Robust Locomotion Navigation in Partially Observable Environments with Safety Guarantees

Jonas Warnke[1], Abdulaziz Shamsah[1], Yingke Li[1,2], Samuel Coogan[2], and Ye Zhao[1]
{jwarnke, ashamsah3}@gatech.edu (first two authors equally contributed)
[1]Laboratory for Intelligent Decision and Autonomous Robots, Woodruff School of Mechanical Engineering,
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA

***Abstract*—This study is working towards an integrated task and motion planning method for dynamic locomotion in partially observable environments with safety guarantees. This planning framework is composed of a symbolic task planner and a reduced-order-model-based motion planner, which are connected by a mid-level keyframe decision-maker as seen in Fig. 2. The mid-level keyframe decision maker generates a keyframe plan via reachability analysis and proposes a robust keyframe policy, which is used to generate low-level phase-space trajectories. The high-level task planner employs a linear temporal logic approach for a reactive game synthesis between the robot and its environment while incorporating the robust keyframe transition policies into the formal task specification design. A belief abstraction method in the task planner enables belief estimation of dynamic obstacle locations and guarantees safe locomotion with collision avoidance.**

I. Introduction

Safety scalable to high-dimensional robotic systems becomes imperative as legged robots maneuver over uneven and unpredictable environments (see Fig. 1). In the robot mobility field, navigation safety is conventionally studied from the collision avoidance perspective [4, 6, 1]. However, in the context of dynamic legged locomotion, maintaining dynamic balancing, i.e., avoiding a fall [7, 5], becomes an essential safety criterion. Reasoning about safety from both levels has been largely under-explored in the field with a few exceptions [3, 9]. Our method takes one step towards using a symbolic planning method to design active navigation decisions with safety guarantees. Meanwhile, we incorporate a belief abstraction approach to enable safe navigation in a partially observable environment.

Hierarchical planning structure: We employ a high-level, temporal-logic-based symbolic planner to generate a locomotion action set $\boldsymbol{a} \in \mathcal{A}$ at each robot keyframe $\boldsymbol{s}$ (more details in later sections). Taking this action $\boldsymbol{a}$ and the current keyframe state $\boldsymbol{s}_c$, the mid-level keyframe decision-maker outputs the next-walking-step keyframe $\boldsymbol{s}_n$ for a robust transition. Finally, the low-level motion planner generates one-walking-step locomotion trajectory based on the keyframe and a Prismatic Inverted Pendulum Model (PIPM). The integrated, hierarchical task and motion planning framework is shown in Fig. 2.

II. LOCOMOTION PLANNING

Phase-space planning (PSP) is a general planning framework for dynamic legged locomotion over highly rough terrain [15, 13, 14]. This planning method is based on a reduced order model (PIPM). This PIPM plans center-of-mass (CoM) trajectories with varying heights, which is suitable for rough terrain locomotion. The PIPM dynamics are shown in Appendix A.

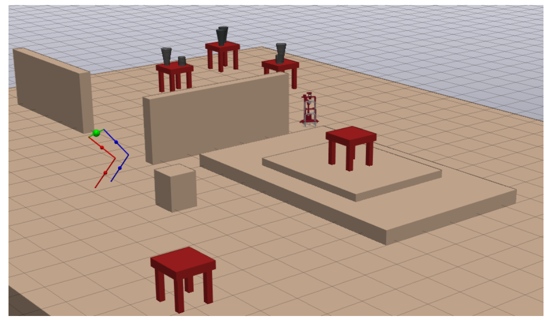


Fig. 1: A conceptual illustration of safe locomotion navigation in a cluttered environment with dynamic mobile obstacles and uneven terrain in Drake [12].

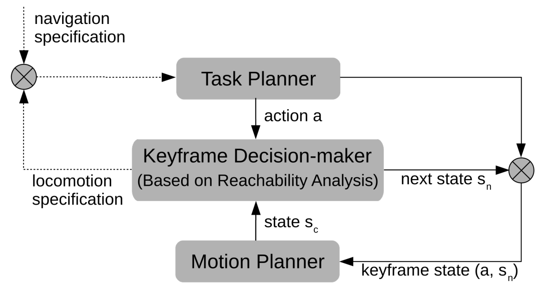


Fig. 2: Block diagram of the proposed locomotion planning framework.

