parking system depends on the path planning method, controller design, and state estimation. In this article, an innovative method for vehicle automatic parking is presented.

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

A. STATE OF THE ART

Two types of path planning methods, online and offline, are available for automatic parking, depending on the control paradigm used. In the online method, the path is planned while the car is moving and can detect obstacles to prevent collisions. Fuzzy control is an example of this method [4], [5], [6], [7]. In [4], linguistic commands are used for fuzzy control. In [5], fuzzy control is applied to select an appropriate path from a series of existing maneuvers according to the surrounding environment.

In [7], a genetic algorithm is used to fine-tune the fuzzy logic parking algorithm. Although these methods are suitable for dealing with uncertainty in the surrounding environment, drawbacks of fuzzy control include the need for expert knowledge and difficulty in generalization. Additionally, a dynamic environment can render a previously planned parking trajectory incorrect, requiring the parking planner to quickly and accurately replan. To address the challenges of speed and optimality in online path planning, the direct stitching method and the computational stitching method have been proposed in [8]. Other challenges include the need for highly accurate sensors and a higher computational burden compared to offline path planning. In the offline method, the path is planned in advance, resulting in a lower computational burden. This method has been explored in [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], and [22]. In [9] and [10], a path without collision is initially designed, which is then converted into an acceptable path and followed by considering the limits of motion. However, this method has a high error rate as a disadvantage, which may cause inaccurate tracking. To solve this problem, an iterative backward and forward motion can be used to guide the car to the desired point [11], [12], although this approach prolongs the time of the parking maneuver. Research in [13] has proposed parallel and forward vertical parking paradigms. The number of maneuvers for parallel parking is selected under vertical parking to create a target line set for each number of maneuvers.

In [14], a retrieving method is presented in which the path is first designed virtually for the car to pull out of the parking spot. Then, the final maneuver is obtained by following the path in the opposite direction. This algorithm eliminates repetitive maneuvers, reducing the error in the offline path planning system and decreasing the parking duration. In this method, two circular arcs are exerted to plan the path for parallel parking with the capability to park in small places. A path consisting of a straight line along with two circular arcs was employed in [15], where the car was able to park from different initial poses. These paths ([14], [15]) are segmented and the car stops at the junction of two path segments to reorient the front wheels, which causes the tires to wear, the tread to be lost, and puts pressure on the steering column, thus increasing the parking duration. Also, due to wheel rotation in the stop mode, the car may slip, which can cause an error in tracking. The polynomial curve was used in [16] and [17] to resolve this problem, which makes a rather smooth path. However, this method cannot be generalized for different initial poses.

In [18] and [19], the path is planned by converting the circular arcs to clothoid sequences to prevent the vehicle from stopping to reorient its front wheels at junction points. The Audi Company has used this method for path planning [23]. It is shown that the path consisting of the clothoid sequence can also be generalized for the various initial poses, in the vehicle's automatic parking [20]. The problem with this method, however, is the need for a larger parking spot. In [21] time-optimal parallel parking is presented which utilizes an Interior-Point Method (IPM)-based simultaneous dynamic optimization methodology to solve the optimization problem numerically. Despite the accuracy of this method, the computation time is not short enough to be used in practical systems. To solve this problem, a real-time near-optimal approach based on IPM is applied [22]. Another issue that needs to be addressed is collision avoidance, which has been tackled using the Interior Point Method (IPM) in the presence of irregularly placed obstacles. First, a time-optimal dynamic optimization problem is formulated, followed by the introduction of an IPM-based simultaneous approach to solve the problem [24]. Other studies, such as [25], have employed the Gauss Pseudo-spectral Method (GPM) and Model Predictive Control in the planning layer and tracking layer, respectively, for autonomous parallel parking problems in narrow spaces. This demonstrates that the tracking performance of this method is superior to traditional solutions like PID, despite the fact that the exact number of maneuvers has not been considered. Reinforcement learning has been applied for motion planning in various fields, as seen in [26], which involves data generation, data evaluation, and training of the network using selected data. In [27], other researchers have studied a deep neural network-based control structure for the automatic parking maneuver process, which was designed and implemented.

Some scholars have based their studies on the DDPG algorithm to study the automatic parking strategy, which has higher requirements for the car's path tracking sensor [28]. To reach good actions and reduce the requirements for automotive sensors, an automatic parking model has been proposed in [29], which is based on the car's parking kinematics model and deep reinforcement learning. Traffic congestion and mismanagement are other issues for which data-driven solutions have been proposed to tackle them. These data-driven solutions utilize the already available infrastructure of surveillance cameras installed at parking lots, thus overcoming the limitations of sensor-based solutions. In [30], an intelligent parking management system has been developed, which employs deep learning to address the limitations of data-driven solutions by leveraging the high performance and fast inference capability of YOLO v5 for vehicle detection.

The motivation of this study is to propose an optimal automatic parallel parking algorithm that improves the existing methods in terms of parking duration, space requirement (a maximum of two maneuvers), path continuity, and measurement error reduction.

B. THE CONTRIBUTION OF THIS PAPER

In this paper, a novel method for automatic parallel parking of vehicles, including path planning, controller design, and estimation of traveled distance is proposed. For this purpose, a path is initially planned considering the system's motion limitations. The control inputs are then adjusted to follow the

designed path. The kinematic and motion constraints of the car are initially examined to generate a path that a car-like vehicle can move on it. A retrieving method is used for path planning; first, a path is designed to get the car out of the parking spot, then followed in the opposite direction to get the car into the parking spot. The path consists of clothoid sequences and a straight line, forming a continuous path with no stops for wheel rotation, enabling the car to park from different initial poses. In this problem, the control inputs, including the vehicle speed and steering angle, are only a function of the traveled distance. The contributions of this paper are in three subjects:

- A new method is proposed for the speed design so that the car can park with one or two maneuvers in smaller spots in addition to reducing the parking duration. An evolutionary optimization method is proposed accordingly. Given the need to minimize both the time and the length of the parking spot, the problem is a multi-objective one. The use of the Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is proposed for multi-objective optimization.
- The Unscented Kalman Filter (UKF) is suggested to estimate the traveled distance, which has a lower error than the conventional Extended Kalman Filter (EKF).
- Performance of the proposed parking system is then examined using the powerful Monte Carlo simulation for different conceivable states compared to other existing methods, and about 40% reduction of the parking duration and 25 centimeters in the required length of the parking spots are revealed by these simulations. The efficiency of UKF and EKF estimators are compared by the Monte Carlo method, supporting the conclusion that the Mean Squared Error (MSE) of the UKF estimator is at least 3 times less than the EKF estimator. Several simulations indicated the high fidelity of the proposed algorithm.

In Section II, a general explanation of the problem is presented, followed by a description of our proposed solution. In Section III, the control inputs of speed and steering angle are designed. The MOPSO algorithm is used to select the speed parameter in Section IV. The estimator is designed in Section V to estimate the traveled distance in each step. Section VI presents the simulation results, and Section VII concludes the paper. The parameters of the path are calculated in the Appendix.

II. PROBLEM STATEMENT

In this section, the stages of automatic parallel parking and the overall block diagram of this system are initially explained, followed by a description of the proposed solution to this problem. In a typical automatic parking process, first, a suitable place should be found, and the initial pose of the vehicle is calculated. Then, a proper path is designed and finally, control inputs are to be set to follow the path. Therefore, automatic parking generally includes three steps:

- Perception: In this step, the dimensions of the parking spot and the pose of the vehicle are recognized. The ultrasound sensor, light sensors, or cameras are usually used for this step [31].
- Path Planning: In this step, a suitable path for parking is planned according to the parking spot dimensions, vehicle pose, and control constraints.
- Execution of the Maneuver: At this stage, the planned path of the previous step is followed by adjusting the control inputs.

