2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Optimal Prescribed-Time Control based Reactive Planning System for Quadruped Robot Navigation

Shaohang Xu[1,2], Wentao Zhang[1], Chin Pang Ho[2], and Lijun Zhu[1]

***Abstract*— In this paper, we propose a reactive planning system for quadruped robots based on prescribed-time control. The navigation of the quadruped robot is fundamentally depicted as omnidirectional movements, while a feedback control law is formulated to address any deviations the robot may encounter. In particular, our proposed feedback control system is theoretically proven to achieve convergence within a predefined finite time that is specified by the user. To further compute the optimal convergent time and the local goal state, we present a high-level planning node encompassing terrain-aware kinodynamic search and spatiotemporal trajectory optimization, which can generate collision-free, smooth, and efficient trajectories. The effectiveness of our proposed framework is validated through both numerical simulation and real-robot experiments in indoor and outdoor environments, including scenarios with cluttered obstacles, slopes, and external disturbances.**

I. Introduction

Quadruped robots offer significant advantages compared to other types of robotic machines for tasks such as rescue operations and load transportation [1]–[3]. In real-world applications, autonomous navigation is a vital task, which requires planning and tracking a collision-free trajectory. However, achieving fully autonomous quadrupedal locomotion remains a challenge. Firstly, high-level planning requires simultaneously taking into account collision avoidance, time efficiency, and complicated terrains. Additionally, tracking the waypoints (subgoals) from high-level planning can result in non-smooth motion on real robots. Furthermore, open-loop planners are unable to address robot deviations in a smooth or automated manner with high frequency.

In this paper, we propose a reactive planning framework to address these issues systematically. Traditional reactive planning relies on Control Lyapunov Functions (CLFs) [4], offering a smooth feedback control law that instantaneously adjusts the path to the goal [5]–[7]. Although CLFs can ensure the convergence of the closed-loop system, the convergence is often achieved as the time tends to be infinite in theory, resulting in an infinite arrival time at the planned goal. To address this concern, we incorporate the prescribed-time control theory [8] into the framework of reactive planning. Next, we give a brief review of the related work and summarize our contributions.

[1] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, China, shaohangxu@hust.edu.cn, wentaozhang@hust.edu.cn, ljzhu@hust.edu.cn

[2] School of Data Science, City University of Hong Kong, HKSAR, shaohanxu2-c@my.cityu.edu.hk, clint.ho@cityu.edu.hk

Fig. 1. Figure showing our quadruped robots navigating in different hardware configurations in indoor (left) and outdoor (right) environments.

A. Related Work

In the context of trajectory planning for legged robots on complex terrains, various methods have been proposed in the literature. Some of these methods are sampling-based, such as [9]–[12]. These methods do not take into account the robot dynamics, which may lead to infeasible motions in practice. In [13], the kinodynamic model is proposed to generate trajectories, but only a small planning horizon is employed. Although the omnidirectional movement of quadruped robots is considered in the above methods, they do not consider that quadruped robots are often weak in lateral walking and turning. Moreover, the optimality of the trajectory is usually not guaranteed.

A better way to achieve optimality is trajectory optimization (TO), as presented in [14]–[17]. In [15], a complete trajectory optimization method is presented to consider centroidal dynamics, footholds, gait schedule, etc. However, the local optima is difficult to avoid due to the non-convexity of the constraints, and the computation time is too long for real robot applications. Recent works [17], [18] incorporate the jumping gait into quadrupedal navigation and optimize the whole-body trajectory. Unfortunately, all the above-mentioned methods only consider the space aspect to avoid collisions but lack consideration of the time aspect.

Traditional motion planning algorithms are often ambiguous about robot deviations from the planned trajectory. Indeed, they provide only *open-loop* control for tracking. In contrast, reactive planning provides smooth *closed-loop* control by replacing the concept of trajectory with that of a vector field [5], [6], [19]–[22]. Recently, a pioneering work [7] presents a CLF-based reactive planning framework and combines the CLF with RRT* for biped robot navigation. Theoretically, CLFs provide a family of trajectories that converge to the goal [4]. However, CLFs can only guarantee the asymptotical stability of the closed-loop system, meaning that the arriving time at the goal can be infinite.

