

SBAMP: Sampling Based Adaptive Motion Planning

Quan A. Pham, Kabir Ram Puri, Shreyas Raorane
General Robotics, Automation, Sensing, & Perception (GRASP) Lab,
University of Pennsylvania,
Philadelphia, PA, USA
Emails: {quanpham, kapuri, raorane}@seas.upenn.edu

Abstract—Autonomous robotic systems must navigate complex, dynamic environments in real-time, often facing unpredictable obstacles and rapidly changing conditions. Traditional sampling-based methods, such as RRT*, excel at generating collision-free paths but struggle to adapt to sudden changes without extensive re-planning. Conversely, learning-based dynamical systems, such as the Stable Estimator of Dynamical Systems (SEDS), offer smooth, adaptive trajectory tracking but typically rely on pre-collected demonstration data, limiting their generalization to novel scenarios. This paper introduces Sampling-Based Adaptive Motion Planning (SBAMP), a novel framework that overcomes these limitations by integrating RRT* for global path planning with a SEDS-based local controller for continuous, adaptive trajectory adjustment. Our approach requires no pre-trained datasets and ensures smooth transitions between planned waypoints, maintaining stability through Lyapunov-based guarantees. We validate SBAMP in both simulated environments and real hardware using the RoboRacer platform, demonstrating superior performance in dynamic obstacle scenarios, rapid recovery from perturbations, and robust handling of sharp turns. Experimental results highlight SBAMP’s ability to adapt in real-time without sacrificing global path optimality, providing a scalable solution for dynamic, unstructured environments.

Index Terms—RRT*, Dynamical Systems, Stable Estimator of Dynamical System (SEDS), Adaptive Navigation; Real-Time Replanning; Lyapunov Stability; Autonomous Navigation

I. INTRODUCTION

Autonomous robots must navigate environments that are both geometrically complex and dynamically changing, such as dodging pedestrians on crowded sidewalks, rerouting around fallen debris in a warehouse, weaving through moving obstacles on a factory floor, or recovering from large perturbations (e.g., sudden collisions or strong external pushes) that displace them far from their planned trajectory. A practical motion planner therefore faces two competing demands:

- *Global path quality*: computing near-optimal, collision-free trajectories over long horizons; and
- *Local reactivity*: instantaneously adapting to newly observed obstacles or goal perturbations.

Sampling-based planners like RRT* [1] guarantee asymptotic optimality in static scenes but incur significant overhead when replanning from scratch under change. Reactive controllers, such as SEDS [2] and LPV-DS [3], offer smooth, real-time adaptation but rely heavily on pretraining and offline demonstrations.

To overcome these limitations, we present *Sampling-Based Adaptive Motion Planning* (SBAMP), a hybrid framework that

combines RRT* global planning with an online, Lyapunov-stable SEDS-inspired controller. By fitting a Lyapunov-stable vector field online to the current RRT* waypoint sequence (eliminating the need for pre-collected data) and interleaving millisecond-scale local updates with lower-frequency global replanning under formal convergence bounds, SBAMP enables the robot to continuously flow through the learned field until new information demands only a lightweight RRT* update—thereby avoiding expensive full replanning.

Our main contributions are:

- A *robust, bi-level SBAMP architecture* combining RRT* global planning with an online, Lyapunov-stable SEDS-inspired controller that significantly augments and rescues RRT* under the most severe perturbations and failure scenarios.
- An *efficient interleaving scheme* that minimizes global replanning while preserving provable stability.
- A *highly modular Python implementation of the SBAMP algorithm* that scales easily and integrates seamlessly with existing robotic systems.
- *Extensive evaluation on RoboRacer [4] hardware and simulation*, showcasing rapid disturbance recovery and robust obstacle resilience.

The rest of the paper is organized as follows. Section II reviews related work. Section III details the SBAMP theoretical framework and describes its real-time deployment on the RoboRacer autonomous racing platform. Section IV describes our experiments and metrics. Section V presents results, and Section VI concludes with future directions.