Keyframe Planning: Our PSP method focuses on using keyframe states to capture essential locomotion dynamics and makes discrete decisions to achieve non-periodic gaits. This study generalizes the original keyframe definition in [15] by introducing diverse navigation actions in 3D environments.

Definition 2.1 (Keyframe State): The keyframe state of our PIPM model is defined as $\boldsymbol{k} = (\boldsymbol{a}, \boldsymbol{s}) \in \mathcal{K}$, where $\boldsymbol{a} = (d, \Delta\theta, \Delta z_{\text{foot}}, i_{\text{st}}, c_{\text{forward}}, c_{\text{stop}}) \in \mathcal{A}$ is an action, d is the walking step length, $\Delta\theta$ is the heading angle, Δz_{foot} is the step height, i_{st} is the desired stance foot index, c_{fwd} and c_{stop} are boolean variables representing step forward or stop respectively, $\boldsymbol{s} = (v_{\text{apex}}, z_{\text{apex}}) \in \mathcal{S}$, is the state of the CoM sagittal apex velocity v_{apex} and the apex CoM height z_{apex} in the global frame.

Robust Keyframe Decision Making: Our decision-making objective is to design robust keyframe transitions in the presence of external disturbances. To this end, we use reachability

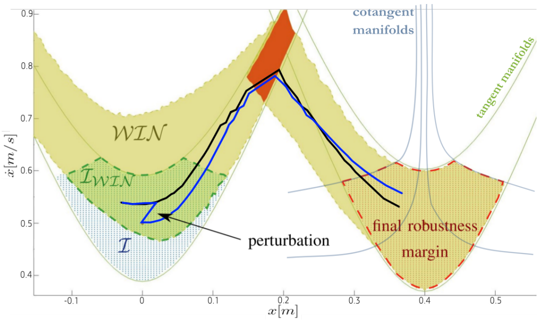


Fig. 3: Phase-space reachability region computation of one locomotion step. The green shaded area is $I_{\mathcal{WIN}}$, i.e., the intersection of initial robustness margin set $\mathcal{I}$ (blue region) and the winning set $\mathcal{WIN}$ (yellow region). The black and blue lines are simulated CoM trajectories under bounded perturbations. The tangent and cotangent manifolds quantify the size of the robustness margins, i.e., $\mathcal{I}$ and the final robustness margin (dashed red region).

analysis to compute feasible alternative keyframe transitions when the nominal transition becomes infeasible under the perturbations. As an initial work, we leverage the reachability analysis method in [16] to investigate the robust keyframe transitions. Using RObustly Complete control Synthesis (ROCS) [8], we compute winning sets $\mathcal{WIN}$ [1] for feasible keyframe transitions. To quantify the robustness, we propose and use the initial winning set $\mathcal{I}_{\mathcal{WIN}}$ defined as below.

Definition 2.2 (Initial Winning Set): $\mathcal{I}_{\mathcal{WIN}}$ is the intersection set of the initial robust margin set $\mathcal{I}$ and $\mathcal{WIN}$, i.e., $\mathcal{I}_{\mathcal{WIN}} = \mathcal{WIN} \cap \mathcal{I}$. [2]

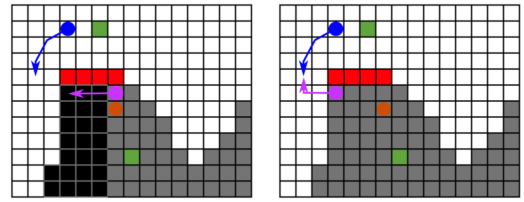
If the initial winning set $\mathcal{I}_{\mathcal{WIN}} \neq \emptyset$, we define the associated keyframe transition as viable. This viable transition is used as a criterion to achieve dynamic balancing safety. Fig. 3 shows the reachability region computation for one downstairs walking step with $\Delta z_{\text{foot}} = -0.2$ m. By examining the size of $\mathcal{I}_{\mathcal{WIN}}$ in the cases of rough terrain and turning heading angles, we observe a inverse relation between the size of $\mathcal{I}_{\mathcal{WIN}}$ and the value of v_{apex} (see Appendix B). Using the size of $\mathcal{I}_{\mathcal{WIN}}$ as an indicator of robustness, we devise a heuristic-based robust keyframe planning policy in the middle-level keyframe decision-maker. For instance, for traversing a rough terrain or steering, the apex velocity is decreased according to the terrain height variation or heading angle change.

