

Real-time Perceptive Motion Control using Control Barrier Functions with Analytical Smoothing for Six-Wheeled-Telescopic-Legged Robot Tachyon 3

Noriaki Takasugi¹, Masaya Kinoshita¹, Yasuhisa Kamikawa¹, Ryoichi Tsuzaki¹, Atsushi Sakamoto¹, Toshimitsu Kai¹, and Yasunori Kawanami¹

Abstract—To achieve safe legged locomotion, it is crucial to generate motion in real-time considering various constraints in robots and environments. In this study, we propose a lightweight real-time perceptive motion control system for the newly developed six-wheeled-telescopic-legged robot, Tachyon 3. In the proposed method, analytically smoothed constraints including Smooth Separating Axis Theorem (SSAT) as a novel higher order differentiable collision detection for 3D shapes is applied to the Control Barrier Function (CBF). The proposed system integrating the CBF achieves online motion generation in a short control cycle of 1 ms that satisfies joint limitations, environmental collision avoidance and safe convex foothold constraints. The efficiency of SSAT is shown from the collision detection time of 1 μ s or less and the CBF constraint computation time for Tachyon 3 of several μ s. Furthermore, the effectiveness of the proposed system is verified through the stair-climbing motion, integrating online recognition in a simulation and a real machine.

I. INTRODUCTION

In recent years, legged robots have attracted attention from the viewpoint of rough terrain movement performance, and their practical application has advanced. On the other hand, the fall risk remains a challenge for legged robots with a high center of gravity. To mitigate the fall risk of legged robots, Tachyon 3, a novel six-wheeled-telescopic-legged robot, was developed as shown in Fig. 1 [1]. Tachyon 3 improves stability and energy efficiency through static walking with six legs and wheel movement and lowers the center of gravity with a telescopic leg. However, due to low center of gravity movement, the constraint of avoiding collisions with the environment and the range of motion of the legs becomes severe. Therefore, it is crucial to generate motion under these constraints.

Model Predictive Control (MPC) [2] and Reinforcement Learning (RL) [3] have been actively studied toward motion generation that takes environmental and physical constraints into account. However, due to its high computational complexity, MPC often requires additional high-frequency control for strictly satisfying the constraints as in [2]. RL struggles to explicitly handle constraints, making it challenging to learn motion under severe constraints. Control Barrier Function (CBF) [4] is a more computationally efficient method for considering the constraints, and has been used for real-time collision avoidance between robots and environments [5], [6], [7]. However, collision avoidance between

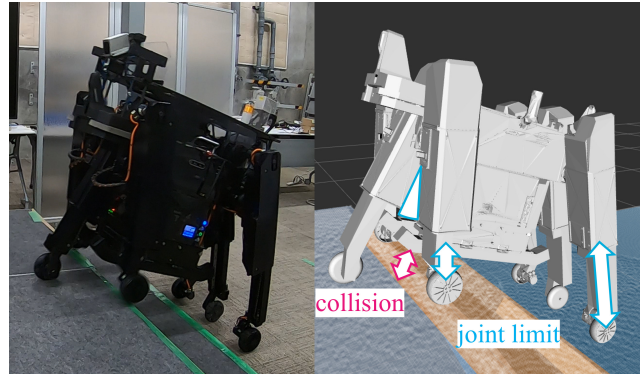


Fig. 1. Six-wheeled-telescopic-legged robot Tachyon 3 which has 6 legs and 4 actuated wheels and 2 omni passive wheels.

convex polyhedron such as cuboid in continuous high-order CBF [8], [9] has not yet been addressed because the high-order derivatives of non-differentiable collision detection are complicated to obtain. Therefore, high-frequency real-time control that takes into consideration collision detection for cuboids has not been achieved.

In this paper, we propose a novel lightweight perceptive real-time control framework for Tachyon 3. Our control methodology operates at a high frequency of 1 kHz, enabling the precise generation of smooth target joint positions for Tachyon 3's actuators, which utilize high-gain position control. We employ the Exponential Control Barrier Function (ECBF) [8] to continuously enforce geometric constraints, including the range of motion and environmental collision avoidance. To obtain higher-order derivatives of the collision avoidance for ECBF, we introduce a novel formulation which effectively smoothes the Separating Axis Theorem (SAT) [10]. This formulation guarantees collision avoidance between the cuboid body of Tachyon 3 and the surrounding environment. To validate the effectiveness of our proposed method, we conduct practical experiments using both simulation and a real machine. These experiments involve perceptive locomotion in unknown environments, demonstrating the safety and computational efficiency of our approach.

A. Related Works

Motion generation techniques for legged robots that address environmental constraints can be categorized into two main approaches: model-free and model-based. Model-free motion generation, for instance, has been advanced through perceptive reinforcement learning for both legged robots [11]

¹Sony Group Corporation, Minato-ku, Tokyo, Japan, 108-0075
noriaki.takasugi@sony.com

and wheeled-legged robots [3]. While these methods can generate robust end-to-end motion by utilizing body-relative point clouds for environmental perception, tuning the reward function to learn specific desired motions is challenging, and the assurance of safety is not guaranteed.

Model-based-control methods, specifically MPC, has also been applied to motion generation of wheeled-legged robots [2]. In this prior research, time varying CBFs are employed to satisfy the 2D convex foothold constraints in MPC. There are also studies on legged robots to generate motion by perceptive MPC [12]. In [12], a Signed Distance Field (SDF) is constructed from a 3D voxel grid map to avoid collision of swing legs. However, it is limited to collision avoidance between an environment and a point or spherical object, that is, collision avoidance between cuboid are not considered in contrast to this study. Moreover, when using MPC in practice, additional high-frequency low level control is required to continuously satisfy severe constraints in legged locomotion as demonstrated in [2], [12].

