석 사 학 위 논 문
Master's Thesis

단순화된 동역학 모델을 사용하는 보행로봇을 위한 충돌 역전파를 활용한 장애물 회피 방법

Collision-Backpropagation based Obstacle Avoidance Method for a Legged Robot Expressed as a Simplified Dynamics Model

2023

김 진 원 (金 鎭 源 Kim, JinWon)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

단순화된 동역학 모델을 사용하는 보행로봇을 위한 충돌 역전파를 활용한 장애물 회피 방법

2023

김 진 원

한 국 과 학 기 술 원

단순화된 동역학 모델을 사용하는 보행로봇을 위한 충돌 역전파를 활용한 장애물 회피 방법

김 진 원

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2022년 12월 16일

심사위원장 윤 성 의 (인)

심 사 위 원 박 대 형 (인)

심 사 위 원 박 해 원 (인)

Collision-Backpropagation based Obstacle Avoidance Method for a Legged Robot Expressed as a Simplified Dynamics Model

JinWon Kim

Advisor: Sung-Eui Yoon

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in

Daejeon, Korea
December 16, 2022

Approved by

Sung-Eui Yoon
Professor of School of Computing

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MRE

김진원. 단순화된 동역학 모델을 사용하는 보행로봇을 위한 충돌 역전파를 활용한 장애물 회피 방법. . 2023년. 14+iii 쪽. 지도교수: 윤성의. (영문 논문)

JinWon Kim. Collision-Backpropagation based Obstacle Avoidance Method for a Legged Robot Expressed as a Simplified Dynamics Model. . 2023. 14+iii pages. Advisor: Sung-Eui Yoon. (Text in English)

초 록

단순화된 동역학 모델은 보행 로봇이 모션을 생성할 때 계산적으로 덜 복잡하게 만드는 데 많이 사용된다. 하지만, 단순화된 동역학 모델에 특화된 충돌 회피에 대한 연구는 많지 않다. 우리는 이 문제에 기여하기 위해 충돌 역전파 기반 장애물 회피 방법 (CBOA)을 제시한다. 우리의 방법은 궤적을 최적화하고 장애물과의 충돌을 방지하기 위해 충돌 비용의 역전파를 사용한다. 실험에 따르면 CBOA는 이전의 연구에 비해 경로의 충돌 가능성을 최대 15.89배 줄일 수 있다.

핵 심 낱 말 충돌 회피, 단순화된 동역학, 경로 최적화

Abstract

Simplified dynamics models have been widely used to make motion planning for legged robots less computationally complex. On the other hand, there hasn't been much study on collision avoidance for a simplified dynamics model. We present the collision-backpropagation based obstacle avoidance method (CBOA) to contribute to this problem. Our method uses the gradient flow of the collision cost to optimize the trajectory and prevent collisions with obstacles. According to our experiment, the CBOA can reduce planned trajectory collisions by up to 15.89 times compared to a previous implicit collision avoidance technique.

Keywords Collision avoidance, Simplified dynamics, Trajectory optimization

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Introduction

A legged robot received much attention because it could traverse uneven terrain. To plan a feasible motion without jerky movements for this legged robot, it is crucial to take into account its dynamics. However, planning is time-consuming when rigid-body dynamics model or other complex dynamics model are taken into account. All of the legs' effects are taken into account in the rigid-body dynamics model, which makes it complicated [1]. Recent studies on legged robots frequently omit the effects of legs in order to simplify the dynamics models, such as the centroidal dynamics model [2–6] or the inverted pendulum dynamics model [7].

By neglecting the effects of legs, a simplified dynamics model for trajectory planning utilizes end-effectors without joints. Consequently, leg collision avoidance is frequently missed in simplified dynamics models [2]. Leg collisions in complicated situations are still a problem, despite prior efforts to prevent them by improving end-effector position [3–5].

We suggest the *collision-backpropagation based obstacle avoidance method* (CBOA) to prevent collision with obstacles during trajectory optimization as a solution to this remained issue. We aim to optimize the end-effector parameterized trajectory because many simplified dynamics models have end-effector information. We employ backpropagation from the collision to the end-effector, *i.e.*, CBOA, to provide collision information to the end-effector.

We compared our approach to two cutting-edge legged robot trajectory planners: TOWR [2] and *edge-based collision avoidance method* (EBCA) [3–5]. The EBCA is a method that deviates from direct collision avoidance by positioning the contact point of end-effector away from the edges. In four settings (Fig. 1.1), we compared the proportion of correctly generated trajectories to those without collisions. As a consequence, our approach produces at best 15.89 times more collision-free trajectories than the EBCA.

