Reactive Locomotion Decision-Making and Robust Motion Planning for Real-Time Perturbation Recovery

Zhaoyuan Gu, Nathan Boyd, and Ye Zhao

Abstract— **In this abstract, we examine the problem of push recovery for bipedal robot locomotion and present a reactive decision-making and robust planning framework for locomotion resilient to external perturbations. Rejecting perturbations is an essential capability of bipedal robots and has been widely studied in the locomotion literature. However, adversarial disturbances and aggressive turning can lead to negative lateral step width (i.e., crossed-leg scenarios) with unstable motions and self-collision risks. These motion planning problems are computationally difficult and have not been explored under a hierarchically integrated task and motion planning method. We explore a planning and decision-making framework that closely ties linear-temporal-logic-based reactive synthesis with trajectory optimization incorporating the robot's full-body dynamics, kinematics, and leg collision avoidance constraints. Between the high-level discrete symbolic decision-making and the low-level continuous motion planning, behavior trees serve as a reactive interface to handle perturbations occurring at any time of the locomotion process. Our experimental results show the efficacy of our method in generating resilient recovery behaviors in response to diverse perturbations from any direction with bounded magnitudes.**

Paper Type – **Recent Work [1]**

I. INTRODUCTION

As legged robots are increasingly deployed in complex environments, the need for robots to accomplish tasks through symbolic planning and decision-making becomes more apparent. Although locomotion robustness has been extensively explored at the motion planning level, resilience to uncertainties and external disturbances at the task planning level has been largely overlooked. Hierarchically integrated task and motion planning (TAMP) is capable of handling logical and whole-body dynamics objectives simultaneously. Unexpected errors or even failures at the lower-level can lead to expensive re-planning at the higher task planning level. On the other hand, high-level discrete task plans can result in infeasible low-level motion plans. With these cascading effects, novel TAMP methods are imperative to make robust locomotion decisions resilient to environmental perturbations and enable robots to efficiently recompute plans at both task and motion planning levels.

At the motion planning level, push recovery of bipedal locomotion has been extensively studied in previous works [2], [3]. Many of these push recovery strategies, however, employ reduced-order models (RoMs) such as inverted pendulum or centroidal momentum model. Challenge arises from solving full-leg collision constraints in these RoMs. Liu et al. [4] demonstrated a complete control framework that considers self-collision under various disturbances, but the framework does not consider more complicated multi-step or non-periodic recoveries. Reactive approaches for high dimensional robots have also been explored [5]–[7], which rely on a distance metric to generate safe repulsive motions. These approaches, however, can lead to significant motion plan discrepancies. In addition, few motion planning strategies incorporate higher-level task planning.

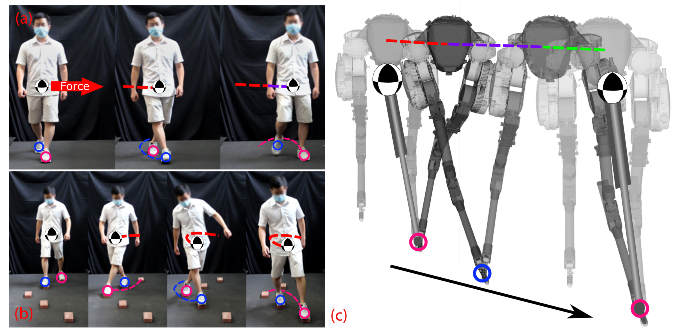

Fig. 1: a) Human is forced to cross legs to recover from an external disturbance. b) Human must execute leg crossing to traverse stepping stones. c) An illustration of recovery motion of bipedal robot Cassie.

For high-level task planning, reactivity is critical to account for environmental changes at runtime. Temporal-logic-based reactive synthesis [8]–[10] has been widely explored to find strategies that generate formally-guaranteed safe and provably correct robot actions in response to environmental events. However, this method has been under-explored for dynamic locomotion problems until recent years. Recent works [11]–[14] adopted linear temporal logic (LTL) to synthesize reactive locomotion navigation plans over rough terrains. Although bipedal walking only involves alternating left and right foot contacts, incorporating external perturbations into formal foot placement decision-making in a provably correct manner remains challenging. Moreover, the feasibility of executing synthesized task plans on high degree-of-freedom legged robots is unexplored. To address these challenges, this study combines collision-avoidance-aware trajectory optimization (TO) with LTL methods to guarantee the task completion of the robot locomotion.