CBF is a fast control method to guarantee safety, such as collision avoidance with the environment. The CBF-based method needs to differentiate the constraints until the control input is explicitly included in the derivative. Collision avoidance constraints between spherical or circular shapes are analytically differentiable by nature, which facilitates their integration with CBFs at any relative order [13]. A collision avoidance between a point and a generic environment is typically addressed by using a SDF, with gradients often calculated through finite differences, as often used in MPC. On the other hand, collision detection between cuboids, as examined in this study, is not directly applicable to CBF due to its non-differentiable nature. For collision avoidance between convex shapes, recent advancements include differentiable optimization-based methods [14] and Gilbert Johnson Keerthi distance algorithm (GJK)'s randomization [15]. The work in [7] demonstrates an arm robot achieving collision avoidance through velocity control using CBF with differentiable collision detection. Yet, these techniques are limited to deriving first-order derivatives and are difficult to apply to ECBF which is dealt with in this study. Examples of the direct application of the non-differentiable distance calculation between convex shapes to CBF include a study [16] using a Non-smooth CBF [17] and a study [6] of obstacle avoidance for a legged robot utilizing MPC with discrete-time CBF [18]. Nonetheless, these approaches suffer from increased computational complexity due to the incorporation of dual variables in the optimization process.

As outlined previously, there is a lack of research on the application of continuous high-order CBFs [8], [9] for collision detection between 3D shapes. Consequently, due to computational efficiency, its application to real-time control (e.g., up to 1 kHz) remains unfeasible, and no existing studies have successfully implemented this approach on a real robot with online 3D perception. In [6], although 3D environmental recognition is employed, it is adapted for 2D CBF to circumvent computational complexity. The study [7] does not incorporate online obstacle recognition.

Furthermore, both [6] and [7] diverge from our work as they do not utilize the second derivative of collision detection.

B. Contribution

We introduce a novel perceptive real-time control system for the Tachyon 3. Our methodology encompasses several key innovations. Initially, we present a technique for the smoothing of constraints by substituting non-smooth operations with their analytical smooth operations. Specifically, we introduce a novel collision avoidance CBF by refining the SAT [10] through analytical smoothing. This analytical smoothing CBF is versatile, applicable to systems of any relative order, and offers rapid computational performance. The efficacy of the Smooth SAT (SSAT) is substantiated through comparative analysis with alternative collision detection algorithms. Subsequently, we implement a streamlined perceptive real-time control utilizing the SSAT CBF for Tachyon 3. Furthermore, we propose a ground-moving origin-relative model, which simplifies positional control for real-time motion planning. The proposed system is capable of traversing obstacles and stairs. The effectiveness of the proposed method is verified through simulation and real machine integrated online recognition.

C. Outline

This paper is structured as follows. In Section II, we explain Tachyon 3's hardware and propose its efficient whole-body kinematic model. In Section III, the perceptive real-time control for Tachyon 3 using ECBF is described. In Section IV, we propose analytically smoothed CBF constraints for Tachyon 3 including SSAT. In Section V, we show the effectiveness of our method. Section VI concludes the paper.

II. SIX-WHEELED-TELESCOPIC-LEGGED ROBOT TACHYON 3

A. Hardware Structure

We developed Tachyon 3, a six-wheeled-telescopic-legged robot with 16 driving and 2 passive joints [1], improving upon the conventional quadruped Tachyon [19]. As shown in Fig. 2, Tachyon 3 features 4 telescopic legs with driving wheels and 2 with omnidirectional passive wheels. Each leg has 2 degrees of freedom: a hip joint with a 45-degree range of motion (5 degrees inward and 45 degrees outward for the middle legs) and a 500 mm expandable prismatic knee joint. The design of Tachyon 3 addresses technical issues of the six-legged robot [20], reducing leg-stair interference and the robot's footprint. Its low center of gravity enhances stability and fall protection. The robot is equipped with IMU, LiDAR, and 3-axis force sensors on the toes.

The six-wheeled-legged configuration allows stable stair navigation and faster climbing by leveraging wheel movement and maintaining a secure support polygon through the alternating replacement of three support legs. On flat surfaces, Tachyon 3 can achieve free rotation by standing on three legs: the left and right driving wheel legs and the center omnidirectional passive leg without changing steps

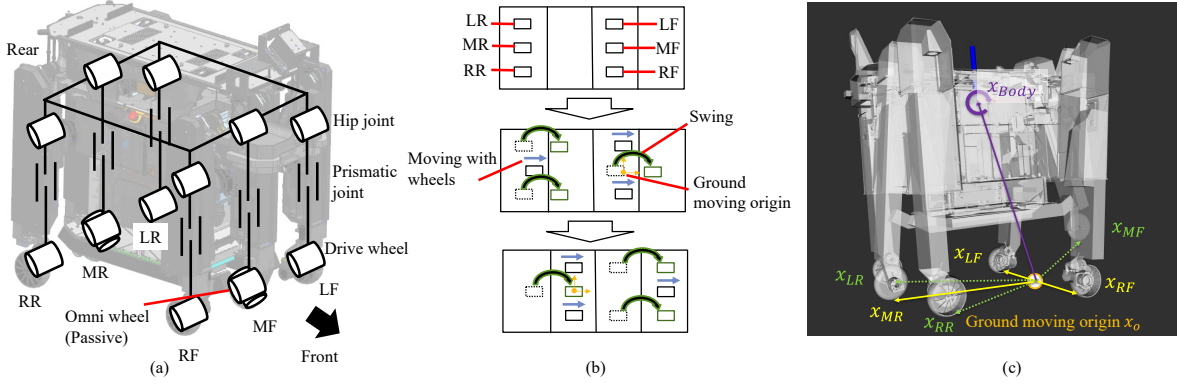


Fig. 2. (a) Tachyon 3 is equipped with three front legs (left-front (LF), middle-front (MF), and right-front (RF)) and three rear legs (left-rear (LR), middle-rear (MR) and right-rear (RR)). In total, it consists of 16 active joints including 6 prismatic joints, 6 hip joints and 4 drive wheels, and additional two passive omni wheels. (b) An example of static locomotion for climbing stairs. First, the three wheels, LR, RR and MF, swing to the next steps and the others move with the wheels. Second, MR, LF and RF next steps and the three other wheels move forward. Finally, each set of three wheels is located on the next steps. (c) The kinematic model for CBF consists of the ground moving origin using unicycle dynamics and origin-relative body and end-effector's wheel state.

like wheeled-ANYmal [3] or steering the drive axis of a wheel.