The main contribution of this paper is to focus on a simplified dynamics model and to take into account leg and end-effector collisions.

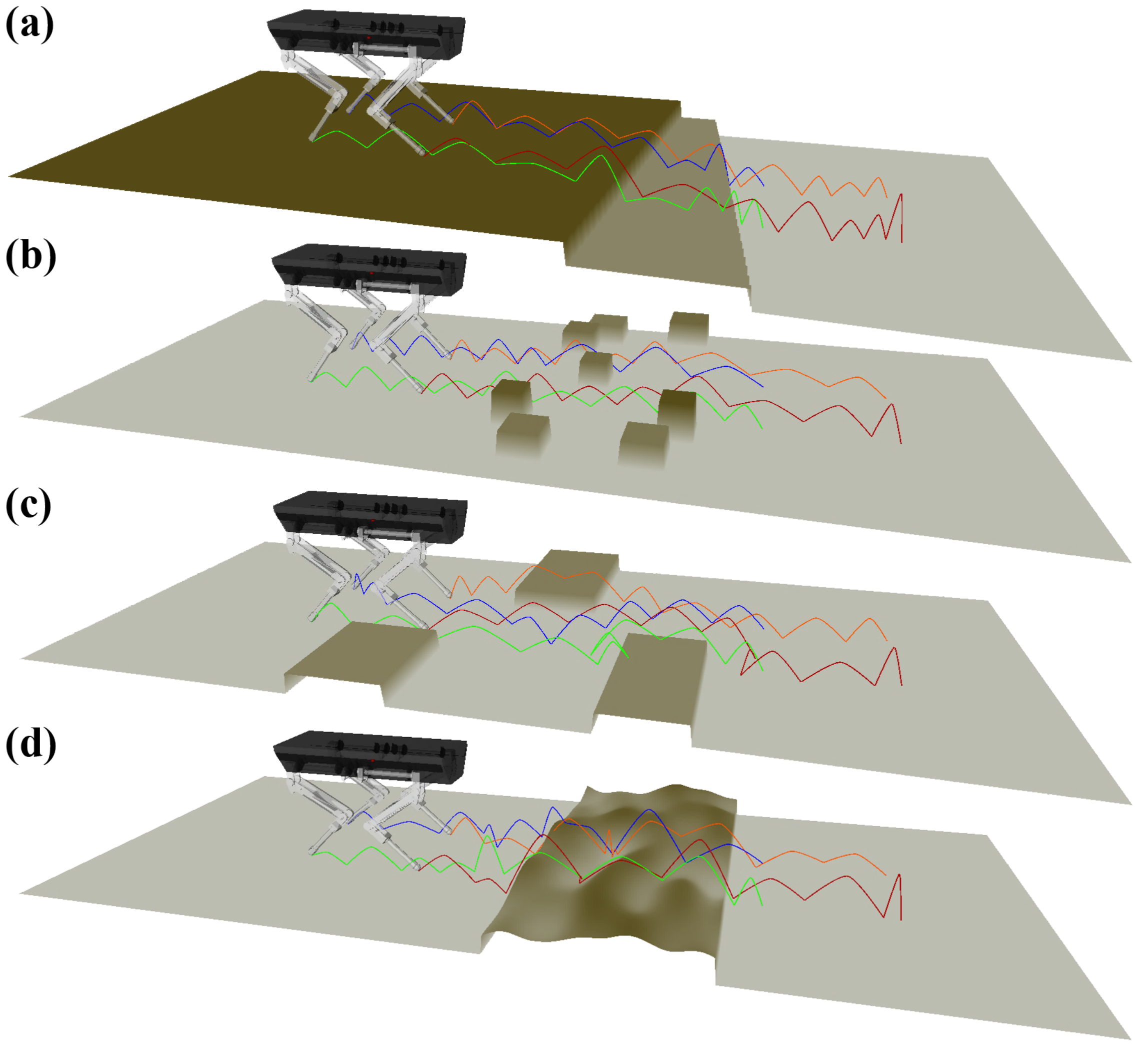


Figure 1.1 The proposed method produces trajectory optimization in four different environments: (a-d) stair, huddle, block, and bumpy terrains. Our approach successfully generates collision-free trajectories with the aid of a gradient flow. The end-effector trajectory for every leg is depicted by a color line.

Chapter 2. Trajectory optimization for a legged robot utilizing a simplified dynamics model

Planning the trajectory of a legged robot frequently makes use of trajectory optimization methods [2–5]. Trajectory planning for a legged robot needs to take constraints into account. As an example, trajectory optimization takes into account dynamics to provide feasible motion, friction cones to prevent the robot's feet from slipping, and a contact schedule to produce gait patterns [2].