Behavior Trees (BTs), as graphical mathematical models, have been widely explored to schedule autonomous tasks and handle unexpected environmental changes [15], [16]. Their reactive and modular structure can authorize multiple behavioral plans and achieve fault-tolerant task executions [17], [18]. Formal methods [11] only account for perturba-

The authors are with the Laboratory for Intelligent Decision and Autonomous Robots, Woodruff School of Mechanical Engineering, Georgia Institute of Technology. {zgu78, nboyd31, yezhao}@gatech.edu

This work was funded by the NSF grant # IIS-1924978 and Georgia Tech Institute for Robotics and Intelligent Machines Seed Grant.

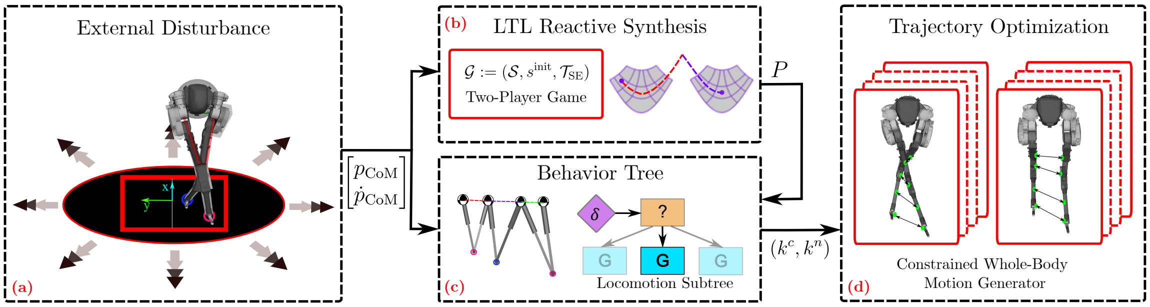

Fig. 2: Block diagram of the proposed framework. a) Experiments of Cassie disturbed during stable walking; b) The high-level task planner synthesis, employing an LTL two-player game; c) The BTs act as a middle layer that reactively executes subtrees based on real-time environmental disturbances; d) A whole-body motion planner is used to generate feasible motions and refine LTL specifications ψ. The high-level task planner and the phase-space planner are integrated in an *online* fashion, as shown by the solid black arrows.

tions applied at specific instances. Handling perturbation at any locomotion phase require further investigation. The BTs naturally handle the continuous environmental perturbations by designing actions online to amend the discrete decision maker.

This study addresses the push recovery problem for legged robots subject to external perturbations that can happen anytime. We propose a combined TAMP framework composed of hierarchical planning layers operating at different temporal and spatial scales (Fig. 2). First, the LTL planning designs safety-guaranteed decisions on keyframe states, including center of mass (CoM) state or foot placements, in response to the keyframe perturbations. When perturbations occur at non-keyframe instants, analytical Riemannian manifolds are used to recalculate a new keyframe transition online for the current walking step. BTs are integrated to allow the updated keyframe states to be any continuous value within the allowable range, instead of a finite set of discrete values quantified in the LTL-based planner. Finally, full-body legged motions are generated using kinodynamic-aware TO for non-periodic multi-step locomotion with self-collision constraints. Compared to our previous robust locomotion work [19], [20], this work (i) studies perturbation recovery from comprehensive perturbations in all directions and during various locomotion phases, and (ii) solves full-body TO to generate dynamically feasible trajectories that refine high-level decisions. The core contributions of this paper are summarized as follows:

- We present a hierarchically integrated LTL-BT TAMP framework for dynamic locomotion that reacts to continuous environmental perturbations for resilient task execution.
- We employ Riemannian manifolds to quantify locomotion keyframe robustness margins and design robust transitions enabled by the reactive task planner.
- We propose a collision-aware, kinodynamic TO that generates collision-free and non-periodic full-body motions and use this TO to refine feasibility specifications in reactive synthesis.

II. Planning Methods

This section details the symbolic decision-making and motion planning framework (Fig. 2). Our hierarchical reactive framework is composed of (i) LTL-level reactive synthesis handling perturbations at keyframe instants, (ii) BT for robust execution of one walking step (OWS) between keyframe instances, (iii) full-body motion primitive generation from kinodynamic-aware TO.

Fig. 2a represents a bipedal robot walking on a perturbing platform. A decision maker (Fig. 2b) plans a multi-step action plan P at each apex instant. The behavior tree in Fig. 2c is executed at every control loop; it modifies the desired keyframe transition based on the instantaneous tracking performance. The modified keyframe transition is the interpolation hyperparameter used to find a full-body trajectory in a set of motion primitives. The motion primitive set and decision maker are created offline.

