2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Real-Time near-optimal Path and Maneuver Planning in Automatic Parking Using a Simultaneous Dynamic Optimization Approach

Jaeyoung Moon, *Student Member, IEEE*, Il Bae, *Student Member, IEEE*,
and Shiho Kim, *Senior Member, IEEE*

Abstract—**We propose a real-time near-optimal motion and trajectory planner using an interior-point method (IPM) based simultaneous dynamic optimization approach applicable to automatic parallel parking. Partitioning an automatic parking zone into two areas with and without inequality constraints drastically improves the convergence efficiency of the IPM-based simultaneous dynamic optimization of nonlinear programs while retaining the accuracy and stability of simulation for optimization. An optimal ready-to-reverse point (RRP), from where backward moving parking maneuver starts, is provided based on the cost function of the tracking time and additional delayed time of gear shifts caused by multiple maneuvers. The stability and accuracy of the proposed method was verified through simulation using AMPL with an IPOPT solver.**

I. Introduction

Autonomous parking systems have been widely investigated as a means of improving driver comfort and safety, because parking in a small tiny space is a difficult task even for a skilled driver. Autonomous vehicles must perform numerous tasks to safely park the vehicle in a narrow space, including precise environment detection and parking maneuvers. Generally, autonomous parking systems consist of three steps: detection of a free parking space, path planning and maneuver decision, and tracking the desired trajectory [1]. In this paper, we limit the scope of research to path planning and maneuvering parameters, under the assumption that the vehicle has surrounding spatial information of free parking space. The geometric or heuristic approaches are prevailing methods used to generate path and maneuvering parameters for automatic parking [2]. Geometric methods are composed of line segments and circular arcs calculated from simple geometrical equations [3]. To retain a continuity of curvature, circle arcs and Clothoidal paths were combined [4-5]. Many methods are presented for parking trajectory using splines, polynomials and Bezier curve [6-9]. And a retrieving approach was proposed based on retrieving the vehicle from the parking spot and reversing the obtained path to park the vehicle [7], [10]. The second type of methods concerns heuristics, including machine learning, search theory and fuzzy logic to learn a human's technique [11-12]. Those methods can be an effective solution to deal with uncertainties and inaccuracies in the mapping of the environment. But those heuristic methods still need to be studied further to provide a real-time automatic parking solution in a tiny space.

Recently, an interior-point method (IPM) based simultaneous dynamic optimization methodology, which can be classified as numerical methods, that includes vehicle kinematics and collision-avoidance constraints has been adopted to generate time-optimal parallel parking maneuvers in a straightforward manner [13]. Despite the merits in terms of accuracy and unification, the computation time of the conventional IPM-based optimization method is not short enough to apply to a real-time generation of path and maneuver in on-site for automatic parking.

By considering an urban traffic situation, we may roughly assume that the computation time allowed to generate a path and control maneuver must be less than 2.0s. This study provides a technique for real-time path planning using an IPM-based simultaneous dynamic optimization of nonlinear programs (NLPs). In the previous approach, the NLP solver must handle a set of both inequality and equality constraints in order to solve optimization problems for the entire parking zone [13-14] because they have tried to discretize the entire parking area without partitioning. However, in the proposed approach, we split the parking area into two maneuvering zones reflecting collision possibilities based on geometric approach [8]. It can simplify the numerical complexity of the non-linear optimization problem by converting the number of inequality constraints into boundary condition of variables. Hence we can reduce the computational overhead of conventional the IPM-based optimization method.

This paper is organized into five sections. Section II describes kinematic model of a vehicle based on its geometric relation and IPM-based simultaneous dynamic optimization problem is presented. In section III, the parking problem sets for dynamic optimization are explained in detail. Simulation results and discussions are presented in section IV. Finally, the conclusions are provided in section V.