However, controlling Tachyon 3 on steps with a low center of gravity poses challenges in real-time motion generation while avoiding collisions. We address this using ECBF with a simple wheeled-legged model.

B. Ground Moving Origin Local Model

Tachyon 3 cannot independently adjust the steering angle of each wheel, which inherently limits the robot's center of rotation to the axle line of either the front or rear drive wheels. To overcome this limitation without creating a complex model, we introduce the concept of a ground moving origin. This concept simplifies the movement of multiple wheels by treating them as a single wheel that shifts its position discretely when the support legs are exchanged. To reduce the impact forces during landing and to decrease lateral sliding during the exchange of support legs, our model includes an acceleration input that smoothly brings the foot's vertical velocity and the body's yaw rate to zero. The robot's state is defined by the relative coordinates of the foot and body with respect to the ground moving origin, which is determined by the wheels in contact with the ground. We assume that the left and right drive wheels move together horizontally to reduce sliding during turns.

The model of the ground moving origin using unicycle dynamics is expressed as:

$$\frac{d}{dt} \begin{bmatrix} p \\ \psi \\ \bar{v} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} R_z(\psi) \begin{bmatrix} \bar{v} \\ 0 \\ 0 \end{bmatrix} \\ \dot{\psi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{a} \\ \dot{\psi} \end{bmatrix} \quad (1)$$

where $p \in \mathbb{R}^3$ is the position of the wheel in the world coordinate system. $\bar{v} \in \mathbb{R}$, $\bar{a} \in \mathbb{R}$ are the velocity and acceleration relative to the wheel, respectively. $\psi \in \mathbb{R}$ is the

rotation angle of the wheel. Additionally, the wheel position and body posture relative to the origin is expressed as:

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \ddot{q} \quad (2)$$

where q, \dot{q}, \ddot{q} represent origin-relative value, velocity, and acceleration, respectively.

Based on equations (1) and (2), the whole-body kinematic model of Tachyon 3 can be described as shown in Fig. 2 (d):

$$\frac{dx}{dt} = f(x) + g(x)u \quad (3)$$

$$x = [x_o \quad x_{body} \quad x_{ee_1} \quad \cdots \quad x_{ee_6}]^T$$

$$u = [u_o \quad u_{body} \quad u_{ee_1} \quad \cdots \quad u_{ee_6}]^T$$

$$f = [f_o \quad f_{body} \quad f_{ee_1} \quad \cdots \quad f_{ee_6}]^T$$

$$g = [g_o \quad g_{body} \quad g_{ee_1} \quad \cdots \quad g_{ee_6}]^T$$

In the formulation, $x_o \in \mathbb{R}^6$ and $u_o \in \mathbb{R}^2$ represent the state and input, respectively, of the ground moving origin dynamics as defined in Equation (1). Similarly, $x_{body} \in \mathbb{R}^6$ and $u_{body} \in \mathbb{R}^3$ denote the origin-relative body's x , z and pitch state and input of the linear dynamics as defined in Equation (2). Additionally, $x_{ee_i} \in \mathbb{R}^4$ and $u_{ee_i} \in \mathbb{R}^2$ represent the origin-relative end-effector's x and z state and input of the linear dynamics as defined in Equation (2). The total number of states $N_x \in \mathbb{N}$ is 36, and the total number of inputs $N_u \in \mathbb{N}$ is 17.

This model is computationally efficient because it combines multiple nonholonomic constraints into a single equation. In addition, since the coordinate system is relative to the ground, it is easy to set the unilateral contact constraint of the wheel and to handle position control. However, this kinematic model doesn't account for full dynamics, which may be crucial during rapid movements. Future work should incorporate a more detailed dynamic model into the CBF framework to enhance safety across various conditions.

III. REAL-TIME PERCEPTIVE MOTION CONTROL FOR TACHYON 3

This chapter describes a perceptive real-time control system utilizing ECBF for Tachyon 3. Fig. 3 shows an overview of the proposed system. The recognition process runs at 1 Hz and the control process runs at 1000 Hz. The proposed system receives the velocity command from the user and outputs the joint command for the next control cycle taking the perceived environmental information. Then, it is converted to the motor current of the joint through PD control. In the following, first, we present an outline of recognition, and then reference motion generation is described. Finally, safety filter using ECBF for Tachyon 3 is shown.

A. Recognition

In this study, planes are extracted by multiple plane segmentation from accumulated point clouds obtained by integrating point clouds from LiDAR and an estimated state. In multiple plane segmentation, clusters of point clouds are generated by region growing [21], and polygonal planes are extracted from each cluster point cloud using RANSAC algorithm [22] and the convex hull algorithm. In addition, a cuboid is extracted from the polygonal plane along the edge to avoid collision with the edge of the step. These polygonal planes and cuboid obstacles are used in real-time motion control.