A. Keyframe-based Non-periodic Locomotion

We separate the entire trajectory into multiple OWS phases that start and end at keyframe states. The keyframe state is defined based on a step-to-step discretization of the continuous walking process, allowing the robot to make CoM apex parameter decisions for each walking step. The i^{th} OWS cycle can be represented by a discrete keyframe transition pair (k^i, k^{i+1}). The keyframe contains the sagittal and lateral CoM apex state, as well as the stance foot index.

Compared to periodic walking where the robot repeats the same motion pattern over multiple steps, keyframe-based walking is non-periodic, which better accommodates rough terrain and environment disturbances.

B. LTL Specifications for Push Recovery

As the complexity of locomotion tasks increases, making safe decisions on keyframe states to recovery from perturbations becomes intricate. To address this challenge, we employ reactive synthesis, which is built upon task specifications and abstractions of dynamical systems [9], [21]. The tasks are represented by LTL specifications, which describe temporal and logical relations of the system properties. The abstraction (i.e., transition system) is a discrete description of the system and environment dynamics. Detailed LTL semantics can be found in [22].

To formally guarantee locomotion task completion under environmental disturbances, we adopt the General Reactivity

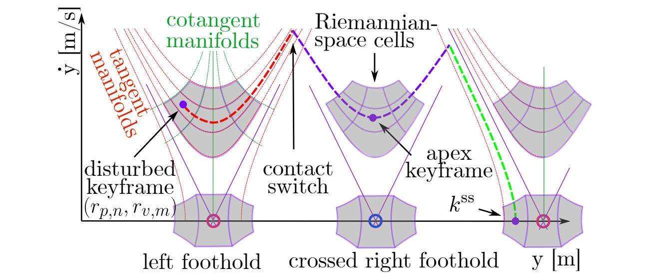

Fig. 3: An illustration of a phase-space Riemannian partition and non-deterministic lateral keyframe transition for disturbance recovery.

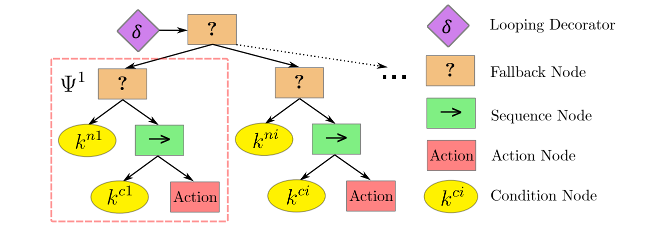

Fig. 4: An illustration of the PABT structure. The PABT groups a set of locomotion subtrees Ψ^i. Each subtree is a fallback tree that encodes a keyframe transition $(k^{c,i}, k^{n,i})$ and a Riemannian recalculation action.

of Rank 1 (GR(1)) [23]. GR(1)—a fragment of LTL—provides correct-by-construction guarantees of the realizability of LTL specifications. Provided a transition system $\mathcal{T}_{\rm SE}$ and LTL specification ψ, the reactive synthesis problem aims for a winning strategy for the robot system such that the execution path satisfies ψ [13]. If the specification is realizable, a decision maker will be constructed and will provide correct actions for any modelled environmental events.

The transition system discretizes the continuous state space (i.e., robot's CoM phase space near the apex state) into Riemannian partitions. The Riemannian partitions are defined for both sagittal and lateral phase-space, each constitutes 12 cells. The Riemannian partitions use the analytical manifolds of CoM dynamics derived from the Prismatic Inverted Pendulum Model (PIPM), which discretizes the phase-space with tangent and cotangent locomotion manifolds, instead of using naïve Euclidean-type discretization. The tangent and cotangent manifolds comply with the PIPM locomotion dynamics and provide an intuitive trajectory recalculation strategy for potential CoM deviation. The discretization of the phase-space plane allows the LTL to define specifications and make symbolic decisions.

The system takes actions $a_{\rm sys}$ to decide the next keyframe state k^n. The environment action is $p_{\rm env}$ that pushes the system from the current keyframe state k^c to a specific Riemannian cell center. In the task planner, we assume that the environment action is a perturbation applied only at a keyframe instant. The perturbation induces a CoM position and velocity jump after applying an external force to the robot's pelvis frame. The system action $a_{\rm sys}$ and environment action $p_{\rm env}$ together decide the next apex keyframe state $k^n = \mathcal{T}_{\rm SE}(k^c, a_{\rm sys}, p_{\rm env})$.