A. THE OVERALL AUTOMATIC PARKING SYSTEM

In this subsection, the general system of vehicle automatic parking is explained. Further, the kinematic model of the vehicle and its motion constraints are investigated. In this research, the car parking problem is considered in a parallel mode, and it is assumed that the dimensions of the parking spot and the initial pose of the vehicle have been determined. First, a path is planned offline to design the automatic parking algorithm. Then, the control inputs are generated to follow the path. The control inputs (ν, δ), the speed and steering angle, are given to the system in each step to control the output of the system (x, y, θ). Using the wheel speed sensors, the vehicle displacement and its angle variations (Δ, ω) are calculated in each step or can be calculated using the output of the system. The estimator then determines the traveled distance (d) using the displacement and angle variations. The control inputs are also produced according to the traveled distance estimation. The overall model of the car's automatic parking system is shown in Fig. 1. According to the general model of the designed system, the vehicle model and its motion constraints must first be evaluated to design an appropriate path and control the vehicle properly. Continuing the process, the kinematic model of the car and its motion limitations are examined. Fig. 2 displays the modeling of the vehicle on the coordinate plane. According to this figure, (x,y) is the position of the center of the rear axle, θ is the vehicle orientation, δ is the angle of the virtual wheel in the middle of the front axle, δ_r and δ_l respectively indicate the angle of the right and left front wheels of the vehicle, a is the wheelbase and 2b denotes the vehicle wheels width. The speed ν is an important indicator which MPSO is proposed for optimization in this paper. The problem is multi-objective, and MPSO has been applied to deal with multi-objective optimization problems which will be more explained in Section IV. The kinematic model of the car-like vehicle is as follows (a discrete-time version of these equations is used in the simulation results section):

$$\begin{cases} \dot{x} = \nu \cos\theta \\ \dot{y} = \nu \sin\theta \\ \dot{\theta} = \left(\frac{\nu}{a}\right) \tan\delta \end{cases} \tag{1}$$

The first two equations are for the geometrical motion of the vehicle and the third one denotes the relationship between the steering angle and vehicle yaw rate. This problem has some motion limitations as follows:

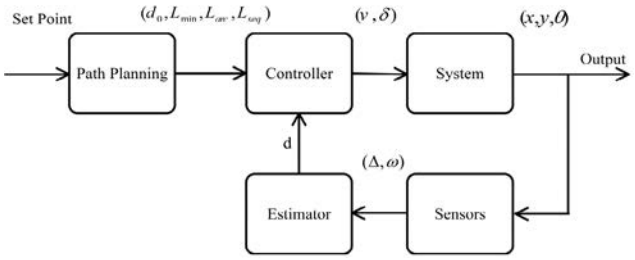

FIGURE 1. General block diagram of the vehicle automatic parking.

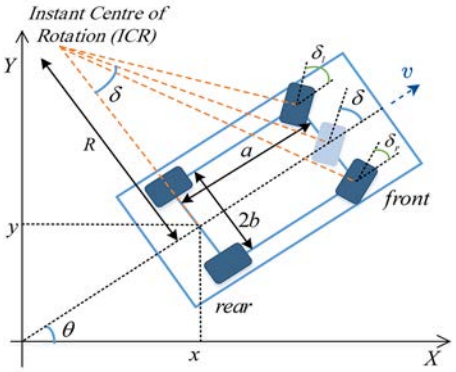

FIGURE 2. Vehicle modeling at the coordinate plane.

- Nonholonomic constraint: The system is nonholonomic as the control inputs are fewer than configuration variables [32]. The nonholonomic feature restricts movement, making it difficult to control the car. Thus, the car will not be able to move on every path.
- Wheel angle limitation: The wheel's angle has a maximum value of $\delta_{\max}$. According to Fig. 2, the minimum radius of rotation in the middle of the rear axle is as follows:

$$R_{\min} = a/\delta_{\max} \tag{2}$$

This problem also has another wheel angle limitation that requires a certain time to rotate the wheels from zero to $\delta_{\max}$. This time is obtained from the equation $t_{\min} = \delta_{\max}/v_\delta$, where v_δ is the maximum steering velocity of the steering wheel. For the intended vehicle, this is set to $v_\delta = 20^o/s$. If this rotation occurs while the car is moving, the minimum length for the rotation of the wheels from zero to $\delta_{\max}$ is calculated from the equation $L_{\min} = v_{longi} t_{\min}$, where v_{longi} is the maximum designed linear velocity. In [21] and [22], this limitation is considered negligible, which is not acceptable in practical scenarios.

- Speed limitation: The vehicle kinematic model includes geometric equations of the system which is suitable for low-speed applications (for example, speeds less than 5 m/s). The dynamics of the system is used for higher speeds. Therefore, the speed function design for a car parking system has limitations such as not exceeding a certain threshold [33].

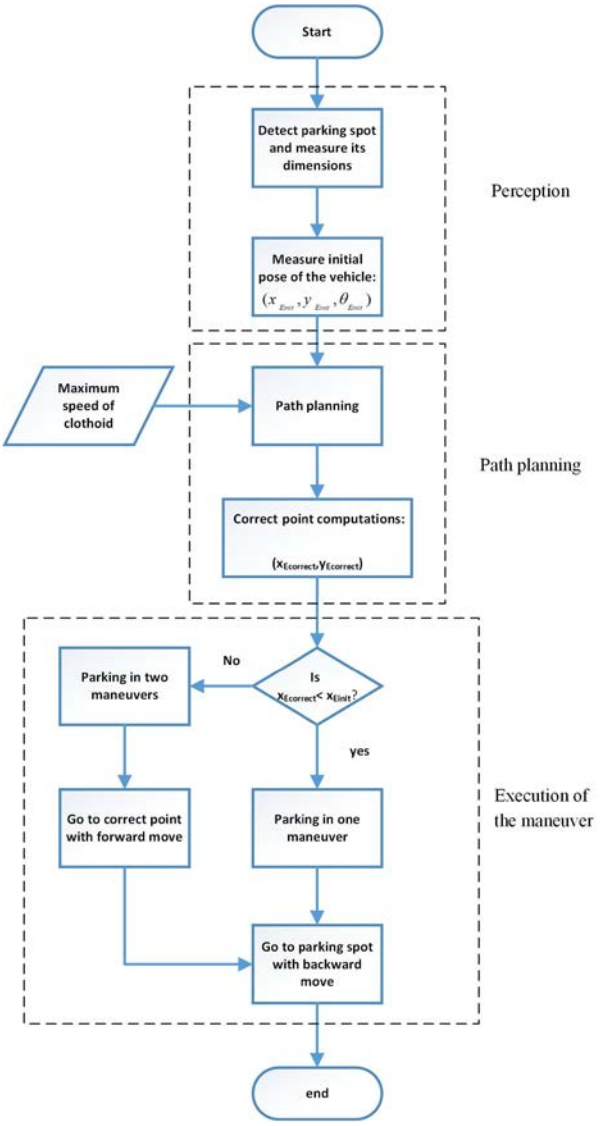

FIGURE 3. Flowchart for the implementation of the proposed method.

Motion constraints should be considered for path planning to enable the car to move along the designed path.

B. THE PROPOSED SOLUTION

The flowchart of the proposed method is demonstrated in Fig. 3. First, a suitable place is found to perform the parking maneuver. Then, the dimensions of the parking spot and the initial pose of the vehicle are measured. In this research, it is assumed that the dimensions of the parking spot and the initial pose of the vehicle are determined. The path is initially planned recursively, for a state where the car is at the parking spot, intending to leave the place (the center of the rear axle of the car is considered as the origin of the coordinates in this case). Then, the designed path is followed in the reverse direction. In this method, the motion constraints are considered. The path consisting of two circular arcs is a routine path for parallel parking. One disadvantage of this

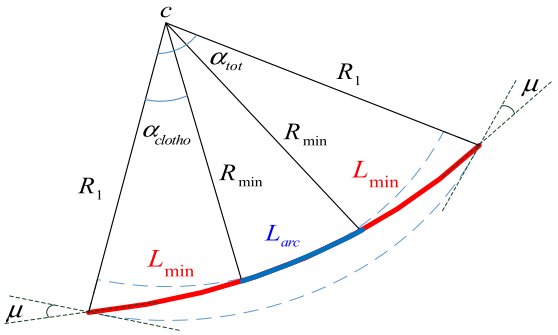

FIGURE 4. Comparison of the circular arc with the combination of two clothoid curves (Lmin) and a circular arc (Larc).

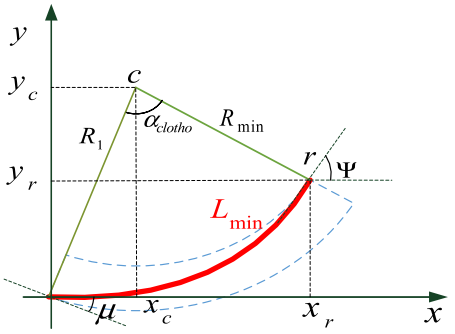

FIGURE 5. Clothoid curve.

method is that it causes the wheels to halt rotation at junction points, which brings about wheel wear and puts pressure on the steering column, besides increasing the parking duration. Another suggested path is consisting of clothoid sequences along with a straight line. This is a continuous-curvature path and the parking maneuver can be done from different initial poses. Each clothoid sequence is included of one clothoid curve and two circular arcs (see Fig. 4). The clothoid is a curve in which the curvature has a linear relationship with the curve length. Fig. 5 shows a clothoid curve. The length and curvature of the path are dependent on the maximum clothoid speed. The parameters of the path are calculated in the appendix. The parking maneuver with two clothoid sequences and a straight line is depicted in Fig. 6. The maximum speed of the clothoid is an important parameter in the optimization of path planning. This speed is calculated offline just once for each car. Then, a path is planned for parallel parking according to the measurements and the maximum clothoid speed.