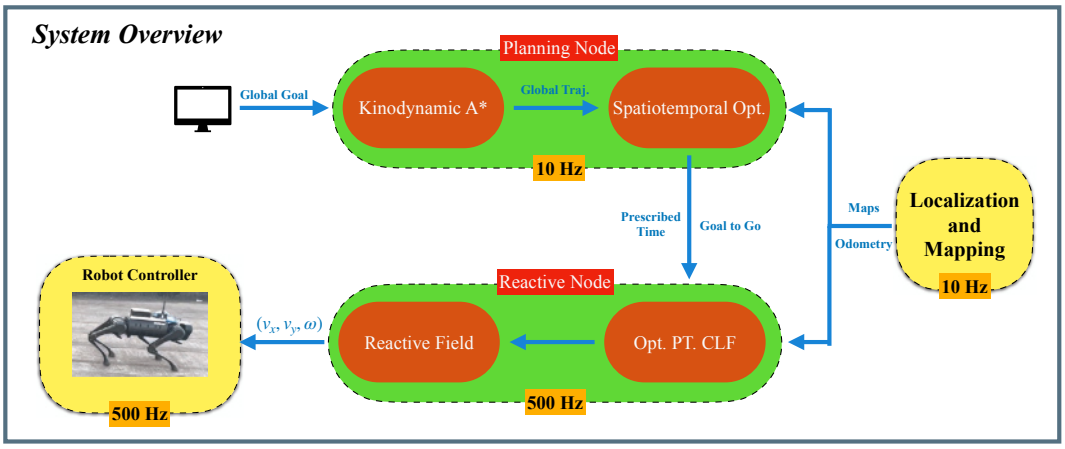


Fig. 2. Figure showing the proposed planning system. See Section II for a detailed introduction.

The issue of the convergent time has been introduced in the control community. To solve this issue, finite-time stability and fixed-time stability are proposed [23], which can guarantee finite/fixed convergent time. However, the finite/fixed convergent time is usually dependent on the controller parameters and cannot be predefined arbitrarily. Recently, Song et al. [8] proposes the concept of prescribed-time stability, which provides a way for controller design to guarantee an arbitrarily prescribed convergent time. However, although the prescribed time is arbitrary, it is still an open problem about how to prescribe an optimal one. In particular, in the context of legged robot navigation, prescribed-time control is rarely explored.

B. Contribution

The contributions of this paper can be stated as follows:

- We propose a novel reactive planning system based on prescribed-time control for quadruped robot navigation. In particular, the omnidirectional movement is considered in the robot dynamics, while the lateral velocity and the turning velocity are minimized by analytically solving a optimization problem. To the best of our knowledge, this is the first time that prescribed-time control is applied to legged robot navigation.
- Theoretical proof is established for the convergence of the closed-loop control system within a predetermined fixed time. A vector field is created to guide the robot to precisely reach the local goal at the planned time, which can autonomously handle robot deviations.
- A high-level planner is proposed to provide the optimal convergent time and the local target for the reactive planning node. In particular, it takes into consideration collision avoidance, trajectory smoothness, and time optimality within a coarse-to-fine framework.
- Extensive simulation and field experiments are conducted to validate the effectiveness of the proposed system on a Unitree A1 robot, as shown in Fig. 1.

The structure of this paper is presented as follows. Section II provides an overview of the entire system. Section III introduces each module of our planning system in detail. Section IV provides a comprehensive experimental evaluation. Finally, we conclude the paper in Section V.

II. System Overview

The overall planning system is illustrated in Fig. 2. Based on state estimation and mapping, we present a kinodynamic A* algorithm to search for a nearly optimal path. The initial time assignment is determined based on an optimal boundary value problem. To enhance smoothness and time optimality, a spatiotemporal optimization method refines the coarse trajectory. The local target with the planned time is obtained from the optimized trajectory.

Instead of waypoint tracking, we introduce a high-frequency reactive planner node based on prescribed-time control. The solution trajectory of the closed-loop system naturally generates a vector field that guides the robot to the subgoal. In practice, it indeed provides automatic and instantaneous planning to deal with robot deviations. Notably, the prescribed-time control law enables the attainment of the planned arrival time at the subgoal. In the final step, the planning system generates control commands for the locomotion controller.

III. Main Method

A. Prelimenary

We first introduce the basic concept of CLF. Consider the control system:

$$\dot{x} = f(x, u), \tag{1}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input. Then, the CLF is defined as follows:

Definition 1: [4], [7] Consider the control system (1) and a differential function $V : \mathcal{X} \to \mathbb{R}$, for which there is some neighborhood $\mathcal{O}$ of the equilibrium state x^0 for (1). Then, V is a CLF for (1) if:

- the set $\{x \in \mathcal{X} | V(x) \leq \epsilon\}$ is a compact subset of $\mathcal{O}$ for each $\epsilon > 0$ small enough.
- $V(x^0) = 0$, and $V(x) > 0$ for each $x \in \mathcal{O}$, $x \neq x^0$.
- For each $x \in \mathcal{O}$, $x \neq x^0$, there exists an admissible control policy u, such that $\nabla V(x) f(x, u) < 0$.