We define steady state keyframes $\mathcal{K}^{\rm ss}$ as the apex states during perturbation-free walking. The lateral velocity of a steady state keyframe $k^{\rm ss}$ is zero. The specifications of the system are defined as follows:

- The robot chooses to maintain stable walking so long as there is no perturbation from the environment.
- In the presence of perturbations, the keyframe state returns to a steady state within two steps.
- The transition is feasible and verified by a low-level full-body TO (Sec. II-E).
- For the recovery motion execution not to be interrupted, we assume the environment perturbation happens at most once every two steps.
- The policy tries to decrease its lateral velocity from k^c to k^n, if it cannot recover to a steady state $k^{\rm ss}$ within one step.

C. Task Planner Synthesis

Given the LTL specifications above, the task planner models the robot system and the environment as two agents. The two agents interplay in a two-player game.

At each keyframe instant, the decision maker uses the estimated current system keyframe state k^c and plans a sequence of transitions until the final state $k^f = k^{\rm ss}$. The action roll-out produces an action plan $P = \{k^c, \dots, k^f\}$.

D. Behavior-Tree-Based Dynamic Replanning

To address continuous perturbations at non-keyframe instants, we propose a perturbation-aware behavior tree (PABT) that online modifies the desired keyframe transition $(k^{c,d}, k^{n,d})$. The PABT complements the reactive synthesis by locally modifying the keyframe transitions, given the real-time captured CoM state $[p_{\rm CoM}; \dot{p}_{\rm CoM}]$.

The PABT groups a set of locomotion subtrees $\Psi = \bigcup\limits_i \Psi^i$. Each Ψ^i encodes a pair of the current-to-next keyframe states $(k^{c,i}, k^{n,i})$. These pairs are represented as condition nodes in the locomotion subtrees (Fig. 4). The locomotion subtrees are fallback BTs that execute their action nodes when the desired keyframe transition from the high-level matches their condition nodes.

The PABT modifies its keyframe transitions locally to handle non-keyframe perturbations. Here we use the recovery strategy [19] to perform a Riemannian recalculation. When the CoM state is perturbed off from the nominal manifold, the recovery strategy computes a new trajectory from the current state to an updated desired keyframe, which still follows the LIPM locomotion dynamics.

The PABT grows when the new action plan P is commanded from the task planner. The PABT constructs new subtrees that represent each new transition (k^c, k^n) from P.

E. Collision-Aware Kinodynamic Trajectory Optimization

The task planner and PABTs generate keyframe transitions robust to perturbations. However, mapping the transitions to whole-body trajectories in real-time often poses a challenge due to the curse of dimensionality. To address this, we use TO to create a set of motion primitives offline. The TO generates desired motions that satisfy the physical constraints while minimizing the trajectory cost [24]–[26]. The TO is

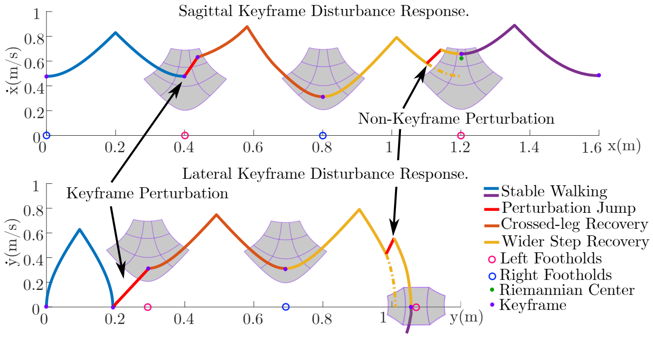

Fig. 5: Lateral and sagittal responses to diagonal disturbances at keyframe and non-keyframe instants while walking at 0.5 m/s apex velocity. Each color represents a single step generated by the LTL-BT.

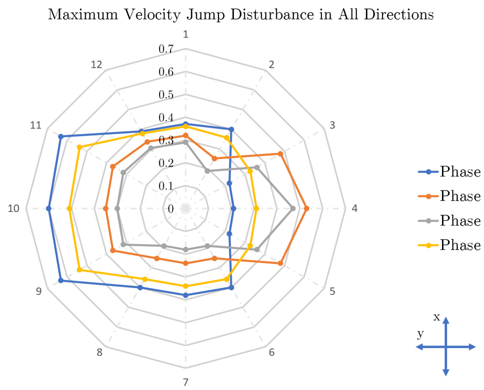

Fig. 6: Maximally allowable velocity change exerted on the CoM for a single step at 30° increments. The perturbation happens at different phases during a right leg stance. Values on the left half resulted in single wider step recoveries, and values on the right half require crossed-leg maneuvers.

also used as a verification to check the feasibility of high-level keyframe transitions. More detailed constraint setting can be found in [1].