II. RELATED WORK

A. Sampling-Based Motion Planning Algorithms

Sampling-Based Motion Planning (SBMP) has become a cornerstone of robotic motion planning, enabling efficient navigation in high-dimensional and complex environments where explicit representation of obstacles is computationally infeasible. Algorithms such as Rapidly-exploring Random Trees (RRT) and its asymptotically optimal variant RRT* have been widely adopted due to their ability to quickly generate feasible, collision-free paths without requiring a full model of the environment [5]–[7]. RRT* iteratively refines its solution, converging to an optimal path as more samples are drawn, and

has been extended for anytime planning to improve path quality during execution [6]. Variants like bi-directional RRT* and heuristic-enhanced methods further accelerate convergence in high-dimensional spaces [8].

Despite their strengths, classical SBMP methods are limited by their static nature; once a path is generated, they lack mechanisms for real-time adaptation to sudden environmental changes or perturbations. Extensions to handle kinodynamic constraints and dynamic environments often increase computational complexity, limiting their practicality for online applications. While heuristic-based improvements mitigate some limitations, these approaches still struggle to balance global optimality with rapid, real-time adaptation in unstructured environments.

B. Learning-Based Dynamical Systems

Learning-based dynamical systems (DS) address real-time adaptability by modeling robot motions as stable attractor systems. A prominent example is the Stable Estimator of Dynamical Systems (SEDS), which fits a Gaussian mixture model to demonstration data under Lyapunov-based constraints, guaranteeing global asymptotic stability while generalizing across demonstrations [2]. Linear Parameter-Varying DS (LPV-DS) generalize this by embedding state-dependence into parameter-varying linear models, enabling scalable control design with stability certificates across operating regimes [3]. Another method, Gaussian Process Modulated Dynamical Systems (GP-MDS), where a state-dependent modulation matrix is learned online via Gaussian Process Regression, preserving the base system’s equilibrium and stability properties while automatically attenuating modulation far from demonstration data [9].

While offering formal stability guarantees, these DS paradigms exhibit complementary shortcomings. SEDS and LPV-DS achieve global convergence but depend on comprehensive offline demonstration datasets to capture the full range of operating dynamics and ensure stability across regimes [2], [3]. GP-MDS, in turn, enables online refinement without batch training yet relies on careful Gaussian process kernel tuning and sparse-data management, adding overhead and risking lag under rapid environmental changes [9]. These dependencies on extensive demonstrations, offline fitting, and nontrivial model tuning obstruct tight integration with global planners and limit responsiveness in real-time, unstructured scenarios.

C. Hybrid Approaches

Hybrid frameworks seek to combine sampling based motion planning’s global exploration with DS’s local adaptability. For instance, recent work integrates RRT* with Lyapunov-certified, demonstration-driven SEDS-inspired controllers to locally “funnel” around nominal waypoints [7], but this approach relies on pre-collected demonstrations and lacks a unified global Lyapunov stability guarantee. Robust sampling-based planners that incorporate forward reachability analysis

to handle uncertainty [10] likewise do not provide unified Lyapunov-style stability proofs. Chance-constrained RRT variants employing tube-based LPV-MPC [11] and LPV embedding with sequential quadratic programming nonlinear MPC [12] achieve probabilistic robustness and efficient trajectory optimization but incur significant online computational overhead due to per-step optimization solves. While these hybrid methods advance adaptability or safety, none simultaneously avoid dependence on demonstrations, guarantee unified stability, and maintain real-time computational tractability.

Unlike prior hybrid planners, SBAMP fits its local SEDS-style Gaussian mixture model on-the-fly from each newly planned RRT* segment, eliminating any need for offline demonstration data, and synthesizes every local controller under a common Lyapunov-function constraint, guaranteeing global asymptotic stability throughout replanning. Moreover, by partitioning work into optimized ROS2 nodes that run RRT* in C++ and then simply evaluate the learned vector field (a weighted sum of linear maps) at control rate rather than solving an optimization at each step, SBAMP sustains real-time performance in dynamic, unstructured environments.

III. SAMPLING BASED ADAPTIVE MOTION PLANNING

A. SBAMP Theoretical Framework

Figure 1 depicts the overall SBAMP control loop. At its core, SBAMP runs two modules in parallel—a global RRT* planner and a local SEDS controller—with a lightweight decision logic that activates the SEDS generator whenever the planner produces a new path.