However, as the number of these constraints increases, trajectory optimization becomes significantly more difficult to optimize. To reduce the complexity of trajectory optimization, prior works [2–5] simplify a whole-body dynamics model to a simplified dynamics model. The centroidal dynamics model, which disregards the effects of all legs, is one of the ways for simplifying dynamics models. As a result, dynamics are calculated without knowledge of the legs using the base and end-effectors.

We need to provide constraints for a trajectory in terms of the parameters of the simplified dynamics model, such as the position of end-effectors. For example, end-effectors should be within the kinematic range to prevent violating joint limitations. However, in the previous work [2], no collision avoidance method is parameterized with the end-effector.

The edge-based collision avoidance approach, which avoids collision using an end-effector [3–5], does not take leg collision into account. We suggest the CBOA approach to take the leg collision and end-effector collision into account.

Chapter 3. A collision-backpropagation method for a simplified dynamics model

To avoid a collision, we need to measure the distance between each leg and the nearest obstacle. We calculate the signed distance from a signed distance field, SDF (Fig 3.1b), given a point of a potential collision, drawing inspiration from earlier works [8, 9]. We use a *collision forward propagation* from the end-effector to calculate the position of the point of the potential collision. The *collision forward propagation* method calculates the collision cost by using the kinematics of the robot (in Sec 3.2). The collision cost is determined by the penalty function in the collision forward propagation step, which provides margin distance to obstacles (in Sec 3.3). The input of the penalty function is the signed distance. Then, we can utilize the *collision backpropagation* to calculate the gradient of the end-effector (in Sec 3.4). Finally, trajectory optimization can perform collision avoidance using the collision cost and gradient.

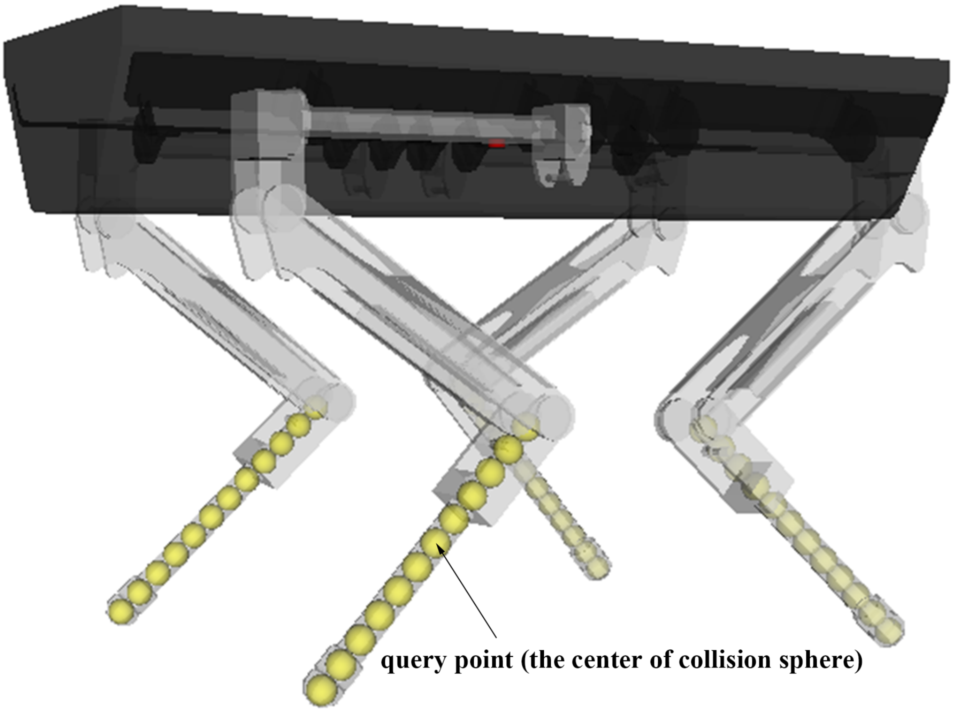
3.1 Collision checking method using a geometric approximation

We need to determine whether the obstacle penetrates the robot's leg in order to check collisions. In order to locate a probable collision point, known as the query point (Fig 3.1a), and determine the signed distance from the SDF using the query point, we approximate the robot's legs to collision spheres. The signed distance from the query point to the closest obstacle is represented by the SDF. If the query point is inside the obstacle, the signed distance has a negative value. We suppose that a collision happens if the signed distance of the query point is less than the radius of the collision sphere.