The proposed TO is different from the existing state-of-the-art methods. Compared to [25], we exclude motions with leg collisions using distance-based constraints, increasing the safety of the feasible motions. Compared to [27], our TO considers the full-body dynamics of the bipedal robot with more aggressive motions (e.g., perturbed leg crossing), providing a more accurate gauge for dynamic feasibility.

III. Results

To demonstrate the robustness of the proposed methods, we tested various scenarios in Matlab simulation with a bipedal robot, Cassie. The full dynamics is modelled by the Simscape engine. The NLP solver IPOPT [28] solved the TO problems. Our framework, together with a virtual constraint controller [29], ran at a rate of 2 kHz online. Perturbation was detected by filtering the CoM velocity estimation. We used SLUGS reactive synthesis toolbox [30] to design LTL specifications and synthesize the decision maker.

We evaluated the performance of our framework through multiple push recovery studies. As shown in Fig. 5, the system was capable of composing multiple OWS trajectories. The robot was firstly disturbed to the non-apex velocity $(\dot{x}, \dot{y}) = (0.63, 0.31)$ m/s at keyframe instant. The keyframe decision maker planned a two-step recovery strategy (one crossed-leg step and one succeeding wider step) to come back to a steady state $k^{\rm ss}$. Disturbances at non-keyframe states required the robot to recalculate a new CoM trajectory to an updated keyframe state. The PABT locally modified the desired keyframe transition and allowed the transitions to start and terminate in non-Riemannian-cell-centers. The reactive synthesis could update the keyframe transitions as long as the CoM state was inside the Riemannian robustness bound (grey areas in Fig. 5).

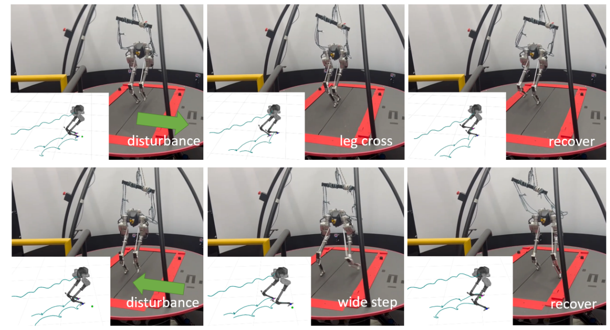

Fig. 7: We tested the framework on CAREN. CAREN moves omnidirectionally and provides precise perturbation during periodic walking. We perturbed in 12 directions and selected 2 trials (left and right perturbations) to represent the results.

In Fig. 6, we compared the maximum impulse velocity changes the system can recover from in 12 directions during OWS in simulation. The robot walked sagittally at 0.5 m/s apex velocity. After the perturbation, it recovers using two steps. When the push direction was lateral left, the robot would take a wider step to come back at $k^{\rm ss}$; otherwise, when the push direction was lateral right, the robot needed to adopt the crossed-leg maneuvers. The perturbations are applied at four different phases, with phases $\phi = 0\%$ and 90% closer to keyframe states (boundary phases), and $\phi = 30\%$ and 60% closer to the contact switch phase (50%). The asymmetry of the maximum allowable disturbances in the lateral directions can be attributed to the more constrained kinematic workspace of the swing legs in the crossed-leg scenario.

Fig. 7 shows the Cassie recovery from perturbation on a Computer Aided Rehabilitation system (CAREN). We exerted disturbances in 12 directions with a maximum of 0.5 m/s peak velocity during periodic walking. The top row represents a rightward platform move induced leg crossing recovery, and the bottom row shows a recovery maneuver by taking a wider step.

IV. Conclusions

In this paper, we presented a locomotion framework for reactive disturbance rejection at the symbolic decision-making and continuous motion planning level. We combined reactive synthesis with BTs to demonstrate safe, continuous, disturbance rejection capabilities.

At the low level, the TO generates full-body locomotion trajectories and refines feasible keyframe specifications in the reactive synthesis to fill the gap between the high-level decisions making and the low-level motion planning.

REFERENCES

[1] Z. Gu, N. Boyd, and Y. Zhao, "Reactive locomotion decision-making and robust motion planning for real-time perturbation recovery," in *IEEE International Conference on Robotics and Automation*, 2022.