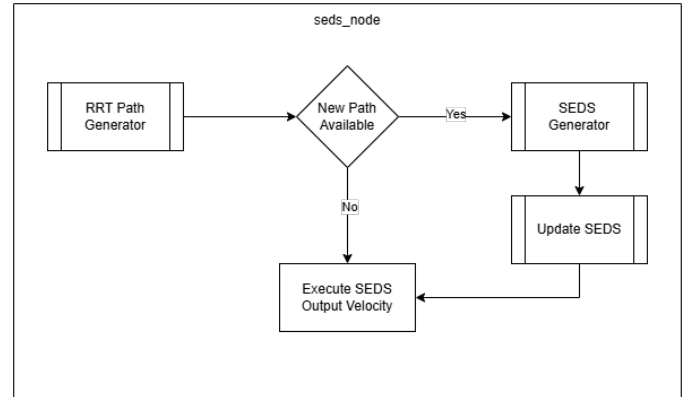


Fig. 1: Flowchart of the SBAMP theoretical framework. When *New Path Available?* is true, the SEDS generator refits the dynamical system to the latest RRT* segment; otherwise, the existing SEDS velocity command is executed.

At a high level, SBAMP is designed to:

- *Integrate SEDS with RRT**: leverage RRT* for global, asymptotically-optimal waypoint generation while employing a Lyapunov-stable mixture dynamics controller (SEDS) for local, online trajectory adaptation without pre-training;

- *Achieve seamless waypoint transitions:* dynamically shift the SEDS equilibrium to the current nearest waypoint, preserving continuity of the commanded velocity.

To realize these objectives, we structure SBAMP as a bi-level framework comprising three interacting components:

1) *Global Path Planning via RRT**: We incrementally grow and rewire a tree $\mathcal{T} \subset \mathcal{C}_{\text{free}}$ using RRT* [6], yielding a waypoint sequence

$$\tau = \{x_0, x_1, \dots, x_g\} \subset \mathbb{R}^n.$$

Every planner cycle (period Δt_G) samples $x_{\text{rand}} \sim \mathcal{U}(\mathcal{C}_{\text{free}})$, extends toward it, and performs local rewiring in $O(\log N)$ to improve path optimality.

2) *Local Trajectory Adaptation via SEDS*: At control rate ($\Delta t_C \ll \Delta t_G$), the robot state $\xi(t)$ is driven by a convex mixture of K linear subsystems [2]:

$$\dot{\xi} = f(\xi) = \sum_{k=1}^K \gamma_k(\xi) (A_k \xi + b_k), \quad \sum_{k=1}^K \gamma_k(\xi) = 1, \quad \gamma_k(\xi) \geq 0, \quad (1)$$

where each $A_k + A_k^\top \prec 0$ (ensuring $V(\xi) = \xi^\top \xi$ decays) and

$$b_k = -A_k x_i \implies f(x_i) = 0$$

at the active waypoint x_i . Whenever the planner updates τ , the decision block in Fig. 1 routes the new segment to the “SEDS generator”, which recomputes $\{b_k\}$ to recenter the attractor at the next waypoint x_{i+1} .

3) *Real-Time Integration and Stability*: The decision diamond “New Path Available?” in Fig. 1 enforces the following logic:

- *Yes*: invoke the SEDS generator to refit $\{b_k\}$ for the new segment $\{x_i, x_{i+1}\}$, then proceed to execute the updated velocity ξ ;
- *No*: continue executing the velocity output of the current SEDS model.

By design, the attractor shift is performed without resetting velocity, i.e.

$$\dot{\xi}^+ = \dot{\xi}^-,$$

which guarantees no jump in the control command. Let $\tau_D > 0$ denote the *average dwell-time*, i.e. the minimum average interval between consecutive switches (formally, the number of switches $N_\sigma(t_1, t_2)$ in any interval $[t_1, t_2]$ satisfies $N_\sigma \leq N_0 + (t_2 - t_1)/\tau_D$). Under the average-dwell-time theorem [13], if the SEDS update period Δt_C and RRT* planning period Δt_G satisfy

$$\Delta t_C \ll \tau_D \leq \Delta t_G,$$

then the switched system remains globally asymptotically stable to the final goal x_g .