The CLF is based on the Lyapunov stability ideas, following the asymptotic stability. Without loss of generality, we assume the equilibrium state x^0 is 0.

Definition 2: [24] Consider a nonautonomous system:

$$\dot{x} = f(x, t), \tag{2}$$

where $f : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in x. Then, the equilibrium state $x = 0$ is asymptotically stable if:

- There exists a class of $\mathcal{K}$ function β such that $\|x(t)\| \leq \beta(\|x(0)\|)$.
- $x(t) \to 0$ as $t \to \infty$.

The concepts of CLF and asymptotical stability provide an elegant way for reactive planning [5]–[7]. However, as we commented before, the above concepts only guarantee convergence as $t \to \infty$, and thus the convergence time may be too slow in practice. To handle this issue, we introduce the concept of prescribed-time stability:

Definition 3: [8], [25] The system (2) is prescribed-time asymptotically stable in time T, if there exist a function $\mu : [0, T) \to \mathbb{R}_+$ with μ increasing to ∞ as $t \to T$ and a class $\mathcal{KL}$ function β such that

$$\|x(t)\| \leq \beta(\|x(0)\|, \mu(t))$$

holds for $\forall t \in [0, T)$, where $T > 0$ is a finite scalar that can be prescribed in the design.

Next, we will design a novel feedback control law and a related CLF for quadruped robot navigation, such that the closed-loop system satisfies the prescribed-time stability.

B. Prescribed-Time Control Design

We leverage the omnidirectional robot model to describe the dynamics of quadruped robot navigation [5]–[7]:

$$\begin{bmatrix} \dot{r} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} -\cos(\delta) & -\sin(\delta) \\ \frac{1}{r}\sin(\delta) & -\frac{1}{r}\cos(\delta) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \end{bmatrix}, \tag{3}$$

where r is the Euclidean distance between the robot position and the target position, $\delta \in [-\pi, \pi)$ is the angle from the heading direction of the robot to the line of sight from the robot to the goal, ω is the yaw angle velocity of the robot, v_x and v_y are the robot velocity along the x- and y- axis in the body frame, respectively. Clearly, in this control system, r and δ are the system states, while v_x, v_y and ω are the control inputs. This nonlinear control system (3) can be reformulated via feedback linearization:

$$\begin{bmatrix} \dot{r} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} v_r \\ v_\delta \end{bmatrix}, \tag{4}$$

where the linearized control input v_r and v_δ are defined as

$$\begin{bmatrix} v_r \\ v_\delta \end{bmatrix} \triangleq \begin{bmatrix} -\cos(\delta) & -\sin(\delta) \\ \frac{1}{r}\sin(\delta) & -\frac{1}{r}\cos(\delta) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \end{bmatrix}. \tag{5}$$

Before we design the prescribe-time control law for v_r and v_δ, we first introduce a proposition from [7]:

Proposition 1: [7] Given the equation (5), the following optimization problem can be analytically solved,

$$\min_{v_y, \omega} v_y^2 + \alpha\omega^2,$$

where α is a positive weight scalar, and the analytical solution is:

$$w^* = \frac{r\cos(\delta)(v_r \sin(\delta) + r v_\delta \cos(\delta))}{r^2\cos^2(\delta) + \alpha}, \tag{6a}$$

$$v_x^* = \frac{-v_r \cos(\delta)(r^2 + \alpha) + \alpha v_\delta \sin(\delta)}{r^2\cos^2(\delta) + \alpha}, \tag{6b}$$

$$v_y^* = \frac{-\alpha(v_r \sin(\delta) + r v_\delta \cos(\delta))}{r^2\cos^2(\delta) + \alpha}. \tag{6c}$$

Remark 1: It can be seen that we can obtain the optimal (i.e., minimum) lateral velocity v_y and ω given some known v_r and v_δ. It is practically viable for legged robots, since the legged robots are often weak in lateral walking and turning, e.g., the Cassie robot in [7] and the Unitree A1 robot.