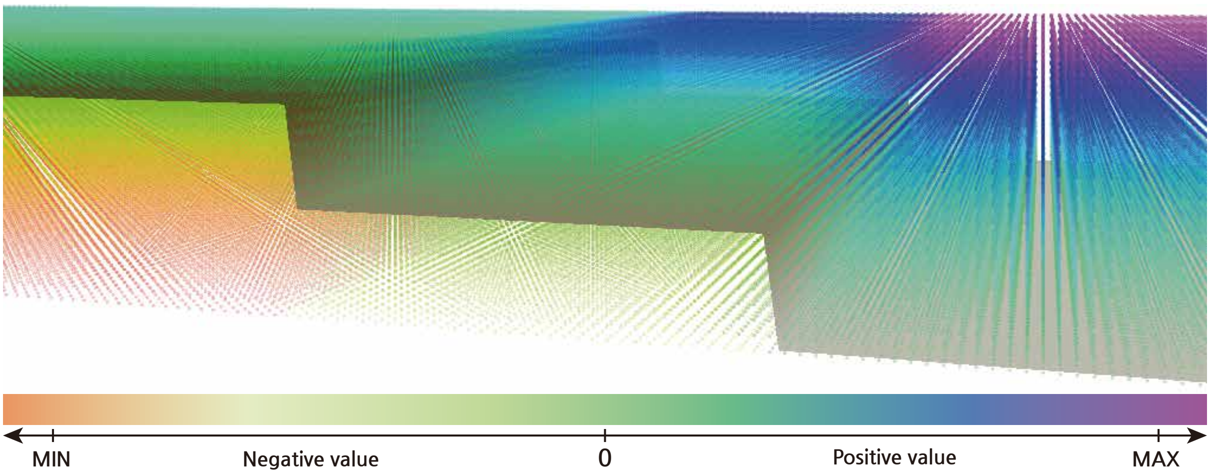
This approach of checking for collisions is a productive way to use the SDF to prevent collisions. Since the SDF is already computed prior to optimization, querying the signed distance only has an O(N) complexity, where N is the total number of collision primitives.

3.2 Collision forward propagation to collision cost

As indicated by the orange arrows in Fig. 3.2, we can use *collision forward propagation* to obtain the end-effector collision cost. As aforementioned, we make the simplified dynamics model check collisions by assuming imaginary legs. The joint values θ of the legs are computed using the inverse kinematics of the end-effector x. The relationship between the position of the end-effector and the joints should be mapped by the inverse kinematics in a one-to-one correspondence. If joints are not determined, it is hard to avoid collisions because the inverse kinematics cannot determine the legs' shape only with the end-effector position. We can suppose that a generally constructed legged robot has a unique solution [10]. After then, forward kinematics is used to determine the positions of collision spheres s.



(a) Visualization of robot and collision spheres



(b) Visualization of signed distance field

Figure 3.1 (a) An illustration of how the HyQ robot can visualize collision spheres. To check for collision, we approximate the lower leg with yellow collision spheres. The query point for collision checking defines the center of the collision sphere. (b) A signed distance field in the stair environment.

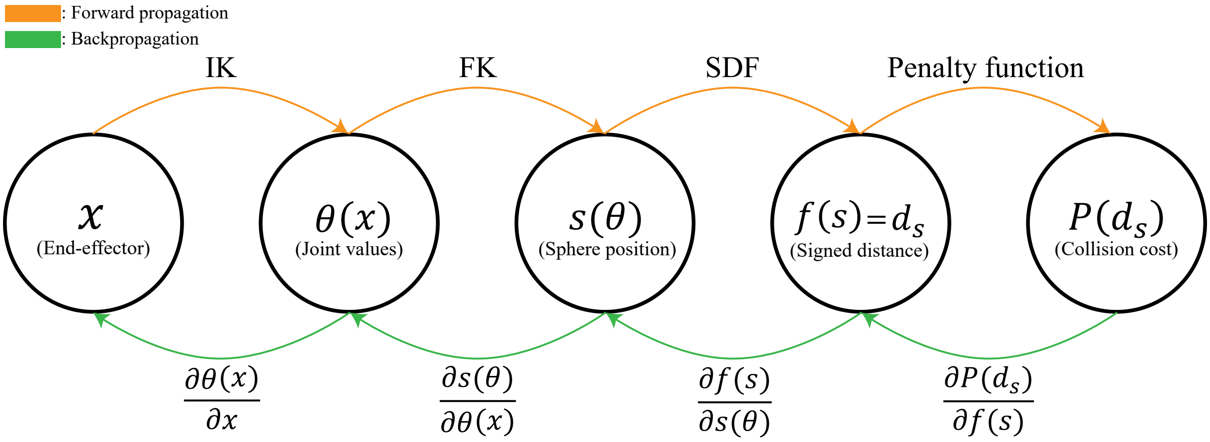


Figure 3.2 The relationship between the end-effector x and the collision cost $P(d_s)$ is depicted in the figure. It is possible to calculate the collision cost during the forward propagation phase (orange arrows). The rate of change of the cost $P(d_s)$ with respect to the end-effector x can be calculated in the backpropagation process (green arrows).