II. IPM-Based Dynamic Optimization Problem

Using the Ackerman's bicycle kinematic model of a front wheel steered vehicles shown in Fig. 1, we can obtain the following set of equations [13-15],

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x(t) \\ y(t) \\ v(t) \\ \Psi(t) \\ \delta(t) \end{pmatrix} = \begin{pmatrix} v(t)\cdot\cos\Psi(t) \\ v(t)\cdot\sin\Psi(t) \\ a(t) \\ v(t)\cdot\tan\delta(t)\,/l_w \\ \omega(t) \end{pmatrix}, \forall t \in [0, t_f] \qquad (1)$$

where $\mathrm{t_f}$ indicates the completion time of the entire process, (x, y) denotes a coordinate of E (the mid-point of the

* This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the "ICT Consilience Creative Program" (IITP-R0346-16-1008) supervised by the IITP (Institute for Information & communications Technology Promotion)

The authors are with the Seamless Transportation Lab (STL), School of Integrated Technology, and Yonsei Institute of Convergence Technology, International Campus, Yonsei University, Incheon 21983, Korea (e-mail:; ekuno90@yonsei.ac.kr, kakao@yonsei.ac.kr, and shiho@yonsei.ac.kr).

978-1-5090-4804-5/17/$31.00 ©2017 IEEE

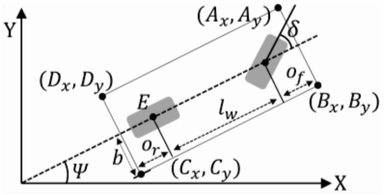


Figure 1. Ackerman's bicycle kinematic model of front-steering vehicles.

TABLE I
NOTATION OF THE TEST VEHICLE SPECIFICATION AND SETTINGS

Symbol	Definition	Value
PL	*Parking slot length*	- (m)
PW	*Parking slot width*	2 (m)
LW	*Lane width*	3.5 (m)
$2b$	*Overall width*	1.6 (m)
l_w	*Wheel base*	2.52 (m)
$o_{f/r}$	*Front/Rear overhang*	0.54/0.54 (m)
v_{max}	*Bound of velocity*	2 (m/s)
a_{max}	*Bound of acceleration*	0.75 (m/s^2)
δ_{max}	*Bound of steering angle*	33 (°)
w_{max}	*Bound of angular velocity*	1.0 (rad/s)
G_t	*Elapsed time on gear-shift*	1 (s)

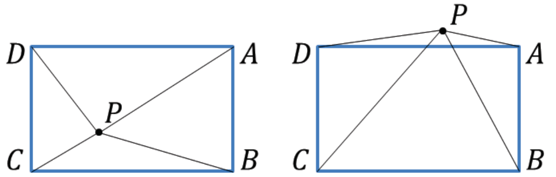
rear-wheel axis), $v(t)$ is the velocity of point E, $a(t)$ denotes corresponding acceleration, $\Psi(t)$ is the yaw angle of the vehicle body, $\delta(t)$ refer to the steering angle of the front wheel, and $\omega(t)$ is the angular velocity of the front wheel. We can calculate the coordinates of four edges, (A_x, A_y), $(B_x\ B_y)$, (C_x, C_y), (D_x, D_y), at any moment using the geometric relation of the yaw angle and vehicle parameters of the rigid body as shown in Fig. 1 [13]. The parametric notations of simulated vehicle are described in Table I.

By discretizing control and state variables, the original problem is transformed into an IPM-based simultaneous nonlinear dynamic optimization problem, which is equivalent to a fully implicit Runge-Kutta method [13]. We can formulate a discretized finite-dimensional nonlinear programming for a given number of elements N_{fe} as a framework of simultaneous optimization as following equations [13-14, 16].

min $\phi(z_{ij}, n_{ij}, u_{ij})$ subject to

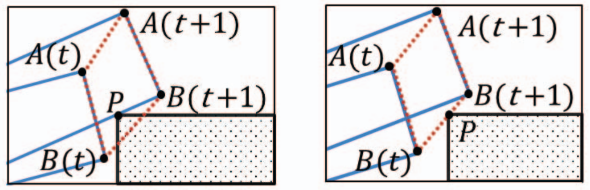
$$\begin{gathered}\sum_{k=0}^{K}\left(\frac{d\phi_j(\tau)}{d\tau}\Big|_{\tau_k}\cdot z_{ik}\right) - hF\left(z_{ij}, n_{ij}, u_{ij}\right) = 0\\ \phi_j(\tau) = \prod_{k=0,\neq j}^{K}\frac{\tau-\tau_k}{\tau_j-\tau_k}\\ G\left(z_{ij}, n_{ij}, u_{ij}\right) \leq\ 0,\ H\left(z_{ij}, n_{ij}, u_{ij}\right) =\ 0\\ i\ =\ 1,2\ldots,N_{\mathrm{fe}},\ j\ =\ 1,2,\ldots,K, h = t_{\mathrm{f}}/N_{\mathrm{fe}}\end{gathered} \tag{2}$$