III. Task Planning

The task planner is responsible for achieving robot navigation goals while guaranteeing collision avoidance. To expand the set of safe actions the task planner can choose, we employ belief tracking of the dynamic obstacle. We devise a variant of the approach in [2] to design a belief-based navigation strategy in a 2D grid world. The grid is split up into coarse belief regions that are used to reason about the obstacle's possible location when it is out of sight. At each keyframe, the task planner evaluates the robot's discrete location ($l_r \in \mathcal{L}$) and heading angle ($h_r \in \mathcal{H}_r$) on the grid, as well as it's belief of the dynamic obstacle location ($b_o \in \mathcal{B}_o$). b_o takes a real value in $\mathcal{L}$ when the obstacle is visible, and a value indexing a set of belief regions when the obstacle is out of sight.

[1] $\mathcal{WIN}$ denotes the keyframe state set satisfying reachability conditions.

[2] The initial robust margin set $\mathcal{I}$ is defined as a robustness region centered around the nominal initial keyframe state [16].



(a) With belief abstraction (b) Without belief abstraction

Fig. 4: A snapshot of a 2D navigation simulation where the robot (blue circle) is going between the two goal states (green cells), while avoiding a static obstacle (red cells) and a dynamic obstacle (orange circle). White cells are visible while grey and black cells are non-visible. Gray cells represent the planner's belief of potential obstacle locations based on the obstacle's last known state. The closest the obstacle could be to the robot, as believed by the planner, is depicted by the pink circle.

To formally guarantee that the robot reaches goal locations *infinitely often* while meeting safety specifications, we use SLUGS [11] to synthesize a planner from General Reactivity of Rank 1 (GR(1)) specifications [10].

To synthesize a winning navigation strategy, a game structure is proposed as $\mathcal{G}_{\text{belief}} := (\mathcal{S}_{\text{belief}}, s^{\text{init}}_{\text{belief}}, \mathcal{T}_{\text{belief}})$ with

- $\mathcal{S}_{\text{belief}} = \mathcal{L} \times \mathcal{B}_o \times \mathcal{H}_r \times \mathcal{A}$ is the augmented state;
- $s^{\text{init}}_{\text{belief}} = (l^{\text{init}}_r, \{b^{\text{init}}_o\}, h^{\text{init}}_r, a^{\text{init}})$ is the initial game state known a priori;
- $\mathcal{T}_{\text{belief}} \subseteq \mathcal{S}_{\text{belief}} \times \mathcal{S}_{\text{belief}}$ are possible transitions in the belief game where $((l_r, b_o, h_r, a), (l'_r, b'_o, h'_r, a'_r)) \in \mathcal{T}_{\text{belief}}$;

We generate the transition system $\mathcal{T}_{\text{belief}}$ by defining specifications ψ for how the augmented state can evolve. We capture motion planning constraints and enforce collision avoidance guarantees within the transition specifications. Additionally, specifications are automatically generated to govern how the belief evolves when the dynamic obstacle is out of sight, and where the obstacle can re-appear. More details on specification generation are in Appendix C.

The task planner models the robot and environment interplay as a two-player turn-based game between the robot and a possibly adversarial environment. The synthesized high-level strategy guarantees that the robot will always win this game.

Fig. 4 depicts a snapshot of a simulation where the planner is only able to safely initiate a turn when employing belief reasoning, as without tracking possible non-visible locations of the dynamic obstacle, the task planner can not guarantee collision avoidance before the robot can safely stop. This comparison underlines the significance of belief tracking.

IV. Conclusion

The proposed task and motion planning framework generates locomotion trajectories that safely pilot the robot through a partially observable environment. This represents a step towards guaranteeing robust safe locomotion in complex environments with formal safety guarantees. In future work, we plan to integrate the robust reachability analysis into the phase-space planning and embed the computed feasible keyframe transitions into the task specification design.

References

[1] Andrea Bajcsy, Sylvia L Herbert, David Fridovich-Keil, Jaime F Fisac, Sampada Deglurkar, Anca D Dragan, and Claire J Tomlin. A scalable framework for real-time multi-robot, multi-human collision avoidance. In *International Conference on Robotics and Automation*, pages 936–943, 2019.

[2] Suda Bharadwaj, Rayna Dimitrova, and Ufuk Topcu. Synthesis of surveillance strategies via belief abstraction. In *IEEE Conference on Decision and Control*, pages 4159–4166. IEEE, 2018.