Now, we are in a position to design our prescribed-time feedback control law, namely, on the time interval $[0, T)$,

$$v_r = -(k_r + \frac{1+m}{T})\frac{T^{1+m}}{(T-t)^{1+m}} r, \tag{7a}$$

$$v_\delta = -(k_\delta + \frac{1+m}{T})\frac{T^{1+m}}{(T-t)^{1+m}} \delta, \tag{7b}$$

where k_r, k_δ are positive weight scalars, T is the prescribed convergence time, t is the current time, m is a positive integer. With the above control law, we can prove the prescribed-time stability of the closed-loop system, which is the main theoretical result of this paper:

Theorem 1: The closed-loop control system (4) with (7) is prescribed-time asymptotically stable in time T.

Proof: Define auxiliary functions $\mu \triangleq \frac{T^{1+m}}{(T-t)^{1+m}}$, and $w_r \triangleq \mu r$, $w_\delta \triangleq \mu\delta$. Then, we define the candidate Lyapunov function as $V \triangleq \frac{1}{2}(w_r)^2 + \frac{1}{2}(w_\delta)^2$. It can be seen that the control system can be decomposed into two components for r and δ. First, let us deal with r and define $V_r \triangleq \frac{1}{2}(w_r)^2$. Then we have

$$\dot{V}_r = w_r \dot{w}_r = w_r \dot{\mu} r + w_r \mu v_r \tag{8a}$$

$$= w_r \frac{1+m}{T}(\frac{T}{T-t})^{2+m} r + w_r \mu v_r \tag{8b}$$

$$= w_r \mu (\frac{1+m}{T} \mu r (\frac{T}{T-t})^{-m}) + w_r \mu v_r \tag{8c}$$

$$\leq w_r \mu (\frac{1+m}{T} \mu r) + w_r \mu v_r \tag{8d}$$

$$= -k_r w_r \mu^2 r = -2k_r \mu V_r. \tag{8e}$$

From Lemma 1 in [8], we know that $\dot{V}_r \leq -2k_r\mu V_r \Rightarrow V_r \leq \psi^{2k_r} V(0)$, where ψ is a monotonically decreasing function of t and $\psi(0) = 1$ and $\psi(T) = 0$ which implies that V and

w are both bounded on $[0, T)$. Then, we have

$$r^2 = \frac{w_r^2}{\mu^2} = \frac{2V_r}{\mu^2} \leq \frac{\psi^{2k_r} r^2(0)}{\mu^2},$$
$$\Rightarrow \|r(t)\| \leq \frac{\psi^{k_r}(t)}{\mu(t)} \|r(0)\|.$$

Similarly, we can also have $\|\delta(t)\| \leq \frac{\psi^{k_\delta}(t)}{\mu(t)} \|\delta(0)\|$. Therefore, from Definition 3, we conclude that the overall closed-loop control system (4) with (7) is prescribed-time asymptotically stable in time T. ■

In real-world scenarios, conventional CLF methods (such as in [7]) can address robot deviations caused by disturbances through automatic and immediate planning. Our proposed method shares this capability, but can further achieve the convergence within the predetermined time.

A basic question here is whether the prescribed-time stability requires infinite control inputs, which are not desirable in robotic applications. To answer this question, we introduce the following proposition.

Proposition 2: [8] The control inputs v_r and v_δ in (7) are both bounded on $[0, T)$ for the control system (4),

On the real robot, it is common that the hardware system is not well-synchronized. Then, it might happen that the states are not exactly zero when $t \geq T$. It is worth noting that in the control community, this fact is often neglected, and the prescribed-time control law is only defined on the time interval $[0, T)$. However, for robotic applications, we want to emphasize that the undefined control law after the prescribed time T is undesirable. To handle this issue, we extend the original control law to the time interval $[T, \infty)$ as follows:

$$v_r = -\frac{k_{r_1}}{k_{r_2} + r} r, \tag{9a}$$
$$v_\delta = -\frac{k_{\delta_1}}{k_{\delta_2} + r} \delta, \tag{9b}$$

where $k_{r_1}, k_{r_2}, k_{\delta_1}, k_{\delta_2}$ are all positive weight scalars. Clearly, the extended control law on $[T, \infty)$ can also drive the state trajectory to the origin, while it cannot guarantee the convergence time.

Proposition 3: $\bar{V} = \frac{1}{2} r^2 + \frac{1}{2} \delta^2$ is a CLF.

Proposition 4: Suppose the control system (4) evolves until the time T, but $r(T)$ and $\delta(T)$ are not both zero. Then, the closed-loop control system (4) is asymptotically convergent to zero with the extended control law in (9), i.e., when $t \to \infty$, $r \to 0$ and $\delta \to 0$,

The above two propositions can be proved by checking Definition 1 and 2, and the details are neglected due to the page limit. Proposition 3 shows that our proposed method is a special case of CLF-based methods [5]–[7]. This is also consistent with the fact that prescribed-time stability is a special case of asymptotical stability.