where $\phi(\cdot)$ refers to the minimization criterion; z_{ij}, n_{ij} and u_{ij} are the discretized differential state variables, algebraic state variables, and control variables, respectively. Also, $G\left(z_{ij}, n_{ij}, u_{ij}\right)$ and $H\left(z_{ij}, n_{ij}, u_{ij}\right)$ are functions to represent all restrictions of algebraic equalities and inequalities. Radau points τ_k $(k = 1, \ldots, K)$ are chosen to satisfy orthogonal properties when collocation point K is determined. Due to the page limit, we skipped discussing the detailed formulation procedure of the framework equations. We recommend prior works [13-14, 16] for more details about the mathematical formulation procedure.



(a) (b)

Figure 2. Schematic showing a collision decision-method, (a) collision case, (b) non-collision case.



(a) (b)

Figure 3. Schematic showing a decision-method for a Collision with P, (a) the collision in two consecutive time steps of discretized elements and (b) an additional collision free judgment condition.

III. PROBLEM SETS FOR PARALLEL PARKING

Firstly, we must determine a parking maneuver for the autonomous parking system based on the forward path generation and backward tracking algorithm [1]. A forward path is generated to escape from the destination parking point, heading-out to the starting point of automatic parking which is RRP(ready to reverse point) of parallel parking maneuver.

A. *Cost function of parallel parking*

The cost function(C_f) of the minimization criterion of the optimization is the completion time of automatic parking(t_f) which is a time required to move vehicle from a target space to RRP. We considered the number of gear-shift operation to calculate the minimum cost considering the vehicle sequential motion of the multi maneuvering parking situation. The resulting cost of the optimization is formulated as

$$C_f = t_{\mathrm{f}} =\ t_{\mathrm{m}} + g_t * \sum G_c(t) \tag{3}$$

Where sum of $G_c(t)$ is the number of gear-shift, t_m is the driving time under the parking and $\mathrm{g_t}$ is an operation time for the gear shift during the parking. $G_c(t)$ is determined as below:

$$G_c(t) = \begin{cases}1, & if\ v(t) * v(t-1) < 0\\ 0, & else\end{cases}, \forall t \in [0, t_f] \tag{4}$$

B. *Collision avoidance condition*

The triangle area judgment method has been described with obstacle P and the vehicle rectangular area of *A, B, C,* and *D*, as shown in Fig. 2 [14]. However, the conventional method only provides sufficient condition for detecting collision among static objects. Although each state satisfies the collision-free condition, a collision may occur if the vehicle moves during the two consecutive steps as shown in Fig. 3. To prevent collision for 2 consequent time steps of discretized elements, the point P remains outside of the rectangle formed with A(t), A(t+1), B(t) and B(t+1). We formulated a collision free condition by extending the triangle area method for moving objects in simultaneous optimization to yield,

$$S_{\Delta PAB} + S_{\Delta PBB'} + S_{\Delta PB'A'} + S_{\Delta PA'A} > S_{\square ABB'A'} \tag{5}$$

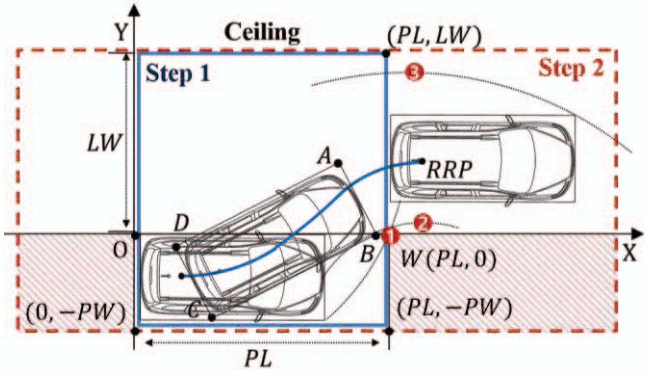


Figure 4. Proposed partitioning method by considering 3-collision possibility (case 1 for Edge B with W, case 2 for Edge C with W and case 3 for edge A with ceiling) during the parallel parking maneuver.