B. Reference Motion Generation

In reference motion generation, the reference footstep and wheel speed in 2D are calculated from the velocity command. As described in Section II, the Tachyon 3 requires a zero yaw rate when the front and rear driving wheels are landed due to its structure. Therefore, a body's yaw trajectory is generated by cubic spline interpolation so that the yaw rate becomes zero when the front and rear driving wheels are landed in response to the turn command. Subsequently, these 2D references are modified to 3D utilizing multiple planes. For 3D adjustment, the nearest plane from the nominal footstep is selected from multiple planes, and the footstep is projected on the selected plane. Since the extracted plane can become a non-convex region when obstacles are included, a safe convex foothold is extracted through a method similar to [12]. Furthermore, to mitigate unnecessary stepping, ground contact is maintained when the wheels traverse linearly along the same plane.

C. CBF QP Formulation

We employ the ECBF [8] to address the high relative degree constraints arising from position limitations with acceleration inputs for the Tachyon 3's kinematic model (3). To impose ECBF constraints on the nominal input $u^{ref} \in$

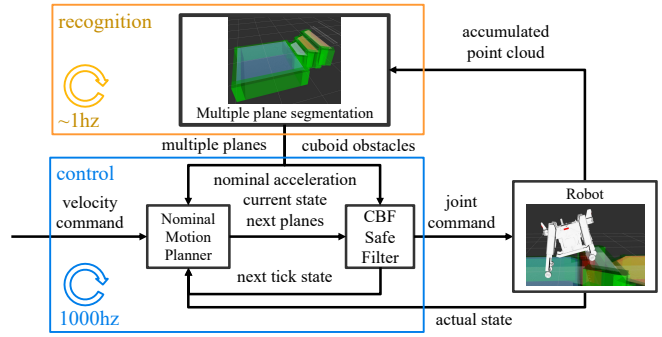


Fig. 3. Overview of real-time perceptive motion control for the six-wheeled-telescopic-legged robot Tachyon 3.

\mathbb{R}^{N_u} , the following convex QP is solved.

$$\begin{aligned}
 u(x) &= \arg \min_{u \in \mathbb{R}^{N_u}, \delta \in \mathbb{R}^{N_{CBF}}} \|u - u^{ref}\|_{w_u}^2 + \|\delta\|_{w_\delta}^2 \quad (4) \\
 s.t. \quad & L_f^{r_i} h_i(x) + L_g L_f^{r_i-1} h_i(x) u \geq -K_i \eta_i(x) + \delta_i \\
 \eta_i(x) &= \begin{bmatrix} h_i(x) \\ \dot{h}_i(x) \\ \vdots \\ L_f^{r_i-1} h_i(x) \end{bmatrix} = \begin{bmatrix} h_i(x) \\ L_f h_i(x) \\ \vdots \\ L_f^{r_i-1} h_i(x) \end{bmatrix} \\
 & Au \leq b \\
 & i = 1, \dots, N_{cbf}
 \end{aligned}$$

Here, $w_u \in \mathbb{R}^{N_u}$ weights nominal input tracking, $h \in \mathbb{R}^{N_{CBF}}$ denotes ECBF constraints, and L_f, L_g are Lie derivatives for f, g in (3). $r_i \in \mathbb{N}$ is the relative degree, with $K_i \in \mathbb{R}^{r_i}$ and $\eta_i \in \mathbb{R}^{r_i}$ representing ECBF gain and time derivatives of $h_i \in \mathbb{R}$. K_i are manually selected to minimize input modifications while aiming for feasibility. Additional input constraints, including control limits, are defined by $A \in \mathbb{R}^{N_c \times N_u}$ and $b \in \mathbb{R}^{N_c}$. For infeasible cases where no solution satisfies both additional and ECBF constraints, we introduce a relaxation parameter $\delta \in \mathbb{R}^{N_{CBF}}$ with penalty weight $w_\delta \in \mathbb{R}^{N_{CBF}}$ to compute approximate solutions. This slack variable allows minor violations of pre-margined constraints, designed to absorb modeling errors and uncertainties, such as those in cuboid obstacles avoidance.

Target inputs are integrated via (3) to obtain wheel velocities and body-relative foot positions. These foot positions are then converted to leg joint angles using inverse kinematics.

IV. ANALYTICAL SMOOTH CBF CONSTRAINTS FOR TACHYON 3

This chapter describes how to design continuously differentiable constraints of ECBF for Tachyon 3's safe locomotion as shown in Fig. 4. Specifically, non-smooth operations in constraints are replaced to smooth approximated operations. We believe that the present method, particularly the smoothing techniques in CBFs, are applicable and useful to widely types of robots.

A. Leg joint limit

The leg joint angle limits are defined by converting from the relative position of each leg to the joint angle as follows

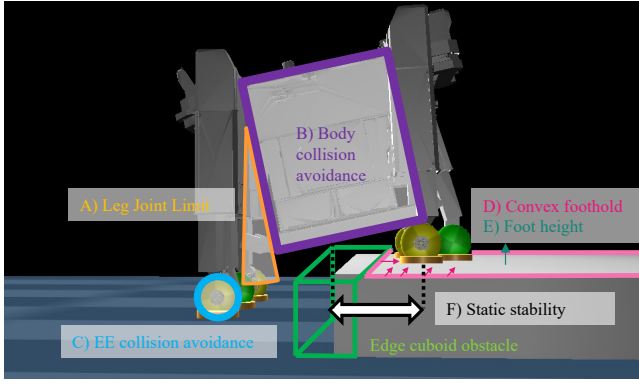


Fig. 4. The six types of ECBF for Tachyon 3's geometric constraints. because the Tachyon 3's knee joint is a prismatic joint.