3.3 Penalty function according to phase

If the cost uses the signed distance directly, the robot avoids obstacles without any margin. As a result, we use a penalty function similar to the one [11] to provide a collision margin:

$$P(d_s) = \begin{cases} s/3\,(d_m - d_s)^3, & d_s < d_m \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

where s is the constant that denotes the effects the signed distance has on the cost, d_m is the margin distance, and d_s is the signed distance.

According to the swing and stance phases, the penalty function $P(d_s)$ is computed. End-effectors should be put on the ground during the stance phase. As a result, the stance phase's z-axis must be zero. On the other hand, d_m does not have to be zero throughout the swing phase. In the stance phase, we define $d_m = 0.01m$ with the exception of the z-axis, and $d_m = 0.03m$ in the swing phase.

3.4 Collision backpropagation to end-effector

Using *collision-backpropagation based obstacle avoidance method* (CBOA), we can calculate the rate of change of the cost with respect to the end-effector after computing the cost using collision forward propagation. As shown in the green arrows in Fig. 3.2, we use CBOA to compute the gradient for the end-effector as follows:

$$\frac{\partial P(x)}{\partial x} = \left(\frac{\partial P(f)}{\partial f(s)}\right)\left(\frac{\partial f(s)}{\partial s(\theta)}\right)\left(\frac{\partial s(\theta)}{\partial \theta(x)}\right)\left(\frac{\partial \theta(x)}{\partial x}\right) \tag{3.2}$$

$$= \left(\frac{\partial P(f)}{\partial f(s)}\right) f'(\mathrm{s}) J_s J_x^{-1} \tag{3.3}$$

$$= \left(\frac{\partial P(f)}{\partial f(s)}\right) f'(\mathrm{s}) J_s J_x^{\dagger}. \tag{3.4}$$

If the collision cost is backpropagated, it will eventually become Eq. (3.2). Each gradient can be obtained using well-known robotics definitions. We can calculate the gradient for $\frac{\partial f(s)}{\partial s(\theta)}$ by using the difference the signed distances between adjacent points. The Jacobian J_s definition, which provides the relationship between joint and sphere velocities, is $\frac{\partial s(\theta)}{\partial \theta(x)}$. The Jacobian inverse Jx^{-1}, which provides the relationship between joint and end-effector velocities, is also known as $\frac{\partial \theta(x)}{\partial x}$. We can switch from Eq. (3.2) to Eq. (3.3) using the definitions provided above. With a singular configuration, the Jacobian inverse is incorrect. Furthermore, rather than being applied globally, the Jacobian inverse solution works well locally. In order to solve these issues, we convert the Jacobian inverse J_x^{-1} into the Levenberg-Marquardt method $J_x^{\dagger}$ [12] (Eq. (3.4)). Levenberg-Marquardt method $J_x^{\dagger}$ is $J^T(JJ^T + \lambda^2 I)^{-1}\vec{e}$. λ is a non-zero damping constant, and $\vec{e}$ is a desired change in the position of an end-effector. This approach yields a numerically stable condition and is robust close to a singularity configuration. We can make collision-free trajectories using trajectory optimization with our method.

We can use the CBOA in the stance and swing phases, as seen in Section 3.3. The $\mathrm{CBOA_t}$ in the stance phase, the $\mathrm{CBOA_w}$ in the swing phase, and the $\mathrm{CBOA_{tw}}$ in the stance and stance phases are used as a typescript to denote the different phases.

Chapter 4. Results

4.1 Experimental setting

We conducted experiments in the four environments depicted in Fig. 1.1. In each setting, we produced a random obstacle shape and assigned a target position. A legged robot used in the experiments was a HyQ [13]. We compared our method to TOWR [2], a trajectory optimization method of a legged robot with simplified dynamics, and the *edge-based collision avoidance method* [3–5] (EBCA).