Together, these three modules realize a provably stable, real-time adaptive planner that seamlessly fuses sampling-based global planning with Lyapunov-stable local control.

B. SBAMP Implementation on RoboRacer

Building on the stability guarantees of Section III-A, we realized SBAMP on the RoboRacer [4] platform using ROS2 Humble. Figure 2 depicts the node graph and dataflow: laser scans and odometry feed into a local occupancy grid, the planner produces or updates a waypoint sequence τ , the SEDS controller issues velocity commands, and—if enabled—a visualization node renders the entire state in RViz2. All SBAMP ROS2 packages and related scripts are publicly available at <https://github.com/Shreyas0812/SBAMP>.

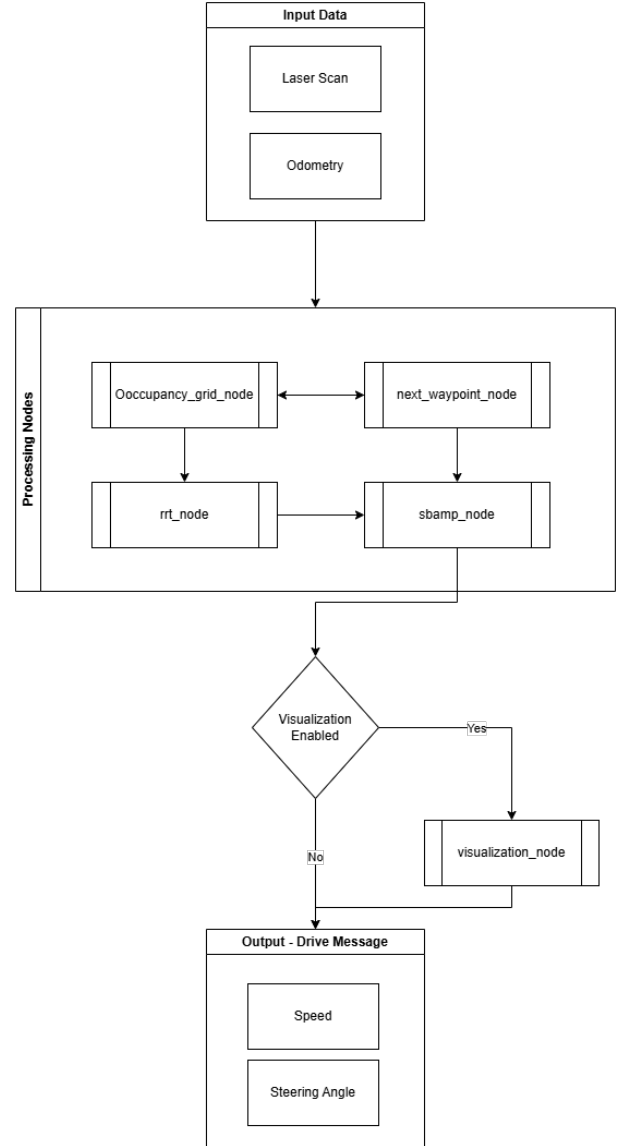


Fig. 2: ROS2 architecture for SBAMP on RoboRacer. Laser scan and odometry feed the occupancy_grid_node; next_waypoint_node and rrt_node produce τ ; sbamp_node generates control; visualization_node is launched if enabled.

1) *Software Architecture*: Five ROS2 nodes form the backbone of SBAMP. The *Occupancy Grid Node* fuses LIDAR scans (from `/scan`) with vehicle odometry

(/ego_racecar/odom) to maintain a local occupancy grid in real time. The *Next Waypoint Node* consumes this grid and extracts the next feasible goal, publishing it on /next_waypoint. In parallel, the *RRT* Node* continuously replans a collision-free path through the environment, outputting the current path segment on /rrt_path. When enabled, the *Visualization Node* subscribes to all intermediate topics and renders waypoints, planned trajectories, and SBAMP’s vector fields in RViz2 for live debugging. Finally, the *SBAMP Node* fits a SEDS-style model to each RRT* segment and transforms the resulting vector field into Ackermann drive commands, which it publishes to the vehicle controller at high frequency. With all nodes in place, the simulation environment is shown in Figure 3.