C. Definition of maneuvering zone

In the previous approach, all the state and control profiles were discretized for entire parking area [13-14]. Therefore, the NLP solver must handle a set of both inequality and equality constraints when solving optimization problems for the entire parking zone. In the proposed approach, we split the parking area into two maneuvering zones, as shown in Fig. 4. The first area (Step 1) is a square area defined by 2 coordinates of (0, -*PW*) and (*PL*, *LW*), and the second area (Step 2) is the remaining part of the maneuvering zone. The main benefit of the partition is that we can solve the dynamic problem without inequalities in the first area during Step 1. If there are only equality constraints in the nonlinear optimization problem, we can solve the optimization using the Lagrangian filter instead of the Karush-Kuhn-Tucker (KKT) conditions. The Lagrange condition requires less computing power converge into feasible regions compared to KKT conditions [16]. Three collision possibilities exist in a parallel parking geometry marked as ①, ② and ③ as shown in Fig. 3 [8]. If we limit the area of optimization by setting a boundary condition within the first area in Step 1, the necessary and sufficient collision-free condition between the vehicle and point W marked as ① is satisfied without inequality constraints normally generated by the triangular judgment method illustrated in Fig. 2. The terminal condition of the first area is that the y coordinate of the *B* corner is greater than 0, and the *x* coordinate of *B* is equal to *PL*, which means that the vehicle is at the edge of the first area toward RRP. The optimized path and parameters from the first area provide the boundary and initial conditions of Step 2, where inequality equation of collision-free constraints against the point W marked as ② should be applied to solve nonlinear optimization. For avoiding the collision ③, the y coordinate of four vertices of *A, B, C* and *D* should be less than the lane width *LW*.

IV. Simulation results and Discussions

To show the convergence and feasibility of the proposed method, simulations under various parking condition and environment were conducted using AMPL (A Mathematical Programming Language) [17] with solver IPOPT, which is an open-source software package of IPM, in the 3.8.0 version [16] and executed on an Intel Core i7-3770 CPU with 8GB RAM. We changed the parameters for *PL* while the dimensions of *PW* and *LW* were fixed at 2 and 3.5 m, respectively. The overall length and width of the vehicle were 3.6m and 1.6m, respectively. The bound of variables were: v_{max} = 2 (m/s), a_{max} = 0.5 (m/s^2), δ_{max} = 33°, and angular velocity ω_{max} = 1 (rad/s), in addition, g_t was set to 1.0s.

TABLE II
Simulated results for N_{FE}=40 (partition when N_1/N_2=30/10)

Case	*PL*	Generated t_F (s)		$\sum G_c$ (Numbers)		Computation time (s)	
		**WP*	***WOP*	**WP*	***WOP*	**WP*	***WOP*
1	6	5.3	5.3	0	0	0.97	30.53
2	5.4	5.4	5.4	0	0	1.02	31.42
3	4.8	9.9	9.9	2	2	1.33	36.36
4	4.2	18.8	18.8	6	6	1.53	45.25

(**WP* = with partition, ***WOP* = without partition)

TABLE III
Simulated results with/without partition of case 4

	N_{fe} (N_1/N_2)	# of equalities (N_1/N_2)	# of inequalities (N_1/N_2)	t_f (s) (N_1/N_2)	Computation time (s) (N_1/N_2)
WP	12 (9/3)	604 (460/144)	64 (0/64)	21.9 (17.1/4.8)	0.45 (0.28/0.17)
WOP	12 only	616	276	21.6	3.01
WP	20 (15/5)	1020 (772/248)	110 (0/110)	18.6 (14.9/3.7)	0.69 (0.41/0.28)
WOP	20 only	1032	460	18.8	10.08
WP	40 (30/10)	2060 (1552/508)	225 (0/225)	18.8 (15.0/3.8)	1.53 (1.08/0.45)
WOP	40 only	2072	920	18.8	45.25
WP	80 (60/20)	4140 (3112/102)	455 (0/455)	18.7 (15.0/3.7)	4.19 (3.39/0.80)
WOP	80 only	4152	1840	18.7	151.69

(**WP* = with partition, ***WOP* = without partition)

We simulated the cases of *PL* from 6 to 4.2 m in increments of 0.2 m and varied N_{fe} from 20 to 200 in increments of 20 to check the feasibility and stability of the real-time generation of path and control parameters for automatic parking. We verified that the simulation converged for all cases without any problems, thus revealing the stability of the proposed method. Generated time, path, and maneuver control parameters from the simulation were nearly the same for the proposed and conventional nonlinear optimization methods, whereas the computation time of the proposed method was approximately 30 times less than that of the conventional method without partition.