The correction point is then calculated. If this point is on the left side of the starting point, the parking operation can be accomplished with one maneuver and the car can be parked by only moving backward. If the correcting point is on the right side, the vehicle must first move forward to reach the correction point and then continues the maneuver by moving backward. Therefore, two maneuvers will be needed to park the car. The execution step is done by designing the

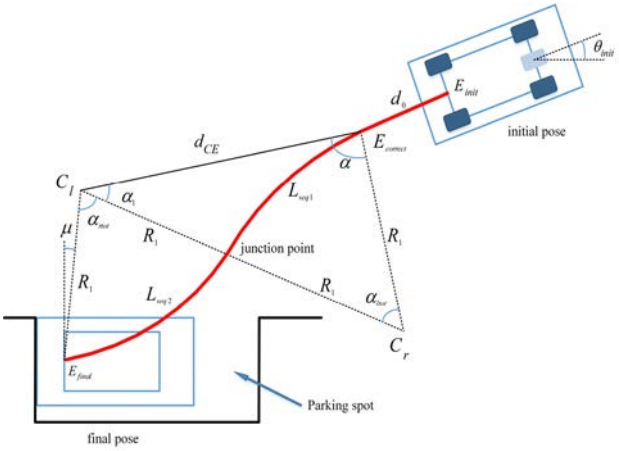

FIGURE 6. The parking path consisting of two clothoid sequences and a straight line.

controllers and the estimator. First, the control inputs are generated to follow the path. These controllers are produced according to the traveled distance. Then, the traveled distance is estimated using UKF. At the end of this procedure, the vehicle will be parked in the desired location.

In the following sections, three subjects are described: designing control inputs, using the MOPSO algorithm to calculate the optimal speed, and designing the estimator.

III. DESIGNING CONTROL INPUTS

In this section, the design of control inputs is presented, including the speed and steering angle as functions of the traveled distance. These controllers are designed to follow the planned path from the previous section. Since the length of the clothoid curve is dependent on speed, the length and curvature of the path are changed following any changes in the speed. Therefore, designing an appropriate speed is important. More details are provided below.

A. DESIGNING THE STEERING ANGLE

The steering angle is a function of the traveled distance and consists of two parts. According to Fig. 6, the movement initially occurs on a straight line with a length of d_0 and a steering angle of zero:

$$\forall d \in (0, d_0)\delta(d) = 0 \tag{3}$$

The parking maneuver is continued in two parts, each of which is a clothoid sequence. Three possible modes may occur for each of these curves [20]. In this case, the path must be designed to initially increase the steering angle and remain constant for a while after reaching its maximum and then decrease to zero. The required time to reach the maximum value of the steering angle depends on the clothoid component A. The time during which the steering angle stays at its maximum is also dependent on the length of the circular arc. Given the L_{arc} and L_{seq} obtained from the appendix, the

steering angle will be calculated from the following equation:

$$\delta(d) = \begin{cases} k_\delta \left|\arctan\left(a/R\right)\right| & d \in [0, L_{\min}] \\ k_\delta \left|\arctan\left(a/R_{\min}\right)\right| & d \in [L_{\min}, L_{\min} + L_{arc}] \\ k_\delta \left|\arctan\left(a/R_i\right)\right| & d \in [L_{\min} + L_{arc}, L_{seq}] \end{cases} \tag{4}$$

where k_δ is $+1$ and -1 for turning the wheels to the left and right, respectively. R_{min} is the minimum radius of the circle. R and R_i are also defined as $R = (A^2_{\min}/d)$ and $R_i = (A^2_{\min}/(L_{seq} - d))$.

B. DESIGNING THE SPEED FUNCTION

In this subsection, an innovative method is proposed to design the speed function which can reduce the parking duration as well as the length of the required space. To obtain the optimal controller, the advantages and disadvantages of path planning methods are examined. The path comprised of two circular arcs is the most optimal route in terms of the required space length for car parking. The problem is the rotation of the wheels in a stopped mode, which causes wear of the tires, loss of tread, and pressure on the steering column, thus increasing parking duration. In the method of path-planning with clothoid sequences, the wheels rotate while moving. Therefore, the problem of stopping mode is fixed, but this method increases the size of the required space for parking the car. The clothoid curve is the reason for the increased length since the length of the clothoid curve depends on the maximum linear velocity; as this speed increases, the length of the clothoid curve increases as well. Consequently, the curvature of the clothoid sequence decreases, requiring the car to have a larger space for parking. At very slow speeds, the length of the clothoid curve also reduces and approaches zero, and the path formed by the clothoid sequence is aligned almost on the planned path with the circular arc. Therefore, the designed path with the clothoid sequence is the optimal method at very low speeds in terms of the required space for car parking. If the length of clothoid curves can be reduced, so does the length of the required space for parking. The proposed method is to reduce the maximum speed for each of the clothoid curves ($V_{clothoid}$) and increase the maximum speed for the rest of the path ($V_{\max}$). This method of speed design reduces the required space for parking the car as well as the parking duration.

As stated, the correction point is initially calculated to do the parking maneuver. Then, the car moves from the starting point (x_{Einit}, y_{Einit}) to the correction point ($x_{Ecorrect}, y_{Ecorrect}$) without turning the steering wheel and continues the parking maneuver by following two clothoid sequences. If this point is to the left of the starting point, the parking operation can be done with one maneuver; otherwise, two maneuvers are required to park the car. The number of maneuvers here refers to the movement of changing the car's direction. In parking with one maneuver, the vehicle is only moving backward, but in parking with two maneuvers, the car first moves forward to reach the correction point. Then, the parking maneuver proceeds by moving backward. The proposed speed function for one maneuver is defined as follows:

$$v(d) = \begin{cases} \max(-V_{\max}, -\sqrt{\lambda}) & d \in [0, d_0 - L_d] \\ -\sqrt{-\lambda + V^2_{\max}} & d \in [d_0 - L_d, d_0] \\ \min(-V_{clothoid}, -\sqrt{\lambda}) & d \in [d_0, d_0 + L_{\min}] \\ \max(-V_{\max}, -\sqrt{\lambda + V^2_{clothoid}}) & d \in [d_0 + L_{\min}, d_1] \\ \min(-V_{clothoid}, -\sqrt{-\lambda + V^2_{\max}}) & d \in [d_1, d_2] \\ \max(-V_{\max}, -\sqrt{\lambda + V^2_{clothoid}}) & d \in [d_2, d_3] \\ \min(-V_{clothoid}, -\sqrt{-\lambda + V^2_{\max}}) & d \in [d_3, L_{total} - L_p] \\ -\sqrt{-\lambda + V^2_{clothoid}} & d \in [L_{total} - L_p, L_{total}] \end{cases} \tag{5}$$

where

$$\lambda = 2|a_c|\Delta d$$

$$\begin{cases} d_1 = d_0 + L_{\min} + L_{arc1}/2 \\ d_2 = d_0 + L_{seq1} + L_{\min} \\ d_3 = L_{total} - L_{\min} - L_{arc2}/2 \end{cases},$$

$$\begin{cases} L_{seq1} = 2L_{\min} + L_{arc1} \\ L_{seq2} = 2L_{\min} + L_{arc2} \\ L_{total} = d_0 + L_{seq1} + L_{seq2} \end{cases}$$

$$L_m = \frac{V^2_{\max}}{2\,|a_c|}, L_p = \frac{V^2_{clothoid}}{2\,|a_c|}, L_d = \frac{V^2_{\max} - V^2_{clothoid}}{2\,|a_c|} \tag{6}$$

In the above equations, a_c is acceleration. Only the first two phrases of (5) are changed in designing the speed of car parking with two maneuvers. This speed function can be defined for $d < d_0$ as follows:

$$v(d) = \begin{cases} \min(V_{\max}, \sqrt{2|a_c|\Delta d}) & d \in [0, d_0 - L_m] \\ \sqrt{-2|a_c|\Delta d + V^2_{\max}} & d \in [d_0 - L_m, d_0] \end{cases} \tag{7}$$

The $V_{\max}$ is considered as large as possible to design the optimal speed function. The $V_{clothoid}$ should be selected as such to minimize both the time and required space for parking the car. Since the parking duration and the required space for parking are correlated inversely, the $V_{clothoid}$ must be selected to optimize both of them as much as possible. The proposed method for this purpose is presented in the next section.

IV. MOPSO ALGORITHM FOR OPTIMAL SPEED CALCULATION

This section describes how to optimally calculate $V_{clothoid}$. This velocity is calculated offline just once for each car. There are two criteria to consider: time and required space for parking the car. The $V_{clothoid}$ must be selected to minimize these two criteria as much as possible. Given the reverse relationship between these two criteria, the problem is multi-objective. The MOPSO algorithm has been suggested for this

problem, as it has the advantage of an intelligent selection of a set of random samples that speeds up the algorithm. Finally, a set of the best answers will be obtained using the algorithm. In this section, the MOPSO algorithm and the way it can be used for optimal calculation of the $V_{clothoid}$ are explained.