$$y_1(x) = y_{knee} = \sqrt{\bar{p}_x^2 + \bar{p}_y^2}, \quad y_2(x) = y_{hip} = \text{atan2}(\bar{y}, \bar{x})$$

where $\bar{p}_x, \bar{p}_y \in \mathbb{R}$ are the relative toe position from the base of the leg. $y_{knee}, y_{hip} \in \mathbb{R}$ represents the length of knee joint and the angle of hip joint, respectively. The min / max limits on the joint angles of each leg are described as follows.

$$h_{i,\text{MIN}}(x) = y_i(x) - y_{\text{MIN}} \quad (5)$$

$$h_{i,\text{MAX}}(x) = y_{\text{MAX}} - y_i(x) \quad (6)$$

B. Body collision avoidance constraints

We impose collision avoidance constraints between the body of Tachyon 3 and the edges of the multiple plane environments. Both the plane edges and Tachyon 3's body are represented as cuboids. To construct differentiable CBFs for ensuring collision avoidance, the SAT [10] is utilized to efficiently detect collision between two cuboids. SAT checks whether there is any overlap along the candidate separating axes and determines that there is no collision if there is no overlap in all the separating axes. For simple geometric shapes like cuboids, the SAT is efficient because there are only a few potential separating axes, and specifically, it is computationally more efficient than the popular GJK [23], an efficient collision detecting method between complex convex polytopes.

In this study, we present a novel approach to smooth the SAT, which is typically non-differentiable, by using analytical smoothing functions. For better understanding, we provide a 2D example of SAT in Fig. 5. There are four potential axes, each corresponding to the direction of a side. Extending this to 3D, we have nine axes from the cross products of each cuboid face's normals, and six axes along the normals of the faces. The collision detection method for the 3D case, using these axis candidates, is described as follows.

$$h_{A,B} = \max(y_1(x), \dots, y_{15}(x)) \quad (7)$$

$$y_i = |n_i \cdot \Delta p_{AB}| - \sum_{\substack{j=A,B, \\ k=x,y,z}} |n_i \cdot r_{jk}|$$

where $n_i \in \mathbb{R}^3$ is the separating axis, $\Delta p_{AB} \in \mathbb{R}^3$ is a vector connecting the centers of two cuboids, and $r_{jk} \in \mathbb{R}^3$

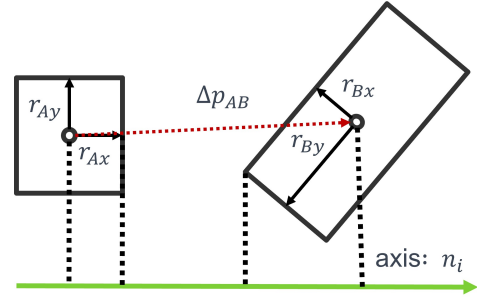


Fig. 5. Separating axis theorem for collision detection between two oriented bounding boxes (2D).

is a vector representing the dimensions (vertical, horizontal, and height) of the cuboid. (7) can be smoothed by replacing the non-smooth maximum and absolute value function.

First, the Log Sum Exp (LSE) [24] is adopted as an approximation to the maximum value function with bounds as in [25],

$$f_{\text{smax}}(x_1, \dots, x_n) = \frac{1}{\alpha} \log \sum_{i=1}^n \exp(\alpha x_i) \quad (8)$$

$$0 \leq f_{\text{smax}}(x_1, \dots, x_n) - \max(x_1, \dots, x_n) \leq \frac{\log(n)}{\alpha}$$

Here, the parameter $\alpha \in \mathbb{R}$ modulates the degree of smoothness, with larger values yielding greater accuracy. To address the overflow problem associated with LSE, we apply it only when the output value is smaller than a threshold, using the calculation of the non-differentiable SAT.

Next, the xtanh function is employed as a precise surrogate for the absolute value function, as described in the following expression [26].

$$f_{\text{sabs}}(x) = x \tanh(\alpha x) \quad (9)$$

$$|f_{\text{sabs}}(x) - |x|| < \frac{1}{\alpha}$$

The parameter $\alpha \in \mathbb{R}$ adjusts the smoothness of the approximation, similar to LSE.

Finally, the detection of collision between the body and the edge is achieved by smoothing (7) using (8) and (9):

$$h_{A,B} = f_{\text{smax}}(y_1(x), \dots, y_{15}(x)) \quad (10)$$

$$y_i(x) = f_{\text{sabs}}(n_i \cdot \Delta p_{A,B}) - \sum_{\substack{j=A,B, \\ k=x,y,z}} f_{\text{sabs}}(n_i \cdot r_{jk})$$

C. End effector collision avoidance constraints

The toe wheel is approximated by a sphere. The smooth function for detection of collision between the sphere toe and the cuboid edge is expressed using superellipsoid:

$$h(x) = \left(\frac{\bar{p}_X(x)}{a} \right)^{2N} + \left(\frac{\bar{p}_Y(x)}{b} \right)^{2N} + \left(\frac{\bar{p}_Z(x)}{c} \right)^{2N} - 1 \quad (11)$$

$$\bar{p}(x) = R_{\text{edge}}^T (p_{\text{foot}}(x) - p_{\text{edge}})$$

where $\bar{p}(x) \in \mathbb{R}^3$ is the relative position of the toe from the center of the cuboid $p_{\text{edge}} \in \mathbb{R}^3$, $R_{\text{edge}} \in \mathbb{R}^{3 \times 3}$. $a, b, c \in \mathbb{R}$ is a parameter representing the length(x), width(y), and

height(z) of the cuboid, respectively, where, $N \in \mathbb{N}$ is a parameter used to adjust the smoothing. A larger value of N leads to a more precise cuboid, while a smaller value of N results in smoother behavior at non-differentiable vertices.