Generally, the convergence time T could be any positive scalar. In this paper, we aim to prescribe an *optimal* arriving time via a high-level planner, which is introduced in the following.

C. High-level Planner

The design criteria for our high-level planner are as follows: (C1) The robot should arrive at the goal as soon as possible; (C2) The control input should be minimized to save onboard energy; (C3) The robot should be away from obstacles within safety margins; (C4) The robot should autonomously decide whether a complex terrain should be climbed or bypassed.

1) Coarse Trajectory Search

First, we simplify the robot dynamics and utilize the kinodynamic A* method to search for a coarse trajectory. To enhance the quality of trajectories for quadruped robots, we integrate motion primitives, terrain evaluation metrics, optimal control minimization, and similarity penalties into both the actual cost and heuristic cost functions.

Node Expansion: In this global planner, we describe the quadruped locomotion along two motion axes independently, without considering the yaw angle. We choose to utilize acceleration as the control variable and take the position and velocity as state variables, i.e.,

$$\dot{x}^c = Ax^c + u^c, \quad A = \begin{bmatrix} 0 & I_2 \\ 0 & 0 \end{bmatrix} \tag{10}$$

where $x^c \triangleq [p_x, p_y, v_x, v_y]^\top$, $u^c \triangleq [0, 0, a_x, a_y]^\top$, $p_x(p_y)$, $v_x(v_y)$ and $a_x(a_y)$ are the position, velocity and acceleration of the base in the x-(y-) axis of the world frame respectively. Note that the velocity and acceleration are both limited: $[v_x, v_y]^\top \in [-v_{\max}, v_{\max}]^2$, $[a_x, a_y]^\top \in [-a_{\max}, a_{\max}]^2$. To generate a set of feasible motion primitives, we discretize the control space along either axis into m_u uniform grids. Each control sample is applied for a duration of $\tau, 2\tau, \cdots, m_t\tau$, respectively. The successor state of each primitive can be obtained by solving the ordinary differential equation (11):

$$x^c(t) = e^{At} x^c(0) + \int_0^t e^{A(t-\tau)} u^c(\tau) d\tau. \tag{11}$$

The actual and heuristic costs of each successor state are updated as follows if it satisfies two conditions: (i) it is collision-free, and (ii) its velocity is within the predefined limits. If these criteria are met, the successor state is added to the open set for expansion, provided it has not already been expanded.

Actual Cost: The actual cost e_c is designed as

$$e_c = \sum_{i=1}^{M} (w_t t_i + w_e f_i^e + w_o f_i^o + w_{te} f_i^{te}), \tag{12}$$

where the actual cost is a combination of four distinct cost terms, with different weights assigned to each term based on its relative importance towards the corresponding criterion. To introduce the cost terms, we first define the trajectory consisting of M motion primitives, with control inputs, durations, and intermediate states denoted by $\{u_1^c, \cdots, u_M^c\}, \{t_1, \cdots, t_M\}$, and $\{x_1^c, \cdots, x_M^c\}$.

The first term in (12) is for criterion (C1) to minimize the total time; The cost term $f_i^e \triangleq \|u_i^c\|^2$ is for criterion (C2); For (3), we have $f_i^o \triangleq m_o(d_s)$, where $m_o(d_s)$ denotes

the number of the obstacle grids around the node within a predefined distance d_s. This term penalizes trajectories close to obstacles and does not rely on any ESDF map. Finally, the cost term $f_i^{te} \triangleq \|h(x_i^c)\|^2$ addresses criterion (C4) by incorporating the terrain cost score $h(x^c)$ associated with the state x^c.

Heuristic Cost: We employ the Pontryagin minimum principle [26] instead of the standard distance functions in A* algorithms to determine the optimal cost $g_e(x_c^c, x_g^c, T)$ from the current state x_c^c to the final goal state x_g^c with the duration T. Note that while x_c^c and x_g^c are given in the searching algorithm, the value of T is unknown. We replace T with $T^* \triangleq \arg\min_T g_e(x_c^c, x_g^c, T)$ to get the optimal value.

In addition to the optimal travel cost g_e, we employ the tiebreaker technique to further improve trajectory quality. Specifically, we prefer trajectories similar to the latest optimized trajectory, if available. Let x_n^c denote the current state on the latest optimized trajectory, and the tiebreaker is $g_t(x_c^c) = \|x_n^c - x_c^c\|^2$. Then, the overall heuristic cost is $h_c = g_e + w_{tb} g_t$ where w_{tb} is the weight for the tiebreaker.