The MOPSO method is used for multi-objective optimization provided by Coello et al. in 2004 [34]. In this method, the particle set is randomly selected, and the problem is solved for each of these values. Then, the non-dominated answers are stored while the dominated ones are deleted. Another set of particles is selected intelligently, and the problem is solved with these values. Again, the non-dominated answers are stored, and the dominated answers are eliminated. This algorithm is repeated until the set of answers can be plotted as a proper graph. If the variables are chosen correctly, the Pareto Font set, a set of the most optimal answers, will be obtained.

The MOPSO algorithm is as follows in general:

1) Initiating the initial population of particles
2) Examining the particles and separating the non-dominated members of the population and storing them in the repository
3) Tabulating the discovered target space
4) Selecting a guide from the repository members ($REP[i]$) and updating the speed ($VEL[i]$) and the position ($POP[i]$) using the following equations:

$$\begin{aligned} VEL[i] &= W \times VEL[i] \times R_1 \times (PBEST[i] - POP[i]) \\ &\quad + R_2 \times (REP[i] - POP[i]) \\ POP[i] &= POP[i] + VEL[i] \end{aligned} \tag{8}$$

   where the $PBESTS[i]$ is the best personal memory of each particle and W is the inertial weight, while R_1 and R_2 are random numbers in the interval [0,1].
5) Updating the best personal memory of each particle
6) Adding non-dominated members of the current population to the repository
7) Eliminating of dominated members from the repository
8) If the number of repository members is more than the specified capacity, additional members are deleted, and the table is rearranged.
9) If the termination condition is fulfilled, go to Step 3; otherwise, the algorithm will be ended.

The method presented in [35] is used to optimally select the coefficients of this algorithm. Once these coefficients are obtained, they do not need to be recalculated. These coefficients are defined as follows:

$$\begin{cases} W = \chi \\ R_1 = \chi\phi_1 \\ R_2 = \chi\phi_2, \end{cases} \quad \begin{cases} \chi = \dfrac{2k}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \\ \phi = \phi_1 + \phi_2 \geq 4, \end{cases} \quad 0 \leq k \leq 1 \tag{9}$$

A proper choice for coefficients can be as follows: $k = 1$, $\phi_1 = 2.05$, and $\phi_2 = 2.05$.

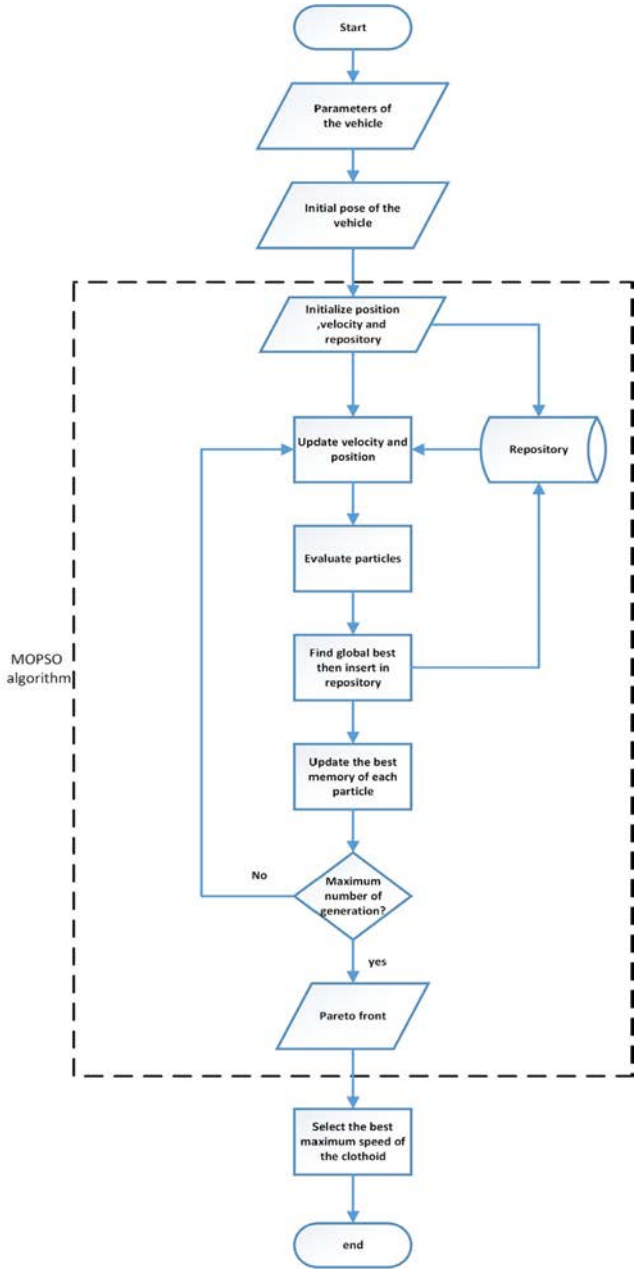

FIGURE 7. Flowchart of the proposed method for choosing the maximum.

The flowchart of the proposed method for calculating the maximum optimal clothoid velocity ($V_{clothoid}$) is presented in Fig. 7. In this problem, the $V_{clothoid}$ should be designed to optimize the car automatic parking algorithm, which should be done in the smallest possible place and in the shortest duration. Considering the multi-objective nature of the problem, the use of the MOPSO algorithm was suggested as a fast algorithm that achieves a set of optimal answers (Pareto Front). From the set of answers, an appropriate answer is chosen according to the optimization criteria. More details are given in the simulation section.

V. ESTIMATOR DESIGN

The estimated traveled distance at each step is used to control both the speed and steering angle of the car. An accurate estimator should be used to properly control the vehicle. To this end, the Unscented Kalman Filter (UKF) is used for the first time. In this section, the advantages of the UKF estimator are discussed first, followed by an explanation of this filter. The section also explains how to use the filter to estimate the traveled distance. In the simulation section, a comparison is made between the proposed filter and the standard Extended Kalman Filter (EKF) which points to the superiority of the filter used in this work.

The Kalman Filter is a simple, optimized, and robust filter used for estimating and tracking problems. The UKF filter, like the EKF, is presented to solve nonlinear problems. In the EKF filter, the distribution of states is approximated with the Gaussian Random Variable (GRV). Moreover, this approximation is linearized which causes a high error in the mean and covariance of the previous state. The UKF solves this problem by selecting a set of minimum sample points for the Gaussian random variable approximation. These sample points completely maintain the mean and the covariance of the Gaussian random variable. Therefore, it causes less error in the estimation of states. The UKF filter is easier to use than the EKF filter and provides a more accurate estimation of states [36], [37]. To continue the process, the UKF filter function is explained:

Suppose that the system equations and observation are defined nonlinear:

$$\begin{cases} x_{k+1} = F(x_k, \upsilon_k) \\ y_k = H(x_k, n_k) \end{cases} \tag{10}$$

$y = g\,(x)$, a nonlinear function of the variable x (with dimensions of L), is considered with a random distribution, where $\bar{x}$ and P_x are the mean and covariance of x, respectively. The matrix χ is introduced in Equation 9 to compute the statistical characteristic of y. This matrix has $2L + 1$ sigma vectors χ_i (with corresponding weights W_i).

$$\begin{aligned} \chi_0 &= \bar{x} \\ \chi_i &= \bar{x} + \left(\sqrt{(L+\lambda)P_x}\right)_i \; i = 1, \ldots, L \\ \chi_i &= \bar{x} - \left(\sqrt{(L+\lambda)P_x}\right)_{i-L} \; i = L+1, \ldots, 2L \\ W_0^{(m)} &= \lambda / (L+\lambda) \\ W_0^{(c)} &= \lambda / (L+\lambda) + (1 - \alpha^2 + \beta) \\ W_i^{(m)} &= W_i^{(c)} = 1 / \{2(L+\lambda)\} i = 1, \ldots, 2L \end{aligned} \tag{11}$$

where, $\left(\sqrt{(L+\lambda)P_x}\right)_i$ is the i$^{\text{th}}$ row of the root of the square of the matrix and λ is the scaling parameter, which is defined as follows:

$$\lambda = \alpha^2 (L + \kappa) - L \tag{12}$$

α is defined as the distribution of sigma points around $\bar{x}$, which is usually small and positive (e.g., 0.01). κ is the second scaling parameter and is usually considered as zero. β is used for the previous distribution composition of x (for Gaussian distribution, $\beta = 2$ is optimal). Sigma vectors are developed with the following nonlinear function:

$$\gamma_i = g(\chi_i) \quad i = 0, \ldots, 2L \tag{13}$$

The mean and covariance of matrix y are defined using the weighted samples mean and covariance of the previous sigma points as follows:

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^{(m)} \gamma_i, P_y \approx \sum_{i=0}^{2L} W_i^{(c)} [\gamma_i - \bar{y}][\gamma_i - \bar{y}]^{\mathrm{T}} \tag{14}$$

According to the presented explanations, the process of the UKF algorithm is described in the following:

- The initial conditions of the algorithm are defined as follows:

$$\begin{aligned} \hat{x}_0 &= E[x_0] \\ P_0 &= E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \\ \hat{x}_0^a &= E[x^a] = [\hat{x}_0^T \; 0 \; 0]^T \\ P_0^a &= E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_n \end{bmatrix} \end{aligned} \tag{15}$$

- For each $k \in \{1, \ldots, \infty\}$, the sigma points are calculated as follows:

$$\chi_{k-1}^a = \left[\hat{x}_{k-1}^a \; \hat{x}_{k-1}^a \pm \sqrt{(L+\lambda)P_{k-1}^a}\right] \tag{16}$$

- The time update is performed with the following equations:

$$\begin{aligned} \chi_{k|k-1}^x &= F[\chi_{k-1}^x, \chi_{k-1}^{\upsilon}] \\ \hat{x}_k^- &= \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1}^x \\ P_k^- &= \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k|k-1}^x - \hat{x}_k^-][\chi_{i,k|k-1}^x - \hat{x}_k^-]^{\mathrm{T}} \\ \gamma_{k|k-1} &= H[\chi_{k|k-1}^x, \chi_{k-1}^n] \\ \hat{y}_k^- &= \sum_{i=0}^{2L} W_i^{(m)} \gamma_{i,k|k-1} \end{aligned} \tag{17}$$

- Then come the measurement update equations:

$$\begin{aligned} P_{\tilde{y}_k \tilde{y}_k} &= \sum_{i=0}^{2L} W_i^{(c)} [\gamma_{i,k|k-1} - \hat{y}_k^-][\gamma_{i,k|k-1} - \hat{y}_k^-]^{\mathrm{T}} \\ P_{x_k y_k} &= \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k|k-1} - \hat{x}_k^-][\gamma_{i,k|k-1} - \hat{y}_k^-]^{\mathrm{T}} \\ \kappa &= P_{x_k y_k} P_{\tilde{y}_k \tilde{y}_k}^{-1} \\ \hat{x}_k &= \hat{x}_k^- + \kappa (y_k - \hat{y}_k^-) \\ P_k &= P_k^- - \kappa P_{\tilde{y}_k \tilde{y}_k}^{-1} \kappa^T \end{aligned} \tag{18}$$

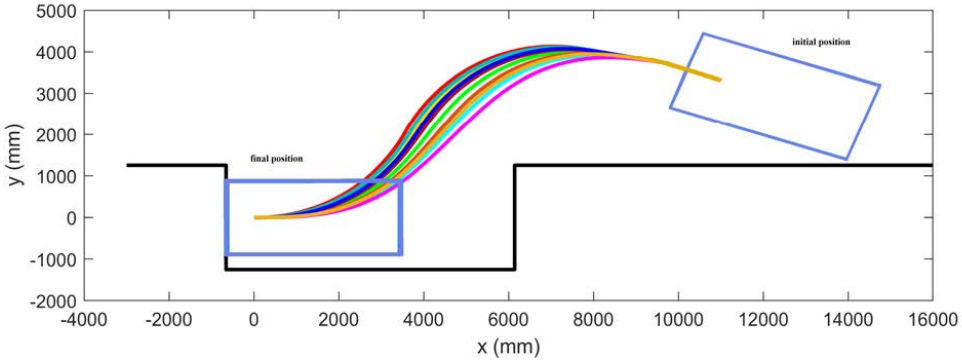

FIGURE 8. Parallel parking using different clothoid speeds.

In the equations above, $x^a = [x^T\ v^T\ n^T]^T$ and $\chi^a = [(\chi^x)^T\ (\chi^v)^T\ (\chi^n)^T]^T$, P_v and P_n represent the system covariance noise and the measurement noise, respectively.

In car automatic parking problems, the traveled distance (d) must be estimated from the beginning to the desired step. To this end, the traveled distance and the vehicle orientation change are initially estimated in each step, and the total traveled distance is then calculated accordingly. Therefore, the state vector will be as $x_k = \left[\ \Delta_k\ \omega_k\ \right]^T$. If Δ_{RR} and Δ_{RL} indicate the displacement of the right and left rear wheels of the vehicle, respectively, and Δ_{FR} and Δ_{FL}, respectively represent the displacement of the right and left front wheels of the vehicle, as shown in Fig. 2, δ_r, δ_l,and δ respectively indicate the angle of the right, left and the virtual front wheels of the vehicle; thus, the measurement equation is defined as follows [38]:

$$\begin{cases} \tan(\delta) = a\omega/\Delta \\ \Delta_{RL} = \Delta - b\omega \\ \Delta_{RR} = \Delta + b\omega \\ \Delta_{FL}\cos(\delta_l) = \Delta - b\omega \\ \Delta_{FR}\cos(\delta_r) = \Delta + b\omega \end{cases} \tag{19}$$

Therefore, the observation vector will be defined as $y = [\tan(\delta),\ \Delta_{RL},\ \Delta_{RR},\ \Delta_{FL}.\cos(\delta_L),\ \Delta_{FR}.\cos(\delta_R)]^T$, which is a nonlinear function of the state vector. Using the UKF algorithm, the state vector values $x = \left[\ \Delta\ \omega\ \right]^T$ are estimated in each step. Finally, the total traveled distance for each step is defined:

$$d_{k+1} = d_k + \Delta_k \tag{20}$$

Note: parking algorithms in tiny spots and parallel parking are not defined for obstacle avoidance (as in another similar research [20], [21] and even for a human driver). The assumption is that the parking spot is identified and clear. this does not mean that in the case of the presence of an obstacle there will be a collision. Since there is no space for collision avoidance maneuvers in parallel parking, in the case of obstacle detection, a new spot should be identified, and the parking maneuvers are repeated.

VI. SIMULATION RESULTS

This section presents the simulation results of the proposed method for the vehicle automatic parallel parking. The whole system was implemented in MATLAB and executed on a Laptop (Intel Core-i5-7200U CPU with 8GB RAM that runs at 2.5 GHz). The sample time was considered 0.01 seconds. Simulation of each parallel parking took less than 2 seconds. The maximum optimal clothoid velocity was initially calculated offline using the MOPSO algorithm, allowing the parking algorithm to be accomplished without any online implementation complexity. Then, various scenarios are presented to examine the performance of the proposed method. In the first scenario, the tracking ability of the proposed system is tested. In the next two scenarios, the car parking is evaluated with one or two maneuvers using the new method. In the fourth scenario, the Monte Carlo analysis is used to evaluate the performance of the system to start the parking from different poses. In the final scenario, the Monte Carlo analysis is eventually used to compare the UKF and EKF estimators.

An optimization algorithm is needed to select a proper $V_{clothoid}$ to simultaneously reduce both the time and the required space for parking the car. For this purpose, the MOPSO technique is proposed as described in Section IV (see Fig. 7). In [20], a clothoid sequence was used for path planning, and the maximum speed for the entire path was considered as 0.6. In this paper, the proposed method is to reduce the maximum speed for each of the clothoids ($V_{clothoid}$) and increase the maximum speed for the rest of the path ($V_{\max}$). Initially, the $V_{\max}$ is considered as large as possible. The value of 3 m/s is appropriate, and in this case, the car's dynamics can be ignored. The length and the curvature of the path are dependent on the $V_{clothoid}$. Fig. 8 shows the planned path for 10 different $V_{clothoid}$ values. The faster the $V_{clothoid}$, the smoother the path. Thus, a larger parking space would be needed for parking as well.

In the case of the MOPSO algorithm, the range of $V_{clothoid}$ is assumed to be between 0.1 and 0.6 m/s, and the number of replicates, particles, and repository capacity is considered to be 100, and 150, respectively. According to these values, the Pareto Front set (the set of optimal solutions) is displayed in Fig. 9. The closest answer to the optimization criterion is chosen from the optimal solutions. The points in the middle of the figure are suitable, which reduce both the duration and the required space for parking, simultaneously. From these points, the one that decreases the required space length by

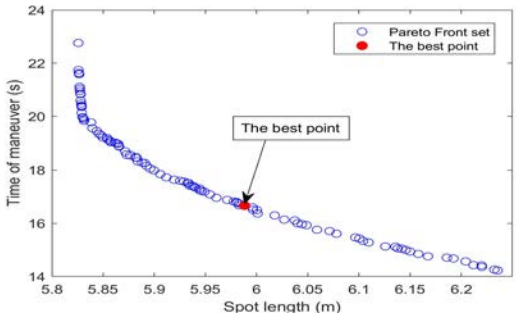

FIGURE 9. Pareto Front set along with the optimal point of choice.