Table II shows that the proposed method achieved optimization in less than 2.0 s, which is satisfactory for real-world applications. Table III shows the results with different finite elements under the condition of *PL* = 4.2, which causes multiple gear-shift maneuvering in the first region during step 1. These results reveal that the proposed problem converges to feasible solutions for various finite elements even under the harsh conditions of multiple gear shifts.

Moreover, the computation time drastically increases with each increment of N_{fe}. However, the simulated completion time (t_f) shows a trend of the asymptotic property approaching the near-optimal value of 18.7 s when N_{fe} is greater than 20. The number of inequality conditions is a main concern of computation time. For example, when N_{fe} is 40, we must manage 2,072 equality and 920 inequality conditions simultaneously, which generates a computation time of 45.25 s. By contrast, a partition with (N_1/N_2=30/10) reduces the

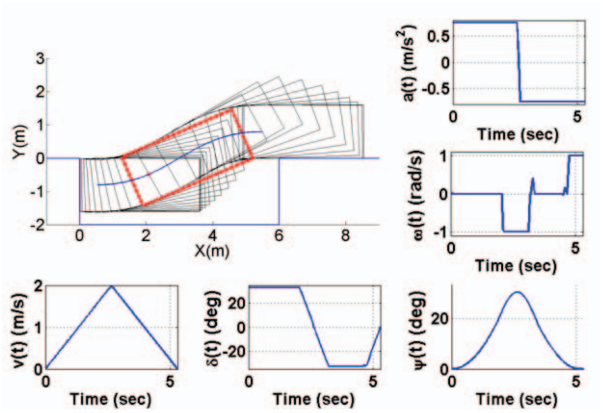


Figure 5. Simulated results using the proposed approach for case 1 (*PL*=6) shown in Table II: Generated trajectory and control/state profiles.

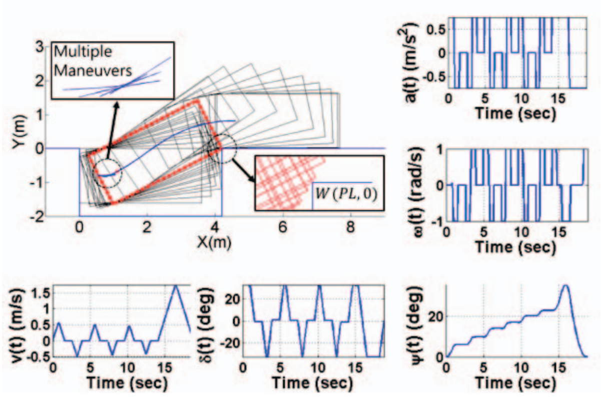


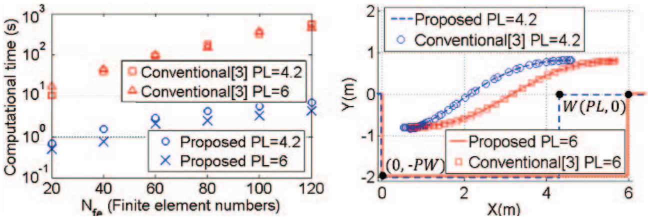
Figure 6. Simulated results using the proposed approach for case 4 (*PL*=4.2) shown in Table II: Generated trajectory and control/state profiles.

inequality conditions to 225 with nearly the same number of equalities. The computation time of step 1 with equality constraints of 1,552 is only 1.08 s, whereas step 2 with equality constraints of 508 and inequality constraints of 225 consumes 0.45 s for computation. Computational efficiency derives from the proposed partitioning method.

Fig. 5 and 6 show the simulated results of generated path and control/state profiles for the *PL*=6 of case 1 and *PL*=4.2 of case 4, where two auxiliary boxes show the enlarged view of multiple maneuvering and the condition of collision avoidance in the first area. Fig. 7 (a) shows comparisons of simulated results from the proposed and conventional methods. The computation time is approximately 30 times less than that of the conventional method, whereas the generated motion and trajectory planning are nearly the same as shown in Fig. 7 (b).