We implement an early termination strategy for our search algorithm. This termination condition requires an expanded node to meet two criteria concurrently: (i) near the final target with a minimal margin, and (ii) an optimal trajectory free from any collisions leading to the final goal. Such a technique effectively circumvents potential numerical issues that may arise from the discretization of samples, thereby improving the overall efficiency of the search algorithm.

2) Spatiotemporal Trajectory Optimization

Our TO builds on [27], [28], and we further incorporate the terrain costs into the optimization framework.

Trajectory Parameterization: Suppose that the trajectory consists of $m-1$ intermediate waypoints. Denote the positions of the intermediate path points as $q \triangleq \mathrm{col}\{q_1, \cdots, q_{m-1}\}$ and the related arriving time stamps as $t \triangleq \mathrm{col}\{t_1, \cdots, t_{m-1}\}$. From Theorem 2 of [27], it is known that the optimal trajectory of the multistage control effort minimization problem comprises m quintic polynomials. Then the overall trajectory is parameterized with $c \triangleq \mathrm{col}\{c_1, \cdots, c_m\}$, where c_i is for the parameters of the i-th piece polynomial and can be computed efficiently from q and t.

Problem Formulation and Numerical Optimization: We propose the following optimization model as the TO problem:

$$\min_{q,t} \lambda_t J_{ti} + \lambda_o J_o + \lambda_d J_d + \lambda_h J_h, \tag{13}$$

where $J_\cdot$ denotes the specific cost that will be introduced in the following and $\lambda_\cdot$ is the related weight. In (13), $J_{ti} \triangleq t_m$ is for (C1). Our trajectory parameterization method naturally satisfies (C2). We leverage the virtual collision avoidance force estimation [28] for (C3) to define

$$J_o \triangleq \sum_{i=0}^{m-1} \max\{(d_c - (q_i - s_i)^\top v_i)^3, 0\},$$

where d_c is the safety margin, s_i and v_i are the obstacle surface point and the direction vector for q_i, respectively. The input limit cost is

$$J_d \triangleq \sum_{i=1}^{m-1} \sum_{d \in \{v,a,j\}} \max\{d_i^2 - d_m^2, 0\},$$

where v_i, a_i, j_i are the velocity, acceleration and jerk, and d_m is the corresponding limit. Finally, J_h is a numerical integration of the terrain cost map over the trajectory

$$J_h \triangleq \sum_{i=0}^{m-1} \sum_{k=0}^{N_t} \tilde{h}(q_i) \|p_i^v\| T_k^i,$$

where $\tilde{h}$ is the terrain cost map, $T_k^i \triangleq \frac{1}{N_t}(t_{i+1} - t_i)$ is a time interval in $[t_i, t_{i+1}]$ discretized into N_t pieces.

We only recalculate the high-level trajectory in cases where the planned trajectory is infeasible or the robot is too close to an obstacle. This enhances the stability of the high-level trajectory, while also reducing the frequency of any unnecessary trajectory changes.

IV. Results and Discussion

In this section, we first demonstrate our prescribed-time control based closed-loop trajectories in numerical simulation. Then, we conduct extensive indoor and outdoor experiments to validate the overall planning system.

A. Numerical Simulation

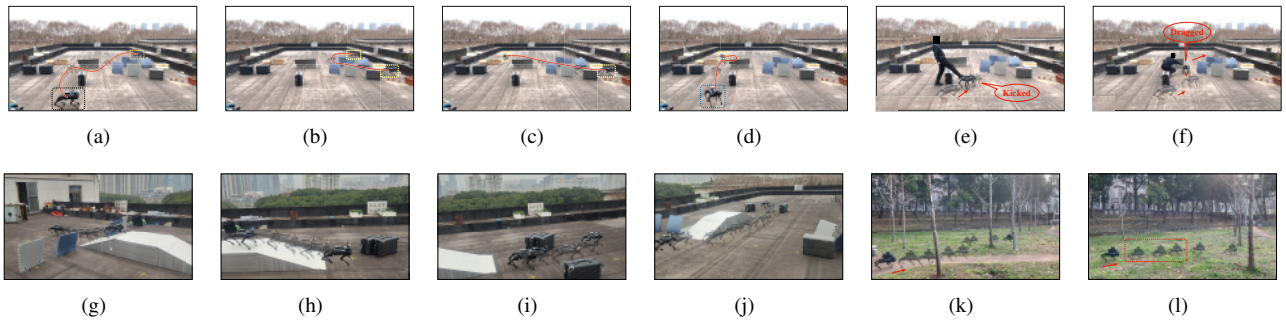
We first show how the closed-loop trajectories vary with 8 different initial positions and orientations, as shown in Fig. 4(a). In this test, we prescribe the convergent time T as 4 seconds, and the other parameters are: $k_r = k_\delta = 0.01, \alpha = m = k_{r_1} = k_{r_2} = k_{\delta_1} = k_{\delta_2} = 1$. The robot can arrive at the target, while automatically aligning its heading and reducing its lateral movement.