TABLE 1. Parameters of renault ZOE [39].

Parameters	*Notation*	*Value*
Wheelbase	a	2588 mm
Track	$2b$	1511 mm
Front overhang	d_{front}	839 mm
Rear overhang	d_{rear}	657 mm
Distance from the left,right wheel to the left, right side of the vehicle (exterior mirror folded)	d_l, d_r	130 mm
Maximum left,right steering angle	$\delta_{l\max}$,$\delta_{r\max}$	33 °
The total width of the vehicle	$_{lvehicle}$	4084 mm
The total length of the vehicle	$w_{vehicle}$	1771 mm

25 centimeters is chosen. This point is indicated in Fig. 9 with a solid circle. The speed corresponding to this choice is 0.3 m/s. By choosing this speed, both the time and the required space length for parking the car will be optimized as much as possible. In the following scenarios, the maximum speed was 0.6 m/s for the whole path in the old method. In the new method, two values for maximum speeds are: $V_{clothoid} = 0.3$ and $V_{\max} = 3$m/s. It should be noted that the optimized speed is suitable for the considered vehicle, i.e., the Renault ZOE. Detailed specifications of this car are shown in Table 1. Suitable maximum clothoid speeds for other vehicles can be calculated easily by the proposed technique.

A. SCENARIO 1: THE TRACKING ABILITY OF THE DESIGNED CONTROLLER

In this scenario, a sample of an automatic parking maneuver using the proposed method is examined. In this case, the tracking ability of the proposed method is depicted in Fig. 10 in which the vehicle can follow the planned path quite well.

B. SCENARIO 2: PARKING THE CAR WITH ONE MANEUVER

In this case, the car can be parked only by moving backward. In Fig. 11, the parking with one maneuver on the XY plane is plotted for both methods. The proposed method in this paper is compared to a similar class parking approach proposed in [20]. It is tried to demonstrate the abilities of the proposed approach (new method) in tiny space parking. The approach proposed in [20] is called the old method. The initial pose is as follows: x=11, y=3.5 meters with an angle of -10 degrees.

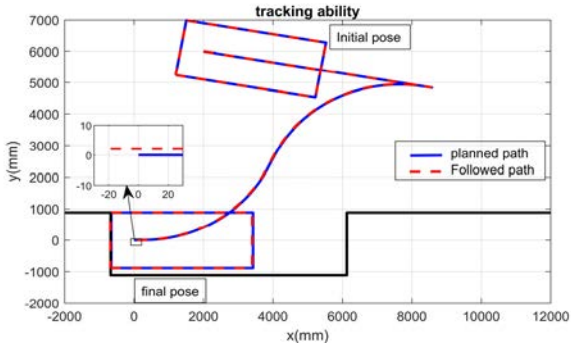

FIGURE 10. The tracking ability of the designed controller.

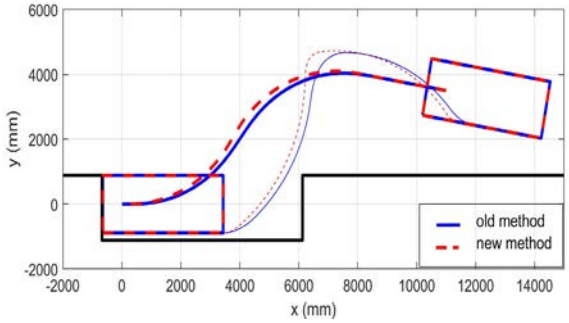

FIGURE 11. Parallel parking in one maneuver: Comparison of the introduced method with the old method.

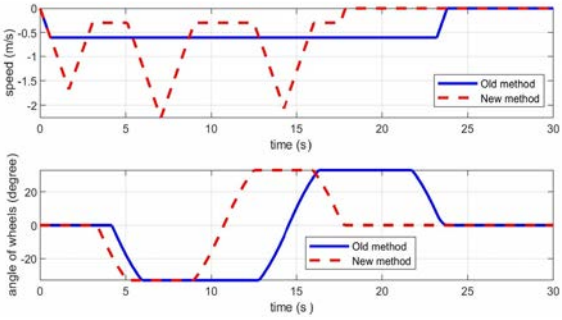

FIGURE 12. Comparison of the speed and steering angle values in two methods for automatic parking with one maneuver.

Based on the calculations, the coordinates of the correction point are obtained as (8.2,4). In this case, the correction point occurs on the left side of the initial pose of the vehicle. Therefore, only one maneuver is needed. Considering the possibility of a collision between the right front corner of the car and the highest front corner of the parking spot, the movement path of this corner of the car is plotted. As seen in Fig. 11, the car corner is more distant from the border of the parking spot in the new method. Thus, the new method requires less space for the parking maneuver compared to the old method proposed in [20]. The space requirement is reduced by 25 centimeters in this case. In Fig. 12, the obtained results of those methods are displayed, including the speed and steering angle. A negative speed refers to backward movement for parking maneuvers. In this figure, the reduction of the vehicle parking duration is seen as 7.5 seconds for the new method due to utilizing an optimized speed via apply-

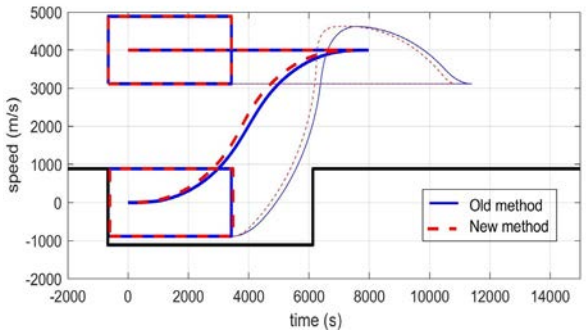

FIGURE 13. Parallel parking in two maneuvers: A comparison of the proposed method with the old method.

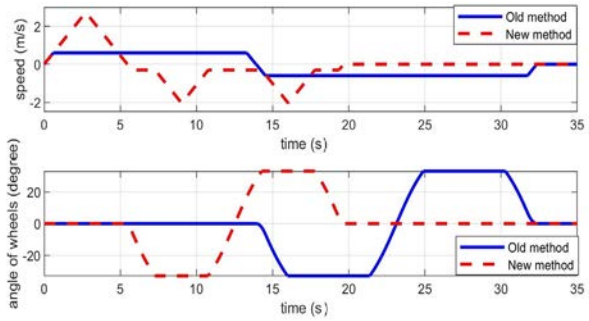

FIGURE 14. Comparison of the speed and steering angle values of two methods for parking with two maneuvers.

ing the proposed MOPSO algorithm. Hence, in the proposed method, the required space for parking the car is reduced by 25 centimeters and the parking duration has decreased by 32%, simultaneously.

C. SCENARIO 3: CAR PARKING WITH TWO MANEUVERS

In this example, the initial pose of the car is considered to be x=0 and y=4 at 0 degrees. Based on the calculations, the coordinates of the correction point are (8,4). In this case, the correction point occurs on the right side of the initial pose. Thus, the car must first go straight to the correction point and then continue the parking by moving backward. Fig. 13 compares the parallel parking with two maneuvers by the new and old methods. According to the figure, the vehicle can be parked in a smaller space by using the new method, saving approximately 25 centimeters. Fig. 14 also compares the design of the controllers of the two methods. As shown in this figure, the velocity is positive at first for both methods, and then becomes negative afterward, which in turn refers to moving forward and backward. As in the previous example, the steering angle is initially zero that later becomes positive and then negative. This figure also shows a 50% reduction in the parking duration for the proposed method.

D. SCENARIO 4: EVALUATING THE PERFORMANCE OF THE PROPOSED METHOD USING THE MONTE CARLO APPROACH

As the length of the curves depends on the maximum velocity of the clothoid, the required space length for all maneuvers (regardless of the error) would be a specific amount by choosing a constant clothoid velocity. For example, if the value of $V_{clothoid}$ is reduced from 0.6 to 0.3, the length of the required space for parking the car decreases by 25 centimeters for each starting point. The Monte Carlo method has been proposed and used to calculate the average parking duration. In the following, the Monte Carlo method is explained. Then, the performance of this system is evaluated.