[3] Néstor Bohórquez, Alexander Sherikov, Dimitar Dimitrov, and Pierre-Brice Wieber. Safe navigation strategies for a biped robot walking in a crowd. In *IEEE-RAS International Conference on Humanoid Robots*, pages 379–386, 2016.

[4] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[5] Steve Heim and Alexander Spröwitz. Beyond basins of attraction: Quantifying robustness of natural dynamics. *IEEE Transactions on Robotics*, 35(4):939–952, 2019.

[6] Michael Hoy, Alexey S Matveev, and Andrey V Savkin. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, 33(3):463–497, 2015.

[7] Twan Koolen, Tomas De Boer, John Rebula, Ambarish Goswami, and Jerry Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The international journal of robotics research*, 31(9):1094–1113, 2012.

[8] Yinan Li and Jun Liu. ROCS: A robustly complete control synthesis tool for nonlinear dynamical systems. In *Proceedings of International Conference on Hybrid Systems: Computation and Control*, pages 130–135, 2018.

[9] Adrien Pajon and Pierre-Brice Wiebe. Safe 3d bipedal walking through linear mpc with 3d capturability. In *International Conference on Robotics and Automation*, pages 1404–1409. IEEE, 2019.

[10] Nir Piterman, Amir Pnueli, and Yaniv Sa'ar. synthesis of reactive(1) designs. In E. Allen Emerson and Kedar S. Namjoshi, editors, *Verification, Model Checking, and Abstract Interpretation*, pages 364–380. Springer, 2006.

[11] V. Raman R. Ehlers. slugs: Extensible gr(1) synthesis, 2016.

[12] Russ Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019. URL https://drake.mit.edu.

[13] Ye Zhao and Luis Sentis. A three dimensional foot placement planner for locomotion in very rough terrains. In *IEEE-RAS International Conference on Humanoid Robots*, pages 726–733. IEEE, 2012.

[14] Ye Zhao, Ufuk Topcu, and Luis Sentis. High-level planner synthesis for whole-body locomotion in unstructured environments. In *IEEE Conference on Decision and Control*, pages 6557–6564, 2016.

[15] Ye Zhao, Benito R Fernandez, and Luis Sentis. Robust optimal planning and control of non-periodic bipedal locomotion with a centroidal momentum model. *The International Journal of Robotics Research*, 36(11):1211–1242, 2017.

[16] Ye Zhao, Yinan Li, Luis Sentis, Ufuk Topcu, and Jun Liu. Reactive task and motion planning for robust whole-body dynamic locomotion in constrained environments. *arXiv preprint arXiv:1811.04333*, 2018.

Acknowledgments

The authors would like to thank the help from Yinan Li and Jun Liu on their support of the reachability analysis implementation, and Suda Bharadwaj and Ufuk Topcu for the discussion on belief space planning. This work was partially funded by the NSF grant # IIS-1924978.

Appendix

A. Prismatic Inverted Pendulum Model

The prismatic inverted pendulum model constrains the CoM path to a piece-wise linear surface [15]. The centroidal dynamics are written as

$$\ddot{p}_{\text{com}} = \begin{pmatrix} w_q^2(x - x_{\text{foot}_q}) \\ w_q^2(y - y_{\text{foot}_q}) \\ a_q w_q^2(x - x_{\text{foot}_q}) + b_q w_q^2(y - y_{\text{foot}_q}) \end{pmatrix} \tag{1}$$

where $p_{\text{com}} = (x, y, z)^T$ is the CoM position, $w_q = \sqrt{g/h_{\text{apex}}}$ and $h_{\text{apex}} = a_q \cdot x_{\text{foot}_q} + b_q \cdot y_{\text{foot}_q} + h$ is the relative apex height of the CoM with respect to the foot position. h is a bias constant equal to 0.8 m, while a_q and b_q are slopes used to adjust CoM path surface to follow the slopes of the terrain [13]. x_{foot} ,y_{foot}, and z_{foot} are the foot placement positions at the q^{th} step.