Then, we further evaluate the convergence details through a comparative study. In this test, we use (9) as the baseline control law for all $t \in [0, \infty)$. Note that it is a CLF-based control law with only the asymptotic stability guarantee. The initial position is set to $(\sqrt{2}, -\sqrt{2})$, and the controller parameters of both methods are the same as that in the first test. Moreover, we manually add a disturbance when $t = 2$, to show how the controllers could handle this robot deviation automatically. The results in Fig. 4(b) and 4(c) show that both methods are robust to the external disturbance and can guide the robot to the origin. Compared to the baseline, our method converges faster and can rigorously guarantee the prescribed convergent time T, namely, 4 seconds.

B. Robot Validation

We validate the proposed planning system on a quadruped robot Unitree A1 with our customized sensor suites, as shown in Fig. 1. The experimental results are shown in Fig. 3.

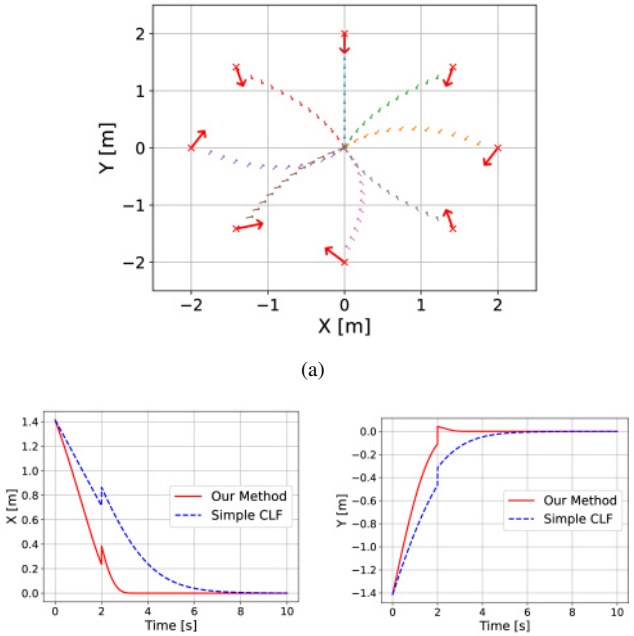
1) Cluttered Indoor Environment: There are multiple static obstacles and hallways in the indoor playground. The objective is to sequentially reach four distinct goal states, each located at the corners of the environment (Fig. 3(a) - 3(d)). During the overall test, the robot can navigate autonomously without any human supervision and reach all targets in sequence.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Fig. 3. Experimental Results. (a)-(d) Results for Test 1. The red lines indicate the trajectory. The black, yellow, and blue dot boxes are for the start, intermediate goals, and end, respectively. (e)-(f) Results for Test 2. (g)-(f) Results for Test 3. (j) Result for Test 4. (k) and (l) Results for Test 5. The red lines highlight the process when the quadruped robot traverses the furrow on the uneven grassland.



(a)

(b) (c)

Fig. 4. Simulation Results. (a) The figure shows 8 different closed-trajectories generated by our prescribed-time control method. The red point and arrow indicate the initial position and heading direction. (b)-(c) The comparative results of our method and a simple CLF-based method. We add a small disturbance into both closed-loop systems when $t = 2$.

2) Navigation with Disturbances: In this scenario, the robot navigates through the same cluttered indoor environment in the experiment before, but is subjected to being kicked and dragged multiple times on its path to the first goal. Fig. 3(e) and 3(f) show that, the robot could quickly recover its posture and maintain balance, illustrating that our navigation framework is robust to significant disturbances.

3) Single-Plank Bridges: The environment consists of a single-plank bridge with up-and-down slopes and obstacles. We set up the environment such that all the feasible trajectories pass through the bridge. In this test, the robot first climbs up the slope (Fig. 3(g)), goes over the single-plank bridge, walks down the slope (Fig. 3(h)), and finally avoids obstacles to reach the target (Fig. 3(i)).