D. Convex foothold constraints

The convex foothold constraints are applied to CBF.

$$h(x) = a_{ij}^T p_i(x) + b_{ij} \quad (12)$$

where $a_{ij} \in \mathbb{R}^2$, $b_{ij} \in \mathbb{R}$ represent the half-space constraints of the j th edge of the convex polygon for the world toe xy position $p_i(x) \in \mathbb{R}^2$'s safe region. These constraints are activated during the foot-lowering phase of the swing and remain in effect throughout the entire stance phase. When the leg is within the safe convex region, the CBF ensures it does not step outside this region. Conversely, when the swing leg is outside the convex safe region, it functions as a Control Lyapunov Function, guiding the leg towards the safe region, as demonstrated in [9].

E. Foot height constraints

To prevent collisions with the horizontal plane, a height constraint of $h(\bar{z}) = \bar{z}_i - \bar{z}_{floor}$ is applied to each toe. This ensures each toe remains above the floor level.

F. Approximate static stability constraints

To maintain static stability, we constrain the body position within the convex hull of the feet. This serves as a proxy for keeping the center of mass in the support polygon. The constraint regarding the local translational position between the body $\bar{p}_{body} \in \mathbb{R}$ and the toes $\bar{p}_i \in \mathbb{R}$ is formulated as:

$$h(x) = a_i(\bar{p}_i(x) - \bar{p}_{body}(x)) + b_i \quad (13)$$

For each foot, both upper and lower limit constraints are applied, resulting in two constraints per foot to define the boundaries of the convex hull.

V. EXPERIMENTS

We herein show the effectiveness of the proposed method through experiments. First, we present a benchmark for the computation time of the proposed SSAT against the previous method. Then, the effectiveness of real-time control for Tachyon 3 using SSAT CBF was confirmed in a simulation. Finally, the proposed method was integrated on the real machine.

A. Comparison of Smooth Collision Detection

The computational efficiency of SSAT is verified by comparing with other smoothing functions and other collision detection methods. Table I shows the results of the calculation time of random cuboid's collision detection. SSAT with LSE and xtnh is the proposed method explained in Section IV. For comparison, two other smoothing functions are considered for SAT : the square root of square (sqrt) $f_{sabs}(x) = \sqrt{(x + \epsilon)^2}$ with a smoothing parameter $\epsilon \in \mathbb{R}$ as smooth absolute value function, and the Boltzmann operator (boltz) $f_{smax}(x_1, \dots, x_n) = \frac{\sum_{i=1}^n x_i \exp(\alpha x_i)}{\sum_{i=1}^n \exp(\alpha x_i)}$ with smoothing

TABLE I
AVERAGE COMPUTATION TIMES OF CUBOID COLLISION
DETECTION

Method	SSAT with LSE and xtnh	SSAT with LSE and sqrt	SSAT with Boltz and xtnh	SAT	GJK	LP
evaluate [μ s]	0.47	0.57	0.47	0.05	0.50	9.35
evaluate + differentiate [μ s]	0.78	0.93	1.04	-	-	-

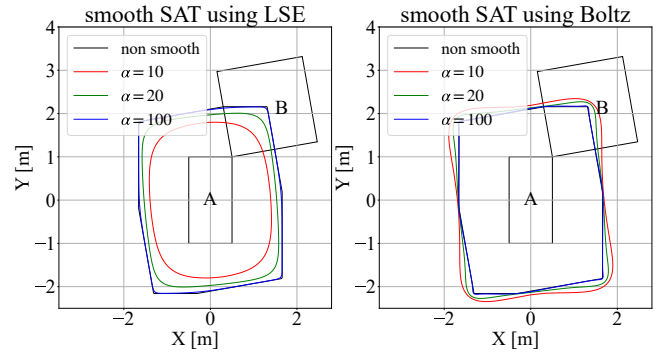


Fig. 6. The collision detection boundary using the smooth maximum method for 2D SAT. The left graph uses LSE, while the right graph uses the Boltzmann operator. The red, green, and blue lines represent the trajectory of the center point of box B as it comes into contact with box A, with different values of the smoothing parameter α . Box B is movable on the XY plane and tilted at a 10-degree angle. The black line represents the boundary when the SAT is used for collision detection.

parameter $\alpha \in \mathbb{R}$ as smooth maximum. SAT and GJK are non-differentiable implementation in hpp-fcl [27]. LP refers to the previous method [7], [14] using Linear Programming.

While the proposed SSAT is slower than the non-differentiable SAT due to the additional smoothing process, it is sufficiently fast to be comparable to GJK (within 1 μ s). Furthermore, compared with the optimization-based method, the proposed method is more than 10 times faster. Therefore, the proposed method is the fastest differentiable cuboid collision detection. About SSAT, the computation time for calculating collision detection value, it's Jacobian and Hessian with respect to the one box's position was also measured. The result of Table I indicates that utilizing LSE and xtnh is most efficient for SSAT.

The effectiveness of the SSAT was assessed by employing both the LSE and the Boltzmann operator as potential smooth maximum functions. As depicted in Fig. 6, the Boltzmann operator results in a non-convex shape, whereas the LSE maintains a convex shape, making it easier to handle within collision avoidance. Furthermore, the LSE has constant upper and lower bounds for the error of the maximum function (8). By incorporating a margin from (8), (9), it is possible to strictly satisfy collision avoidance. Considering these advantages and the computational time aspect mentioned earlier, the LSE has been selected for SSAT.