V. Conclusions

We propose a real-time near-optimal motion and trajectory planner using an interior-point method (IPM) based simultaneous dynamic optimization applicable to automatic parallel parking. Partitioning an automatic parking zone into two areas with and without inequality constraints drastically improves the convergence efficiency of the IPM-based simultaneous dynamic optimization of nonlinear programs while retaining the accuracy and stability of simulation for optimization. The efficiency and accuracy of the proposed method was verified through computer simulation using the AMPL with the IPOPT solver.



(a) (b)

Figure 7. Comparison of proposed and conventional methods (a) computational time versus finite elements numbers, (b) generated path of $\boldsymbol{E(x,y)}$ for the case 1 (*PL*=6) and case 4 (*PL*=4.2).

References

[1] M. Y. I. Idris, “Car Park System: A Review of Smart Parking System and its Technology,” Information Technology Journal, vol. 8 no. 1, pp 101 –113, 2009.

[2] H. Vorobieva, "Automatic parallel parking in tiny spots: path planning and control," IEEE Tr. on Intelligent Transportation Systems, vol. 16 no. 1, pp. 396 –410, 2014.

[3] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," Pacific journal of mathematics, vol. 145 no. 2, pp. 367 –393, 1990.

[4] A. Scheuer and T. Fraichard, “Collision-free and continuous-curvature path planning for car-like robots,” In Robotics and Automation, Proceedings of IEEE ICRA, pp. 867–873, 1997.

[5] B. Muller, J. Deutscher and S. Grodde, "Continuous curvature trajectory design and feedforward control for parking a car," IEEE Tr. on control systems technology, vol. 15 no. 3, pp. 541–553, 2007.

[6] I. Bae, J. Moon, H. Park, J. Kim, and S. Kim, "Path generation and tracking based on a Bezier curve for a steering rate controller of autonomous vehicles," In International IEEE Conf. on Intelligent Transportation Systems, pp 6–9, Oct., 2013.

[7] J. Moon, I. Bae, J. Cha, and S. Kim, "A trajectory planning method based on forward path generation and backward tracking algorithm for automatic parking systems." In International IEEE Conf. on Intelligent Transportation Systems, pp. 8–11, Oct., 2014.

[8] F. Gómez-Bravo, F. Cuesta, A. Ollero, and A. Viguria, "Continuous curvature path generation based on β-spline curves for parking manoeuvres," Robotics and autonomous systems, vol. 56 no. 4, pp. 360–372, 2008.

[9] C. Lee, C. Lin and B. Shiu, "Autonomous vehicle parking using hybrid artificial intelligent approach," Journal of Intelligent and Robotic Systems, vol. 56, no. 3, pp. 319-343, 2009.

[10] S. Choi, C. Boussard and B. d'Andréa-Novel, "Easy path planning and robust control for automatic parallel parking," IFAC Proceedings, vol. 44, no. 1, pp. 656-661, 2011.

[11] F. Gómez-Bravo, F. Cuesta and A. Ollero, "Parallel and diagonal parking in nonholonomic autonomous vehicles," Engineering applications of artificial intelligence, vol. 14 no. 4, pp. 419–434, 2001.

[12] Y. Zhao and E. G. Collins, "Robust automatic parallel parking in tight spaces via fuzzy logic," Robotics and Autonomous Systems, vol. 51 no. 2, pp. 111–127, 2005.

[13] B. Li, K. Wang and Z. Shao, "Time-Optimal Maneuver Planning in Automatic Parallel Parking Using a Simultaneous Dynamic Optimization Approach," IEEE Tr. on Intelligent Transportation Systems, vol. 17 no. 11, pp. 3263–3274, 2016.

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

[15] K. Kondak and G. Hommel, “Computation of Time Optimal Movements for Autonomous Parking of Non-Holonomic Mobile Platforms”, Proceeding of IEEE ICRA, pp. 2698-2703, 2001

[16] A. Wächter and LT. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," Mathematical programming, vol. 106, no. 1, pp. 25–57, 2006.

[17] R. Fourer, D. Gay and B. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. San Francisco: Scientific Press, 1993.