The Monte Carlo method examines the statistical function of the system and has many applications, one of which is to evaluate the system's performance in the presence of uncertainty. In this method, the range of selecting the variables is specified, and then, a large number of system variables are generated randomly and independently with a normal distribution, and the system output uncertainty is examined [40], [41], [42], [43]. The Monte Carlo method can be summarized in the following four steps:

1) Extracting and defining the scope of possible inputs
2) Generating random inputs within the defined range
3) Simulating the system's behavior for randomly selected inputs
4) Performing a statistical analysis of the results

In this problem, the initial pose of the car for the parking maneuver is uncertain. The Monte Carlo method can be used to evaluate the performance of the system with the uncertainty of the vehicle initial pose. Thus, a lot of initial poses are randomly generated, and the system performance is examined for each of these situations. Due to the complexity of the problem, in this case, 1000 different initial poses are selected randomly. These 1000 samples are selected in a range in which all the operations would be successful. The parking maneuver is performed for each of these starting points and the statistical specifications of these 1000 samples are computed. The range of variations for this study is as follows:

$$\begin{cases} -2 < x_{init} < 17 & (m) \\ 4 < y_{init} < 8 & (m) \\ -10 < \theta_{init} < 10 & (\text{deg}) \end{cases} \tag{21}$$

For example, Fig. 15 shows the parking maneuver for 5 different random initial poses in one or two maneuvers. The simulation statistical results for 1000 random starting points are indicated in Table 2. According to the results, the average parking duration is reduced by 42% and the average length of the required space for the car parking is decreased by 25 centimeters. Given that the length of the standard parking spot is 6.1 meters, the vehicle can be parked with one or two maneuvers in locations with standard dimensions using the proposed method, which was not possible by the old method.

E. SCENARIO 5: COMPARISON OF NONLINEAR FILTERS PERFORMANCE

In this scenario, the performance of the nonlinear UKF is compared with the EKF filter, with equal noise conditions

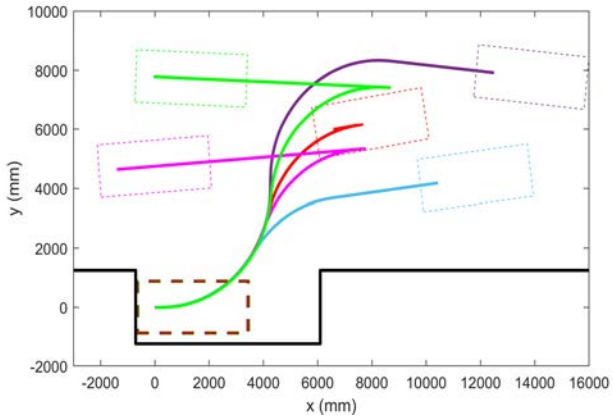

FIGURE 15. Automatic parking maneuver starting from different poses.

TABLE 2. Evaluating the presented method with the old method.

Criterion	Old method	New method
Mean value of parking time (s)	30.82	17.92
The standard deviation of parking time (s)	5.62	1.85
The average length of required space for parking (m)	6.25	6.00
The standard deviation of the length of required space for parking (m)	0.004	0.013

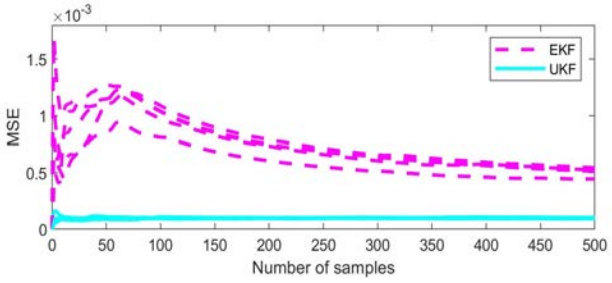

FIGURE 16. Comparison of MSE of EKF and UKF estimators for 5 random initial poses.

considered for the two systems. The Monte Carlo method is used to compare 1000 samples of the performance of estimators which are used to estimate the traveled distance. Fig. 16 exhibits the MSE of estimating the traveled distance by two estimators for 5 random samples, in which, the noise covariance matrix of the system and observations are diagonal with the intensities of 10^{-8} and 10^{-5}, respectively. In addition, Table 3 highlights the statistical comparison of these methods.

From Fig. 16 and Table 3, it is evident that the UKF estimator has significantly lower error than the EKF estimator. Hence, the UKF filter is a more suitable estimator in this scenario. The error of the UKF filter mainly depends on the first component of the system's covariance noise matrix, and a smaller value of this component results in lower filter error. Although the error also depends on other components of the covariance noise of the system and the observations, the maximum error of UKF is only 3.5 centimeters in 1000 samples, which is negligible. Therefore, the UKF filter is a more appropriate choice for estimating the traveled distance.

TABLE 3. Comparison of the final situation errors of EKF and UKF.

Statistical Performance		EKF	UKF
Mean error	x (mm)	125.6	16.3
	y (mm)	89.2	5.3
	θ (degree)	0.89	0.05
Standard deviation error	x (mm)	93.9	10.6
	y (mm)	72.6	4.1
	θ (degree)	0.70	0.04

VII. CONCLUSION

This work proposed an optimized algorithm to address the problem of automatic parallel parking for vehicles. The method includes path planning, controller design, and traveled distance estimation. One feature of the paper is the design of an optimal maneuver to enable the car to park in smaller spaces with one or two maneuvers, significantly reducing the parking duration. The UKF filter has also been used to estimate the traveled distance, which is a simple, robust, and optimal filter with a reasonable error. Different simulations have been carried out for the proposed method, which demonstrates the effectiveness of the method. It is important to note that the assumption is that a clear parking spot has been identified, as there is no room for collision avoidance maneuvers during parallel parking. If an obstacle is detected, a new parking spot must be found, and the parking maneuver needs to be repeated. The proposed technique involves mostly offline calculations, and its algorithm allows for relatively fast online implementation. For future research on parking in smaller spaces, one could build upon the proposed method to further reduce the number of required maneuvers and increase the speed of parking maneuvers, ultimately improving efficiency in parking. A similar process can be applied for the optimal design of vertical parking or automatic path generation in limited spaces.

APPENDIX

At first, the parameter A_{min} must be calculated according to: $A^2_{\min} = R_{\min}L_{\min}$. The parking maneuver consisting of the clothoid sequence is shown in Fig. 6. Obtaining the exact path inevitably involves calculating μ and R_1. First, a point is considered on the path. Every point on the clothoid path with component A has the following characteristics [16]:

$$q_r : \begin{cases} x_r = A\sqrt{\pi}C_f(\frac{A}{R\sqrt{\pi}}) \\ y_r = A\sqrt{\pi}S_f(\frac{A}{R\sqrt{\pi}}) \\ \theta_r = \frac{A^2}{2R^2} \\ \delta_r = \frac{1}{R} \end{cases} \tag{22}$$

where C_f and S_f are Fresnel integrals, equal to $C_f(x) = \int_0^x \cos((\pi/2)u^2)du$ and $S_f(x) = \int_0^x \sin((\pi/2)u^2)du$, respec-

tively. The proposed point is located at a distance of L_{min} from the origin of the coordinates, at which the radius is equal to R_{min}. In this case, the coordinates of the center of the circle C_l in Fig. 6 are calculated as follows:

$$x_{cl} = x_r - R\sin\theta_r, \quad y_{cl} = y_r + R\sin\theta_r \tag{23}$$

Therefore, μ and R_1 will be calculated as follows:

$$R_1 = \sqrt{x_{cl}^2 + y_{cl}^2}, \mu = \arctan\left(x_{cl}/y_{cl}\right) \tag{24}$$

The correction point is obtained from the following equation:

$$3R_1^2 - d_{CE}^2 + 2dR_1\cos\alpha = 0 \tag{25}$$

where:

$$\begin{cases} d_{CE} = \sqrt{(x_{Cl} - x_{Ecorrect})^2 + (y_{Cl} - y_{Ecorrect})^2} \\ \alpha = \arccos\left((y_{Ecorrect} - y_{Cl})/d_{CE}d_{CE}\right) - \mu + \theta_{E_{init}} \\ y_{Ecorrect} = y_{E_{init}} - (x_{E_{init}} - x_{Ecorrect})\tan\theta_{E_{init}} \end{cases} \tag{26}$$

According to Fig. 6, the angles and lengths of each of the curves are calculated using (7) and the law of cosines:

$$\begin{aligned} \theta_{cr} &= \cos^{-1}\left(\frac{5R_1^2 - d_{CE}^2}{4R_1^2}\right) \\ \theta_{cl} &= \theta_{cr} + \alpha + \cos^{-1} \\ &\quad \times \left(\frac{d_{CE}^2 + R_1^2 - \left(x_{Ecorrect}^2 + y_{Ecorrect}^2\right)}{2d_{CE}R_1}\right) - \pi \\ \alpha_{clotho} &= \cos^{-1}\left(\frac{R_{\min}^2 + R_1^2 - (x_r^2 + y_r^2)}{2R_{\min}R_1}\right) \end{aligned} \tag{27}$$

Therefore, the length of the entire curve path is calculated as:

$$\begin{cases} L_{arc} = R_{\min}\alpha_{arc} \\ L_{seq} = L_{arc} + 2L_{\min} \end{cases}, \quad \alpha_{arc} = \alpha_{tot} - 2\alpha_{clotho} \tag{28}$$

REFERENCES

[1] VC Support. (May 2022). *Parallel Parking With Park Assist Pilot*. [Online]. Available: https://support.volvocars.com/lb/Pages/article.aspx?article=4c4380adf3f56bc8c0a801510c91778c

[2] Toyota. (May 2022). *Parking Aids*. [Online]. Available: https://www.toyota-europe.com/world-of-toyota/safety-technology/parking-aids

[3] Carbuyer. (May 2022). *Best Self-Parking Cars*. [Online]. Available: http://www.carbuyer.co.uk/reviews/recommended/best-self-parking-cars

[4] M. Sugeno, T. Murofushi, T. Mori, T. Tatematsu, and J. Tanaka, "Fuzzy algorithmic control of a model car by oral instructions," *Fuzzy Sets Syst.*, vol. 32, no. 2, pp. 207–219, Sep. 1989.