Here we examined the computation time of a single CBF constraint for SSAT of Tachyon 3. Since the proposed method can be computed analytically, code generation [28]

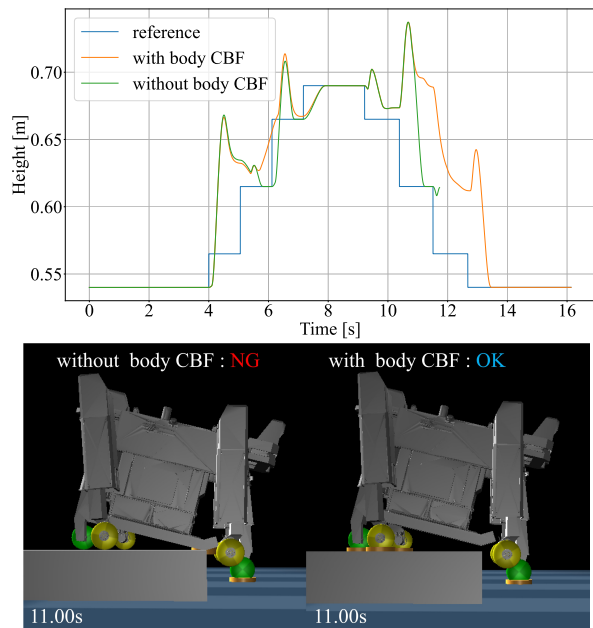


Fig. 7. The above row shows height trajectories of the body. The bottom row shows the difference of the pose with and without the proposed body SSAT CBF.

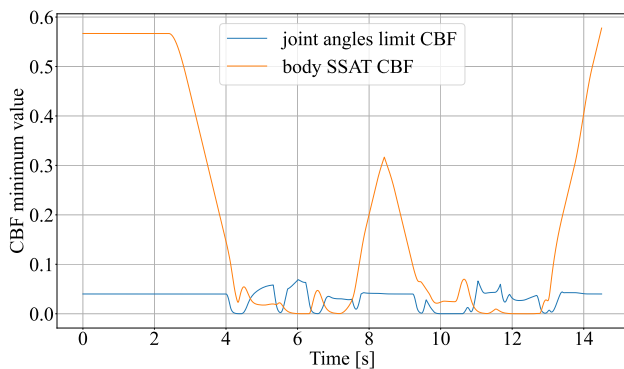


Fig. 8. Body SSAT CBF and leg joint angles limit CBF minimum values. using symbolic differentiation is performed. Thanks to the offline optimization of code, sparse data can be handled efficiently. Furthermore, constant computational complexity is suitable for achieve real-time control. On an embedded PC of Tachyon 3, the computational time of CBF calculation was less than $4 \mu\text{s}$. This computation time is faster than the collision detection of LP in Table I, even though the information of the second derivative is required. As such, it can be a faster method than the existing CBF [6], [7] using optimization-based method.

B. Simulation Experiments for Tachyon 3

The CBF constraints for Tachyon 3, as proposed in Section IV, have been verified by a simulation of locomotion. In this experiment, we compare the safety with and without body SSAT CBF via walking over a step using a low center of gravity posture. MuJoCo [29] is used for the simulation. Fig. 7 shows the height trajectory of the body. The orange modified body trajectory is raised against the blue reference body trajectory. Notably, the green trajectory demonstrates that, without the implementation of the body SSAT CBF, the

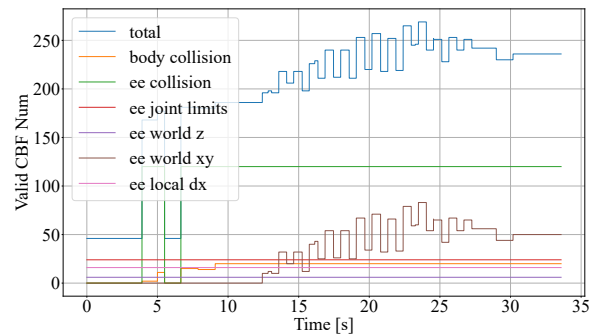


Fig. 9. The number of valid CBF. The horizontal axis is time [s].

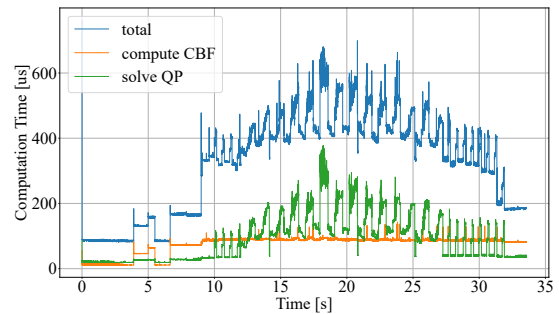


Fig. 10. Computation time of real-time control. The vertical axis is computation time [μs] and the horizontal axis is time [s].

robot experiences collisions with environmental boundaries. This adverse event is effectively avoided by activating the body SSAT CBF. Furthermore, we analyzed which CBF constraints influenced the adjustment of body height. Fig. 8 displays the minimum values of both the body SSAT CBF and the leg's joint angle limits CBF. The proximity of these values to zero suggests that the constraints were actively engaged. Consequently, it is evident that the cuboid collision avoidance CBF for the body and the joint angle limits CBF for the legs are efficacious in ensuring safety when the robot navigates over a step with a low center of gravity.

C. Hardware Experiments for Tachyon 3

The practical usefulness of the proposed method is shown by a perceptive experiment using Tachyon 3 in an unknown environment. In this experiment, the real-time control proposed in Section III generates motion at a high frequency of 1000 Hz considering safety, such as range of motion and environmental collision avoidance, as shown in Fig. 1. Using the proposed method, Tachyon 3 walks up while recognizing 5 step stairs with a height of 16.5 cm and a depth of 35 cm. The body height is modified above by 15 cm from the standard height when walking on stairs.