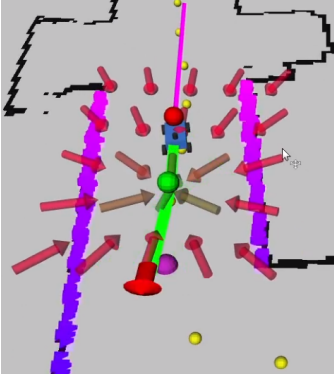


Fig. 3: RoboRacer simulation environment with SBAMP-generated vector field and RRT* path.

2) *Platform Configuration*: All experiments were performed on the RoboRacer platform, whose kinematics obey

$$\dot{x} = v \cos \theta, \quad \dot{y} = v \sin \theta, \quad \dot{\theta} = \frac{v}{L} \tan \delta.$$

Perception is provided by an 812-beam SICK TIM781 LIDAR, and actuation uses ROS2 to send Ackermann steering commands directly to the vehicle hardware. Figure 4 illustrates the F1/10 racing-car used for our real-world validation, with the roof-mounted LIDAR sensor.



Fig. 4: F1/10 hardware platform used for real-world validation.

3) *System Integration*: Table I lists the four core ROS2 packages that constitute our implementation. Each package encapsulates one aspect of the SBAMP pipeline, from global planning to real-time control.

TABLE I: Core ROS2 Package Specifications

Package	Functionality
rrt_star	OMPL-based path generation
sed_control	GMM optimization with CVXPY
waypoint_manager	Cubic spline interpolation
fltenths_utils	LIDAR data preprocessing

4) *Dynamical System Training*: We collected over 500 RRT*-driven trajectories, logging state-velocity pairs $\{\xi^{(k)}, \dot{\xi}^{(k)}\}$ at 50 Hz. These samples were used to solve

$$\min_{A_i, b_i} \sum_{k=1}^N \|\dot{\xi}^{(k)} - f(\xi^{(k)})\|^2$$

subject to Lyapunov stability constraints

$$\sum_{i=1}^K \gamma_i(\xi) = 1, \quad \gamma_i(\xi) \geq 0,$$

using the CVXPY [14] optimization library.

IV. EXPERIMENTS

We compare SBAMP to standard RRT* through three complementary studies: (1) quantitative simulation of perturbation sensitivity, (2) qualitative simulation of extreme-failure recovery, and (3) real-world hardware validation. These experiments together characterize SBAMP’s replanning performance, rescue capability, and robustness under physical disturbances.

A. Experiment 1: Robustness to Lateral Perturbations

1) *Objective*: Measure how increasing lateral displacements degrade the replanning frequency f_{plan} of RRT* versus SBAMP.

2) *Simulator & Vehicle Model*: F1TENTH-gym ROS2 environment with Ackermann kinematics at $v = 1$ m/s.

3) *Protocol*: At a fixed straight-away point, “teleport” the vehicle laterally by various distances immediately before each planning cycle. For each Δd , perform $N = 20$ independent runs.

4) Metrics:

- Replanning frequency f_{plan} (Hz)
- Forward velocity v (m/s)

5) *Analysis*: Plot f_{plan} versus Δd for both methods. A slower decay in f_{plan} indicates greater robustness.

B. Experiment 2: Recovery from Extreme Failures

1) *Objective*: Qualitatively illustrate scenarios in which SBAMP's dynamical-system controller recovers when RRT* loses its waypoint buffer.

2) *Environment*: A 5 m×2 m corridor with obstacles.

3) *Disturbance Modes*:

- *Translational jump*: teleport by increasing Δd .
- *Rotational offset*: yaw the vehicle by up to 90° (RRT* often fails above 60°).
- *Corner trap*: place the car into a corner, facing the wall.

4) *Procedure*: For each mode, increase disturbance magnitude until each planner fails (cannot replan or collides). Record the failure threshold for RRT* and SBAMP.

5) *Outcomes*:

- Failure magnitude (distance, angle, or trap severity)
- Video snapshots of recovery vs. stall/collision

C. Experiment 3: Real-World Stress Test on F1/10 Hardware

1) *Objective*: Demonstrate SBAMP's stability and obstacle avoidance under physical disturbances on the RoboRacer F1/10 platform.