[2] B. J. Stephens, "Push recovery control for force-controlled humanoid robots," Ph.D. dissertation, Carnegie Mellon University, 2011.

[3] P.-b. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 137–142.

[4] C. Liu, J. Ning, K. An, and Q. Chen, "Active balance of humanoid movement based on dynamic task-prior system," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, 2017.

[5] C. Zhou, C. Fang, X. Wang, Z. Li, and N. Tsagarakis, "A generic optimization-based framework for reactive collision avoidance in bipedal locomotion," in *IEEE International Conference on Automation Science and Engineering*, 2016, pp. 1026–1033.

[6] A.-C. Hildebrandt, R. Wittmann, D. Wahrmann, A. Ewald, and T. Buschmann, "Real-time 3d collision avoidance for biped robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4184–4190.

[7] A. Dietrich, T. Wimböck, H. Täubig, A. Albu-Schäffer, and G. Hirzinger, "Extensions to reactive self-collision avoidance for torque and position controlled humanoids," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3455–3462.

[8] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[9] J. Liu, N. Ozay, U. Topcu, and R. M. Murray, "Synthesis of reactive switching protocols from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1771–1785, 2013.

[10] K. He, A. M. Wells, L. E. Kavraki, and M. Y. Vardi, "Efficient symbolic reactive synthesis for finite-horizon tasks," in *International Conference on Robotics and Automation*, 2019, pp. 8993–8999.

[11] Y. Zhao, Y. Li, L. Sentis, U. Topcu, and J. Liu, "Reactive task and motion planning for robust whole-body dynamic locomotion in constrained environments," *The International Journal of Robotics Research, In Press*, 2022.

[12] S. Kulgod, W. Chen, J. Huang, Y. Zhao, and N. Atanasov, "Temporal logic guided locomotion planning and control in cluttered environments," in *American Control Conference*, 2020, pp. 5425–5432.

[13] J. Warnke, A. Shamsah, Y. Li, and Y. Zhao, "Towards safe locomotion navigation in partially observable environments with uneven terrain," in *IEEE Conference on Decision and Control*, 2020, pp. 958–965.

[14] A. Shamsah, J. Warnke, Z. Gu, and Y. Zhao, "Integrated task and motion planning for safe legged navigation in partially observable environments," *arXiv preprint arXiv:2110.12097*, 2021.

[15] A. Marzinotto, M. Colledanchise, C. Smith, and P. Ögren, "Towards a unified behavior trees framework for robot control," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 5420–5427.

[16] S. Li, D. Park, Y. Sung, J. A. Shah, and N. Roy, "Reactive task and motion planning under temporal logic specifications," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 12 618–12 624.

[17] M. Colledanchise and P. Ögren, *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.

[18] M. Iovino, E. Scukins, J. Styrud, P. Ögren, and C. Smith, "A survey of behavior trees in robotics and ai," *arXiv preprint arXiv:2005.05842*, 2020.

[19] Y. Zhao, B. R. Fernandez, and L. Sentis, "Robust optimal planning and control of non-periodic bipedal locomotion with a centroidal momentum model," *The International Journal of Robotics Research*, vol. 36, no. 11, pp. 1211–1242, 2017.

[20] Y. Zhao, B. R. Fernandez, and L. Sentis, "Robust phase-space planning for agile legged locomotion over various terrain topologies," in *Robotics: Science and Systems*, vol. 12, 2016.

[21] H. Kress-Gazit, T. Wongpiromsarn, and U. Topcu, "Correct, reactive, high-level robot control," *IEEE Robotics & Automation Magazine*, vol. 18, no. 3, pp. 65–74, 2011.

[22] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[23] N. Piterman, A. Pnueli, and Y. Sa'ar, "Synthesis of reactive(1) designs," in *Verification, Model Checking, and Abstract Interpretation*. Springer, 2006, pp. 364–380.

[24] A. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, 01 2010.

[25] A. Hereid and A. D. Ames, "Frost: Fast robot optimization and simulation toolkit," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 719–726.

[26] M. Koptev, N. Figueroa, and A. Billard, "Real-time self-collision avoidance in joint space for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1240–1247, 2021.

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

[28] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 03 2006.

[29] Y. Gong, *et al.*, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *American Control Conference*, 2019, pp. 4559–4566.

[30] R. Ehlers and V. Raman, "Slugs: Extensible gr(1) synthesis," in *International Conference on Computer Aided Verification*, 2016, pp. 333–339.