TOWR is a trajectory optimization method for a legged robot with a simplified dynamics model that does not consider a collision. EBCA is a method that includes a collision cost, which implicitly inhibits a trajectory from colliding with obstacles during the stance phase. The collision cost is a differential Gaussian function defined as $\sum e^{-d^2/2\sigma^2}$. d represents the distance between the contact point and the nearest edge, and σ is the standard deviation. It is difficult to directly avoid collisions with the EBCA using collision checking. In addition, this method is inapplicable to the swing phase because the EBCA estimates the distance between the contact point and the nearest edge. For the sake of a fair comparison, we employ EBCA and $\mathrm{CBOA_w}$.

4.2 Experiment results

The quantitative results are shown in Table 4.1. The *Collision-free trajectory ratio* indicates the proportion of collision-free trajectories relative to the total number of planned trajectories. The *total collision ratio* indicates the proportion of actual collisions relative to the total number of collision checking. The *optimization time* is the mean optimization time of trajectory planning.

According to the *collision-free trajectory ratio column*, $\mathrm{CBOA_{tw}}$ approaches outperform other methods. In particular, our methods generate 15.89 times more collision-free trajectories than the EBCA in the huddle environments. Furthermore, applying the stance cost such as the EBCA in conjunction with the $\mathrm{CBOA_w}$ can improve collision avoidance performance.

As seen in the *total collision ratio* column, the $\mathrm{CBOA_{tw}}$ methods have the least number of collisions with obstacles. Even when using only the $\mathrm{CBOA_t}$ approach, the $\mathrm{CBOA_t}$ method outperforms the EBCA by a maximum of 20.75 times.

Although our methods take more time than other methods, the performance significantly improves thanks to our method's precise collision avoidance ability. The $\mathrm{CBOA_t}$ method has a lower optimization time than the EBCA and has reasonable collision avoidance ability. The $\mathrm{CBOA_t}$ or $\mathrm{CBOA_{tw}}$ method can be selected depending on whether collision avoidance quality or optimization time is important.

Even while our approaches take longer than other ways, the obstacle avoidance performance is much better because of our precise collision avoidance. The $\mathrm{CBOA_t}$ approach optimizes in less time than the EBCA and has adequate collision avoidance capabilities. $\mathrm{CBOA_t}$ or $\mathrm{CBOA_{tw}}$ can be chosen based on whether collision avoidance quality or optimization speed is more significant.

Table 4.1 Trajectory optimization methods were compared for four types of terrains. The *edge-based collision avoidance method* (EBCA) in bumpy terrain is not performed since bumpy terrain has no edges. The collison-backpropagation based obstacle avoidance (CBOA) is our method. The $\mathrm{CBOA_t}$ refers to the stance phase cost, the $\mathrm{CBOA_w}$ refers to the swing phase cost, and the $\mathrm{CBOA_{tw}}$ refers to the stance and swing phases. We measure the ratio of collision-free trajectories among total trajectories, *collision-free trajectory ratio*, and the ratio of actual collisions over the total collision checkings, *total collision ratio*. The $\mathrm{CBOA_{tw}}$ outperforms The EBCA up to 15.89 times in planning the *collision-free trajectory*.

Terrain	Method	Collision-free trajectory ratio (%) ↑	Total collision ratio (%) ↓	Optimization time (s) ↓
Stair	TOWR	13.04	40.02	2.03
	EBCA	13.01	5.35	6.87
	$^*\mathrm{CBOA_t}$	15.94	0.62	5.06
	EBCA + $^*\mathrm{CBOA_w}$	45.95	4.52	8.84
	$^*\mathrm{CBOA_{tw}}$ (ours)	**88.06**	**0.0**	10.27
Block	TOWR	26.47	49.16	1.83
	EBCA	21.88	41.58	3.43
	$^*\mathrm{CBOA_t}$	47.06	2.94	3.68
	EBCA + $^*\mathrm{CBOA_w}$	40.32	29.9	4.45
	$^*\mathrm{CBOA_{tw}}$ (ours)	**85.29**	**0.8**	6.05
Huddle	TOWR	2.63	51.55	1.85
	EBCA	5.63	32.58	4.8
	$^*\mathrm{CBOA_t}$	21.05	1.57	4.65
	EBCA + $^*\mathrm{CBOA_w}$	21.21	26.24	8.65
	$^*\mathrm{CBOA_{tw}}$ (ours)	**89.47**	**0.0**	11.96
Bumpy	TOWR	5.17	15.39	2.13
	$^*\mathrm{CBOA_t}$	8.52	4.1	26.35
	$^*\mathrm{CBOA_{tw}}$ (ours)	**57.98**	**2.33**	34.8

* *our method*

4.3 Optimization success rate

The trajectories that were successfully optimized were used to calculate the experimental results in Section 4.2. However, complexity issues, the non-convex nature of constraints, or costs can sometimes cause optimization to fail. The *optimization success ratio* is shown in Table 4.2. Optimization success does not imply a collision-free trajectory, but rather a trajectory that satisfies constraints whether or not the collision cost is satisfied. Our methods have a higher *optimization success ratio* than those of the EBCA. However, our method's optimization success rates in a bumpy environment are poor. This may be due to the fact that it is more challenging to avoid collisions in a rough environment. Notwithstanding, our methods can generate trajectories in environments where EBCA cannot be applied.