Fig. 9 shows the number of CBFs applied to the QP during experiments. The blue curve, representing the total number of CBFs, demonstrates that a maximum of 269 CBF constraints are computed and incorporated into the QP solver. Fig. 10 shows the computation time of each control update measured on the onboard PC with the CPU Intel(R) Core(TM) i7-8850 H CPU @ 2.60 GHz. The robot was able to generate motion within 1 ms as shown by the blue line (total computation time). Orange line represents the calculation time of CBF constraints with a maximum

of 269 CBF constraints computed in about 80 μ s. Among them, 20 obstacles are set for the six toes and the one body, resulting in a total of 140 CBF. The previous study using CBF [6] deals with only 4 obstacles of convex shape with a maximum number of vertices of 15 for 1 body's rectangle, and the paper [7] deals with 7 primitive shapes across 3 rectangular obstacles. Therefore, compared with these studies using CBF, our formulation can deal with sufficiently larger number obstacles with a sufficiently small computation time for a control cycle of 1 ms.

VI. CONCLUSIONS

In this paper, we developed a lightweight perceptive real-time control system generating feasible motion within 1 ms using an efficient kinematic model and analytically smooth CBF formulations to satisfy various constraints such as 3D collision avoidance and joint limits for Tachyon 3. Notably, we proposed a novel SSAT as an higher-order differentiable collision detection method. The computational efficiency of SSAT was demonstrated through comparison with other collision detection methods. The selection of collision detection method depends on the required precision and the nature of the workspace, with SSAT proving particularly efficient for cuboid-approximated robots in constrained environments. We validated the safety performance of the proposed CBF in simulations and demonstrated its practical applicability through hardware experiments of perceptive locomotion over stair climbing with on-the-fly environment perception. Our control system can generate motion without trajectory planning such as MPC. The computationally efficient SSAT CBF can be applied to body-environment collisions, other links, and self-collision avoidance.

The present system is limited to application to the movement of environments composed of planes in statically stable robots. An issue for the future is dealing with cumbersome environments and dynamic movement. Since CBF QP itself can be leveraged a real-time safety filter for strict constraint satisfaction, it is expected to work with other reference motion generation such as RL and MPC.

REFERENCES

- [1] Sony Group Corporation, "New Mobility", https://www.sony.com/en/SonyInfo/research/technologies/new_mobility.
- [2] M. Bjelonic, "Planning and control for hybrid locomotion of wheeled-legged robots," Ph.D. dissertation, ETH Zurich, 2021.
- [3] J. Lee, M. Bjelonic, and M. Hutter, "Control of wheeled-legged quadrupeds using deep reinforcement learning," in *Climbing and Walking Robots Conference*. Springer, 2022, pp. 119–127.
- [4] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6271–6278.
- [5] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8352–8358.
- [6] Q. Liao, Z. Li, A. Thirugnanam, J. Zeng, and K. Sreenath, "Walking in narrow spaces: Safety-critical locomotion control for quadrupedal robots with duality-based optimization," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2723–2730.

- [7] B. Dai, R. Khorrambakht, P. Krishnamurthy, V. Gonçaves, A. Tzes, and F. Khorrami, "Safe navigation and obstacle avoidance using differentiable optimization based control barrier functions," *arXiv preprint arXiv:2304.08586*, 2023.
- [8] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 322–328.
- [9] W. Xiao, C. A. Belta, and C. G. Cassandras, "High order control lyapunov-barrier functions for temporal logic specifications," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4886–4891.
- [10] S. Gottschalk, "Separating axis theorem," 1996.
- [11] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [12] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, 2023.
- [13] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [14] K. Tracy, T. A. Howell, and Z. Manchester, "Differentiable collision detection for a set of convex primitives," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3663–3670.
- [15] L. Montaut, Q. Le Lidec, A. Bambade, V. Petrik, J. Sivic, and J. Carpentier, "Differentiable collision detection: a randomized smoothing approach," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3240–3246.
- [16] A. Thirugnanam, J. Zeng, and K. Sreenath, "Nonsmooth control barrier functions for obstacle avoidance between convex regions," *arXiv preprint arXiv:2306.13259*, 2023.
- [17] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [18] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3882–3889.
- [19] Y. Kamikawa, M. Kinoshita, N. Takasugi, K. Sugimoto, T. Kai, T. Kito, A. Sakamoto, K. Nagasaka, and Y. Kawanami, "Tachyon: design and control of high payload, robust, and dynamic quadruped robot with series-parallel elastic actuators," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 894–901.
- [20] L. Mao, Y. Tian, F. Gao, and Y. Zhao, "Novel method of gait switching in six-legged robot walking on continuous-nondifferentiable terrain by utilizing stability and interference criteria," *Science China Technological Sciences*, vol. 63, no. 12, pp. 2527–2540, 2020.
- [21] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 36, no. 5, pp. 248–253, 2006.
- [22] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Computer graphics forum*, vol. 26, no. 2. Wiley Online Library, 2007, pp. 214–226.
- [23] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.
- [24] P. Blanchard, D. J. Higham, and N. J. Higham, "Accurately computing the log-sum-exp and softmax functions," *IMA Journal of Numerical Analysis*, vol. 41, no. 4, pp. 2311–2330, 2021.
- [25] T. G. Molnar and A. D. Ames, "Composing control barrier functions for complex safety specifications," *IEEE Control Systems Letters*, 2023.
- [26] Y. J. Bagul, "A smooth transcendental approximation to $|x|$," *International J. of Math. Sci. & Engg. Appls.(IJMSEA)*, vol. 11, no. II, pp. 213–217, 2017.
- [27] J. Pan, S. Chitta, D. Manocha, F. Lamiraux, J. Mirabel, J. Carpentier, *et al.*, "Hpp-fcl: an extension of the flexible collision library."
- [28] O. Čertík, I. Fernando, S. Garg, T. Rathnayake, *et al.*, "Symengine: A fast symbolic manipulation library," URL <https://github.com/symengine/symengine>, 2013.
- [29] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.