4) Modified Single-Plank Bridge: This scenario is similar to the one before, but several obstacles have been removed to create new feasible trajectories that do not require crossing the bridge. Automatically, the robot bypasses, rather than climbs up, the single-plank bridge, demonstrating that our framework can automatically determine whether to conquer the complex terrains or find a more energy-efficient way.

5) Wild Grassland: The robot navigates through the wild environment containing trees and grass. In particular, the robot can traverse furrows without losing any balance, demonstrating its flexible terrain adaptability in the wild.

V. Conclusion

This paper introduces a reactive planning system composed of a high-level trajectory planner and a prescribed-time control-based reactive planner. In the high-level planner, we utilize kinodynamic A* and spatiotemporal optimization to determine the local subgoal and its optimal time. Subsequently, we propose a feedback control law in the reactive planner, which effectively guides the robot towards the local goal and automatically addresses any deviations that may occur. In particular, we provide theoretical proof for the prescribed-time asymptotic stability of the closed-loop control system within the planned time. The effectiveness of the proposed method is validated through comparative tests in numerical simulations, and the planning system is further tested on a real robot in both indoor and outdoor environments.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 62173155, 52188102, 62225306, U2141235, Program for HUST Academic Frontier Youth Team, the CityU Start-Up Grant (Project No. 9610481), and Chow Sang Sang Group Research Fund sponsored by Chow Sang Sang Holdings International Limited (Project No. 9229076).

REFERENCES

[1] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020.

[2] I. D. Miller, F. Cladera, A. Cowley, S. S. Shivakumar, E. S. Lee, L. Jarin-Lipschitz, A. Bhat, N. Rodrigues, A. Zhou, A. Cohen, *et al.*, "Mine tunnel exploration using multiple quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2840–2847, 2020.

[3] A. S. Aguiar, F. N. dos Santos, J. B. Cunha, H. Sobreira, and A. J. Sousa, "Localization and mapping for robots in agriculture and forestry: A survey," *Robotics*, vol. 9, no. 4, p. 97, 2020.

[4] E. D. Sontag, *Mathematical control theory: deterministic finite dimensional systems*. Springer Science & Business Media, 2013, vol. 6.

[5] J. J. Park and B. Kuipers, "A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4896–4902.

[6] ——, "Feedback motion planning via non-holonomic rrt* for mobile robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4035–4040.

[7] J.-K. Huang and J. W. Grizzle, "Efficient anytime clf reactive planning system for a bipedal robot on undulating terrain," *IEEE Transactions on Robotics*, 2023.

[8] Y. Song, Y. Wang, J. Holloway, and M. Krstic, "Time-varying feedback for regulation of normal-form nonlinear systems in prescribed finite time," *Automatica*, vol. 83, pp. 243–251, 2017.

[9] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1184–1189.

[10] M. Brunner, B. Brüggemann, and D. Schulz, "Hierarchical rough terrain motion planning using an optimal sampling-based method," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5539–5544.

[11] L. Wellhausen and M. Hutter, "Rough terrain navigation for legged robots using reachability planning and template learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6914–6921.

[12] J. Norby and A. M. Johnson, "Fast global motion planning for dynamic legged robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3829–3836.

[13] P. Fernbach, S. Tonneau, A. Del Prete, and M. Taïx, "A kinodynamic steering-method for legged multi-contact locomotion," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3701–3707.

[14] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 295–302.

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

[16] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[17] M. Chignoli, S. Morozov, and S. Kim, "Rapid and reliable quadruped motion planning with omnidirectional jumping," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6621–6627.

[18] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li, *et al.*, "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2132–2139.

[19] S. Paternain, D. E. Koditschek, and A. Ribeiro, "Navigation functions for convex potentials in a space with convex obstacles," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 2944–2959, 2017.

[20] F. Golbol, M. M. Ankarali, and A. Saranli, "Rg-trees: Trajectory-free feedback motion planning using sparse random reference governor trees," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6506–6511.

[21] O. Arslan and D. E. Koditschek, "Sensor-based reactive navigation in unknown convex sphere worlds," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 196–223, 2019.

[22] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[23] A. Polyakov, *Generalized homogeneity in systems and control*. Springer, 2020.

[24] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[25] J. Holloway and M. Krstic, "Prescribed-time output feedback for linear systems in controllable canonical form," *Automatica*, vol. 107, pp. 77–85, 2019.

[26] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE transactions on robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[27] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.

[28] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.