Table 4.2 The table shows the optimization success ratio when adding the collision cost term. The optimization success ratio indicates the optimizer finds the solution within the given time. Our approach outperforms the *edge-based collision avoidance method* (EBCA).

Terrain	**Method**	**Optimization success ratio (%)** $\uparrow$
Stair	EBCA	89.78
	$\mathrm{CBOA}_{\mathrm{tw}}$ (ours)	**97.1**
Block	EBCA	95.52
	$\mathrm{CBOA}_{\mathrm{tw}}$ (ours)	**100**
Huddle	EBCA	95.95
	$\mathrm{CBOA}_{\mathrm{tw}}$ (ours)	**100**
Bumpy	EBCA	-
	$\mathrm{CBOA}_{\mathrm{tw}}$ (ours)	**50.22**

Chapter 5. Conclusion

We present a backpropagation-based collision avoidance method for a legged robot expressed as a simplified dynamics model. Our approach is intended for end-effector parameterized trajectory optimization, such as a method using *centroidal dynamics model.* In comparison to earlier works, our method was up to 15.89 times more likely to produce a collision-free trajectory. Nevertheless, our method takes longer to generate a trajectory because of the collision cost computation. Future work can use a learning-based method to shorten computation times.

The content written in this dissertation has been submitted to the ICCAS conference in 2022 and is accpeted.

Bibliography

[1] A. W. Winkler, "Optimization-based motion planning for legged robots," Ph.D. dissertation, ETH Zurich, 2018.

[2] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[3] O. Melon, M. Geisert, D. Surovik, I. Havoutis, and M. Fallon, "Reliable trajectories for dynamic quadrupeds using analytical costs and learned initializations," in *International Conference on Robotics and Automation.* IEEE, 2020, pp. 1410–1416.

[4] D. Surovik, O. Melon, M. Geisert, M. Fallon, and I. Havoutis, "Learning an expert skill-space for replanning dynamic quadruped locomotion over obstacles," 2021.

[5] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, "Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization," in *International Conference on Robotics and Automation.* IEEE, 2021, pp. 9805–9811.

[6] T. Kwon, Y. Lee, and M. Van De Panne, "Fast and flexible multilegged locomotion using learned centroidal dynamics," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 46–1, 2020.

[7] X. Li, J. Gao, Q. Huang, H. Lu, Z. Xu, and Y. Liu, "A new control method of quadruped robot walking on rough terrain based on linear inverted pendulum method," in *International Conference on Robotics and Biomimetics.* IEEE, 2014, pp. 2410–2415.

[8] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *International Conference on Robotics and Automation.* IEEE, 2009, pp. 489–494.

[9] R. Buchanan, L. Wellhausen, M. Bjelonic, T. Bandyopadhyay, N. Kottege, and M. Hutter, "Perceptive whole-body planning for multilegged robots in confined spaces," *Journal of Field Robotics*, vol. 38, no. 1, pp. 68–84, 2021.

[10] M. A. Sen, V. Bakircioglu, and M. Kalyoncu, "Inverse kinematic analysis of a quadruped robot," *International journal of scientific & technology research*, vol. 6, no. 9, pp. 285–289, 2017.

[11] M. Schwienbacher, T. Buschmann, S. Lohmeier, V. Favot, and H. Ulbrich, "Self-collision avoidance and angular momentum compensation for a biped humanoid robot," in *International Conference on Robotics and Automation.* IEEE, 2011, pp. 581–586.

[12] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

[13] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq– a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

Acknowledgments in Korean

로봇을 조금 더 연구해보고 싶다는 생각으로 입학한 석사과정이 어느덧 졸업을 앞두고 있습니다. 삶은 선택의 연속이고 운칠기삼이라고 하는데 좋은 교수님과 선배, 동료, 그리고 후배들 밑에서 석사과정을 하여 후회되지 않을 최고의 선택이었다고 생각이 됩니다. 특히나 항상 학생의 관점에서 생각해주시고 지도해주시는 윤성의 교수님 덕분에 인격적인 성장뿐 아니라, 많은 배움을 얻었습니다. 정말 감사드립니다.