2) *Hardware Setup*:

- F1/10 race car with 812-beam LiDAR and ROS2 Ackermann control (as discussed in Section III-B2)
- Course: loop around Levine Hall with straightaways, tight turns, and cluttered corridors

3) *Disturbance Protocol*: During SBAMP runs, apply:

- *Pushes*: manual lateral shove up to 1m
- *Spins*: rotate heading up to 90°
- *Obstacles*: place unexpected objects on the path

4) *Measurements*: For each trial, record whether SBAMP:

- Holds pose when the waypoint buffer drains
- Resumes smoothly during the DS recovery and when RRT* replans
- Avoids collision with the unexpected obstacles

V. EVALUATION

All simulation experiments and real-hardware tests are available in video form at <https://youtu.be/mtq3qeJFjX0>.

A. Real-Time Replanning Latency Analysis

Figure 5 depicts the replanning frequency f_{plan} as a function of lateral perturbation magnitude Δd for both standard RRT* and SBAMP. As Δd increases from 2.25m to 2.75m, RRT*'s replanning frequency falls below the minimum 2Hz

threshold required for stable 1 m/s operation, whereas SBAMP consistently maintains approximately 60Hz, thereby ensuring real-time responsiveness under significant disturbances.

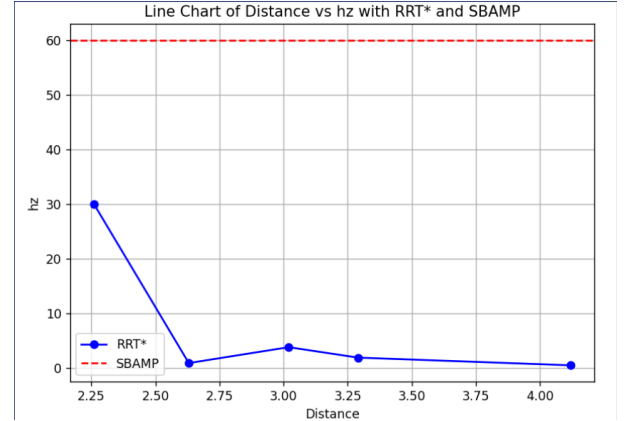


Fig. 5: Replanning Frequency vs. Lateral Perturbation for RRT* and SBAMP

The replanning frequency f_{plan} quantifies the maximum allowable latency between successive global-planning updates. Suppose at time t RRT* produces a path segment $\{x_0, x_1, x_2, x_3\}$. If the interval $1/f_{\text{plan}}$ exceeds the time required for the vehicle to traverse from x_0 to x_3 , then, absent a new trajectory, the controller lacks a valid reference and stability cannot be guaranteed.

SBAMP addresses this by coupling each RRT* waypoint with a locally stable dynamical-system (DS) attractor. Even when the global planner's update is delayed, the DS controller continuously generates a velocity command driving the vehicle toward the last computed waypoint. As soon as RRT* delivers a new path, SBAMP instantiates a new DS attractor that smoothly interpolates from the vehicle's current state to the next waypoint. Consequently:

- The vehicle always possesses a well-defined reference trajectory, preventing unbounded drift when planning updates are slow.
- Transitions between successive RRT* solutions occur without discontinuities in the control signal.

For a nominal speed of 1 m/s, ensuring that the vehicle moves no more than 0.5m between updates requires $f_{\text{plan}} \geq 2$ Hz. SBAMP's effective replanning frequency of 60Hz thus provides a substantial safety margin, whereas RRT* alone can fall below this threshold under perturbations, compromising real-time adaptability and stability.

B. Experiment 2: Resilience in Planner Failure Scenarios

Figures 6–8 compare standard RRT* (left) against SBAMP (right) under three challenging scenarios: large translation perturbations, large rotational perturbations, and tight-corner recovery.

Under a substantial lateral displacement (Figure 6), RRT* often fails to reconnect to the original trajectory within a single

planning cycle, leaving the vehicle without a feasible path. In contrast, SBAMP’s dynamical-system controller immediately generates a locally stable attractor toward the last RRT* waypoint, ensuring prompt re-entry into the corridor and seamless hand-off once a new global path is available. We have also analyzed this behavior in the quantitative analysis under Section V-A.