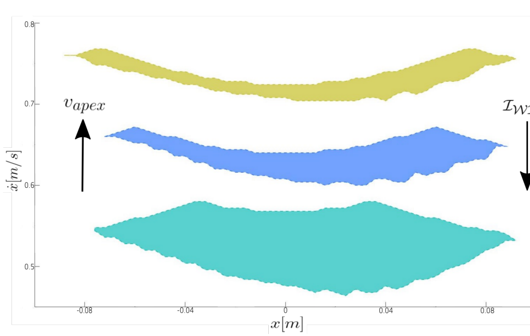


Fig. 5: Initial winning set variation with respect to different apex velocities. Three $\mathcal{I_{WIN}}$ sets for one walking step with $\Delta z_{\text{foot}} = -0.2$ m at three different v_{apex}. Teal shaded region is $\mathcal{I_{WIN}}$ for $v_{\text{apex}} = 0.5$ m/s, blue shaded region is $\mathcal{I_{WIN}}$ for $v_{\text{apex}} = 0.6$ m/s, and yellow shaded region is $\mathcal{I_{WIN}}$ for $v_{\text{apex}} = 0.7$ m/s.

B. Initial Winning Set Variation for Different Apex Velocities

The robustness of a keyframe transition depends on the specific keyframe values. To quantify this robustness, we use the size of $\mathcal{I}_{\mathcal{WIN}}$ as an indicator. Fig. 5 shows the relationship between v_{apex} and the size of $\mathcal{I}_{\mathcal{WIN}}$ when the robot goes down a stair step of 0.2 m height. While it is possible to traverse the step with $v_{\text{apex}} = 0.6$ m/s, the size of $\mathcal{I}_{\mathcal{WIN}}$ increases when reducing v_{apex} from 0.6 m/s to 0.5 m/s, which corresponds to a more robust transition. Examining $\mathcal{I}_{\mathcal{WIN}}$ for three different values of apex velocities we observe a trend in its size. The number of uniformly sampled states in the $\mathcal{I}_{\mathcal{WIN}}$ region for $v_{\text{apex}} = 0.5$ m/s, 0.6 m/s, and 0.7 m/s is 769, 358, and 270, respectively.

C. Belief Space Reasoning

To leverage belief space reasoning for collision avoidance, it is necessary that the belief over-approximates the possible dynamic obstacle locations. We automatically generate transition rules for the allowable belief state b_o. In the reactive synthesis, b_o is modeled as an environment state, allowing transition rules to be formulated as environment safety specifications. As part of the turn-based nature of the game, we allow the obstacle to move one cell in any cardinal direction per walking step. When the dynamic obstacle is at the boundary of a visible range, b_o can transition to any adjacent visible cell, or to a belief index, if the obstacle is not visible at the next keyframe. This belief index represents the set of belief regions the obstacle could have entered. Once out of sight, the obstacle is believed to be in any cell represented by the belief index. Therefore, b'_o must represent the current belief region set plus any adjacent belief regions. Using this method, we can guarantee that the obstacle location is always in the belief region set, i.e. the belief is an over-approximation of the possible obstacle locations. With the ability of reasoning about what non-visible cells the obstacle could be in, the planner is able to determine where the obstacle can reenter the visible range. As shown in Fig. 4, the ability to assure that the obstacle will not appear in front of the robot can result in safe navigation behavior that would otherwise not be guaranteed.

D. Safety Guarantees and Reasoning about Conservativeness

Guaranteeing safe navigation behavior is imperative for locomotion task planning. To guarantee collision avoidance in an environment with dynamic obstacles, the task planner needs to anticipate any possible move that the obstacle can make. Guaranteeing safety in a partially observable environment results in overly conservative behavior, as the planner accounts for the possibility that the dynamic obstacle might appear in front of the robot at any time. Fig. 4b shows a scenario where the planner can not guarantee collision avoidance over a multi step turn if the obstacle can appear in front of the robot. Belief space planning eliminates this conservative behavior when the belief indicates the obstacle can not appear in front of the robot as can be seen in Fig. 4a. The conservative behavior may still exist because the belief is an over-approximation of where the dynamic obstacle could be, i.e. the planner may believe that the obstacle could appear in front of the robot at the next step, when in reality the obstacles could not have moved that far, given where it was last observed. The belief tracking accuracy can be improved by reducing the size of each belief region, however this increases the computational complexity, as the number of belief regions increases. A compromise between overly conservative behavior and computational complexity is made.

From the reachability analysis standpoint, the reachability controllers are synthesized using a finite-abstraction based method that over-approximates the dynamics of the continuous system. In the future, we will use adaptive partitions of the abstract intervals around the keyframe states to approximate the reachable set with a fine granularity while maintaining the computational burden to be tractable. Meanwhile, non-uniform grids of the robustness margins via the use of locomotion manifolds will be explored to reduce conservatism by reducing the number of unnecessary keyframe transitions, which can be tremendous if a small granularity is used.