연구실 생활은 결국 스스로와의 싸움인 것 같습니다. 연구가 잘되는 듯싶다가도 어느 순간 잘못된 방향으로 빠지게 되는 일이 부지기수였습니다. 이에 슬럼프도 왔었지만 좋은 연구실 분위기와 동료 선배와의 소통으로 어려움을 극복할 수 있었습니다.

저의 연구실 생활은 너무나도 행복했었습니다. 사수로 만난 희찬이형은 항상 후배를 배려하고 지도해주며, 자신의 연구를 꾸준히 수행하셨습니다. 특히나 목표를 설정하면 어떻게든 달성해내는 모습에서 후배로서 존경스러웠으며, 많은 지도로 연구를 할 수 있었습니다. 또한 운이 좋게도, 입학 당시 동갑내기 친구들인 형렬이, 세빈이, 그리고 민성이가 있었기에 연구실 적응을 빠르게 할 수 있었고 즐거운 추억을 많이 만들 수 있었습니다. 동갑의 친구들이었지만, 누구보다 배울 점이 많았던 친구들이었고 덕분에 항상 즐거웠습니다.

로봇팀의 강민철형은 감사하게도 저를 아껴주셔서 즐거운 연구실 생활을 할 수 있었고 함께 갔던 출장 및 워크숍에서의 추억들을 잊지 못할 것 같습니다. 항상 복도에서 마주치면 웃으며 인사해주시고 안부를 물어봐 주시던 인규형에게도 매우 감사합니다. 졸업을 이미 하신 용선이형에게도 짧은 시간이나마 함께 연구실 생활을 하여 즐거웠습니다. 맨 처음 연구실 생활에 아직 적응하지 못했을 때 어랑족에서 밥을 사주셨을 때가 아직도 가끔 생각이 납니다. 제 옆자리인 태영이형은 항상 규칙적인 생활을 하시며 꾸준히 연구하시는 모습이 모범이 되셨으며 부담스럽지 않게 다가가 이야기를 나눌 수 있는 좋은 선배이셨습니다. 김민철형은 같이 연구실에 입학하여 스터디 및 팀 프로젝트를 함께 진행하며 많은 것을 배울 수 있는 좋은 연구실 동기이자 인생선배였습니다. 또한 자주 뵙지는 못했지만 미팅할 때마다 뵙는 문성주 연구원님과 많은 인생 상담을 해주신 장충수 형님께도 감사드립니다.

또한 비전팀의 임우빈, 김재윤, 정준식, Guoyuan, 조윤기, 그리고 김우재 모두 즐거운 연구실 생활을 지낼 수 있도록 해줘서 매우 고마웠고 렌더링팀의 김재윤, 그리고 한규범에게도 감사의 인사를 드립니다. 후배인 이주민, 선주형, 손우정, 그리고 Yaxin Wang 모두 각자의 연구를 열심히 잘하는 모습이 너무 멋있고 응원합니다.

석사를 진학하고 졸업할 때까지 항상 응원과 지지를 해주신 우리 가족에게도 감사인사를 남깁니다. 또한 바쁘기도 하고 떨어져 있어 자주 만나지 못했을 텐데도 항상 응원해준 여자친구에게도 너무나도 고맙습니다. 졸업하고 사회에 나가는 것이 조금은 두렵기도 하지만, 새로운 삶 속으로 한 발자국을 내딛어보도록 하겠습니다. 그동안 감사했습니다.

Curriculum Vitae in Korean

이 름: 김 진 원
생 년 월 일: 1995년 11월 03일
주 소: 대전시 유성구 대학로 291 한국과학기술원 전산학부(E3-1) 3443호
전 자 주 소: mqjinwon@gmail.com

학 력

2011. 3. – 2014. 2. 대일고등학교

2015. 3. – 2021. 2. 광운대학교 (학사)

2021. 3. – 2023. 2. 한국과학기술원 (석사)

경 력

2019. 6. – 2019. 12. 한국과학기술연구원 지능로봇연구단 학생연구원

연 구 업 적

1. **J. Kim** and K. Kim, "Robust multi object detection using siamese network," in Korea robotics society annual conference. KROS, 2020.
2. **J. Kim**, H. Shin, and S.-E. Yoon, "Collision backpropagation-based obstacle avoidance method for a legged robot with simplified dynamics model," in Korea robotics society annual conference. KROS, 2022.
3. **J. Kim**, H. Shin, and S.-E. Yoon, "Collision-backpropagation based obstacle avoidance method for a legged robot expressed as a simplified dynamics mode," in International Conference on Control, Automation and Systems. IEEE, 2022.