[5] F. Gómez-Bravo, F. Cuesta, and A. Ollero, "Parallel and diagonal parking in nonholonomic autonomous vehicles," *Eng. Appl. Artif. Intell.*, vol. 14, no. 4, pp. 419–434, Aug. 2001.

[6] T. S. Li, S.-J. Chang, and Y.-X. Chen, "Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot," *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, pp. 867–880, Oct. 2003.

[7] Y. Zhao and E. G. Collins, "Robust automatic parallel parking in tight spaces via fuzzy logic," *Robot. Auto. Syst.*, vol. 51, nos. 2–3, pp. 111–127, May 2005.

[8] B. Li, Z. Yin, Y. Ouyang, Y. Zhang, X. Zhong, and S. Tang, "Online trajectory replanning for sudden environmental changes during automated parking: A parallel stitching method," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 748–757, Sep. 2022.

[9] J. P. Laumond, S. Sekhavat, and F. Lamiraux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot Motion Planning and Control*, J. P. Laumond, Ed. Berlin, Germany: Springer, 1998, pp. 1–53.

[10] B. Muller, J. Deutscher, and S. Grodde, "Continuous curvature trajectory design and feedforward control for parking a car," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 541–553, May 2007.

[11] I. E. Paromtchik and C. Laugier, "Motion generation and control for parking an autonomous vehicle," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1996, pp. 3117–3122.

[12] C. Laugier, T. Fraichard, P. Garnier, I. E. Paromtchik, and A. Scheuer, "Sensor-based control architecture for a car-like vehicle," *Auto. Robots*, vol. 6, pp. 165–185, Apr. 1999.

[13] L. Cai, H. Guan, Z. Y. Zhou, F. L. Xu, X. Jia, and J. Zhan, "Parking planning under limited parking corridor space," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1962–1981, Feb. 2023.

[14] S. Choi, C. Boussard, and B. d'Andrea-Novel, "Easy path planning and robust control for automatic parallel parking," *IFAC Proc.*, vol. 44, pp. 656–661, Jan. 2011.

[15] T. Inoue, D. M. Quan, and L. Kang-Zhi, "Development of an auto-parking system with physical limitations," in *Proc. SICE Annu. Conf.*, 2004, pp. 1015–1020.

[16] W. Nelson, "Continuous-curvature paths for autonomous vehicles," in *Proc. Int. Conf. Robot. Autom.*, 1989, pp. 1260–1264.

[17] D. Lyon, "Parallel parking with curvature and nonholonomic constraints," in *Proc. Intell. Vehicles Symp.*, 1992, pp. 341–346.

[18] A. Scheuer and T. Fraichard, "Continuous-curvature path planning for car-like vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot Syst. Innov. Robot. Real-World Appl.*, Sep. 1997, pp. 997–1003.

[19] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Trans. Robot.*, vol. 20, no. 6, pp. 1025–1035, Dec. 2004.

[20] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 396–410, Feb. 2015.

[21] B. Li, K. Wang, and Z. Shao, "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3263–3274, Nov. 2016.

[22] J. Moon, I. Bae, and S. Kim, "Real-time near-optimal path and maneuver planning in automatic parking using a simultaneous dynamic optimization approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 193–196.

[23] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[24] B. Li and Z. Shao, "A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles," *Knowl.-Based Syst.*, vol. 86, pp. 11–20, Sep. 2015.

[25] D. Qiu, D. Qiu, B. Wu, M. Gu, and M. Zhu, "Hierarchical control of trajectory planning and trajectory tracking for autonomous parallel parking," *IEEE Access*, vol. 9, pp. 94845–94861, 2021.

[26] J. Zhang, H. Chen, S. Song, and F. Hu, "Reinforcement learning-based motion planning for automatic parking system," *IEEE Access*, vol. 8, pp. 154485–154501, 2020.

[27] R. Chai, A. Tsourdos, A. Savvaris, H. Chai, Y. Xia, and C. L. P. Chen, "Design and implementation of deep neural network-based control for automatic parking maneuver process," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1400–1413, Apr. 2022.

[28] L. Junzuo and L. Qiang, "An automatic parking model based on deep reinforcement learning," *J. Phys., Conf. Ser.*, vol. 1883, no. 1, Apr. 2021, Art. no. 012111.

[29] L. Junzuo and L. Qiang, "An automatic parking model based on deep reinforcement learning," *J. Phys., Conf.*, vol. 1883, no. 1, Apr. 2021, Art. no. 012111.

[30] S. Rafique, S. Gul, K. Jan, and G. M. Khan, "Optimized real-time parking management framework using deep learning," *Exp. Syst. Appl.*, vol. 220, Jun. 2023, Art. no. 119686.

[31] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2584–2601, Oct. 2017.
[32] M. Brady, *Robot Motion: Planning and Control*. Cambridge, MA, USA: MIT Press, 1982.
[33] R. Rajamani, *Vehicle Dynamics and Control*. Berlin, Germany: Springer, 2011.
[34] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.
[35] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
[36] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE*, vol. 3068, pp. 182–193, Jul. 1997.
[37] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. Adapt. Syst. Signal Process., Commun., Control Symp.*, 2000, pp. 153–158.
[38] P. Bonnifait, P. Bouron, P. Crubille, and D. Meizel, "Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2001, pp. 1597–1602.
[39] R Group. (May 10, 2018). *ZOE Design*. [Online]. Available: https://www.renault.ie/vehicles/new-vehicles/zoe/design.html
[40] P. Williams, "A Monte Carlo dispersion analysis of the X-33 simulation software," in *Proc. AIAA Atmos. Flight Mech. Conf. Exhib.*, Aug. 2001, pp. 6–9.
[41] C. P. Robert, *Monte Carlo Methods*. Hoboken, NJ, USA: Wiley, 2004.
[42] E. Baumann, C. Bahm, B. Strovers, R. Beck, and M. Richard, "The X-43A six degree of freedom Monte Carlo analysis," in *Proc. 46th AIAA Aerosp. Sci. Meeting Exhib.*, Jan. 2008, p. 2008.
[43] J. Enayati, P. Sarhadi, M. P. Rad, and M. Zarini, "Monte Carlo simulation method for behavior analysis of an autonomous underwater vehicle," *Proc. Inst. Mech. Eng., M, J. Eng. Maritime Environ.*, vol. 230, no. 3, pp. 481–490, Aug. 2016.

SAEEDE MOHAMMADI DANIALI received the B.S. degree in electronic engineering and the M.S. degree in control engineering from the Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran, in 2015 and 2018, respectively.

Since 2017, she has been a Researcher with the Babol Noshirvani University of Technology. Her research interests include automatic control and intelligent systems.

ALIREZA KHOSRAVI received the Ph.D. degree in control engineering from the Iran University of Science and Technology (IUST), Iran, in 2008. He is currently an Associate Professor with the Department of Electrical Engineering, Babol Noshirvani University of Technology, Babol, Iran. His research interests include robust and optimal control, modeling and system identification, and intelligent systems.

POURIA SARHADI received the Ph.D. degree in control engineering from the Babol Noshirvani University of Technology (BNUT), in 2016. He was an Invited Lecturer with BNUT (2016–2018). He was a Research Fellow with the University of Surrey, U.K. (2019–2021), and Queen's University Belfast, U.K. (2021–2022). He is currently a Senior Lecturer with the School of Physics, Engineering and Computer Science, University of Hertfordshire, U.K. He has held key roles in the European Commission's Horizon 2020 and U.K. Research and Innovation (UKRI) projects. He is also recognized as a U.K. "Global Talent" by the UKRI. Over the past 15 years, he has been involved with research in autonomous vehicles (marine, aerial, and ground) leading to more than 40 journal articles. His research interests include control theory and applications, adaptive control, system identification, machine learning, autonomous vehicles, and systems engineering.

FATEMEH TAVAKKOLI received the M.S. degree in control engineering from the Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran, in 2018.

Since 2018, she has been a Researcher with the Babol Noshirvani University of Technology. Her research interests include adaptive control, robust control, and intelligent systems.