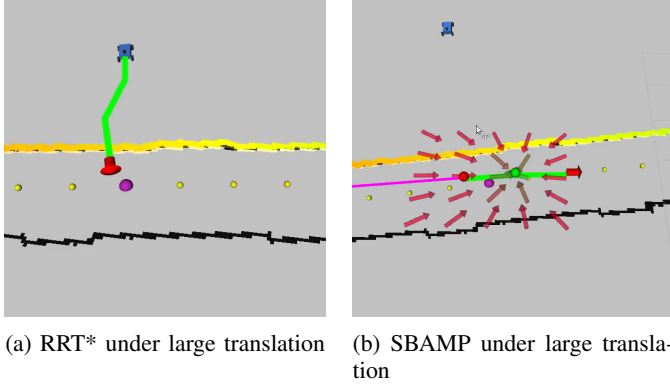


Fig. 6: Comparison of planner recovery under large translational perturbations.

When subjected to a rotation exceeding 60° (Figure 7), RRT* alone frequently times out or returns a path that directs the vehicle into obstacles, due to excessive planning latency. SBAMP, however, continues to drive the vehicle toward the nominal waypoint, maintaining stability until the planner produces a safe trajectory.

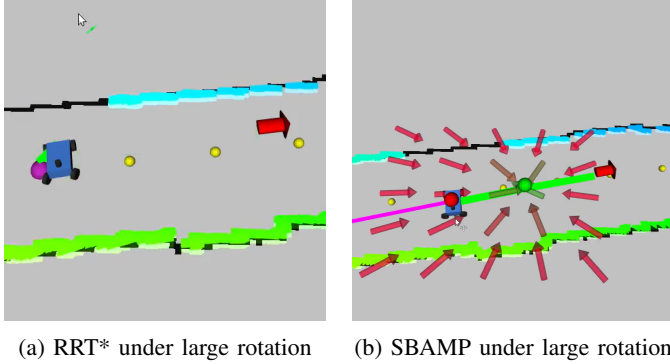


Fig. 7: Comparison of planner recovery under large rotational perturbations.

Finally, in a tight-corner environment (Figure 8), RRT*’s sparse sampling can yield waypoints that steer the vehicle dangerously close to opposing walls. SBAMP mitigates this by following only the immediately reachable waypoint via its SEDS-based vector field and pausing further progression until the next plan arrives, resulting in smoother, collision-free recovery.

These qualitative observations demonstrate that SBAMP markedly enhances resilience in scenarios where RRT* alone cannot ensure timely, safe replanning or recovery from severe disturbances.

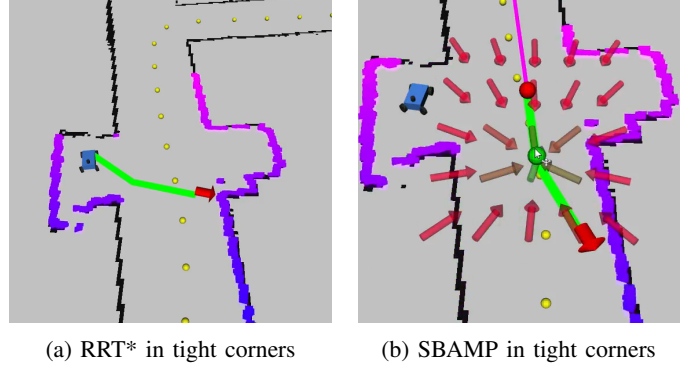


Fig. 8: Comparison of planner performance in tight-corner scenarios.

To emphasize SBAMP’s exceptional resilience, Figures 9a–9b demonstrate that SBAMP reliably returns the vehicle to the planned corridor even after extremely large translational and rotational displacements. In each case the SEDS-based attractor guides the robot back to the vicinity of the last RRT* waypoint, at which point a new global trajectory is computed and followed without discontinuity. These results confirm that SBAMP’s integration of dynamical-systems control with sampling-based planning provides robust recovery from severe disturbances anywhere within the connected free-space.

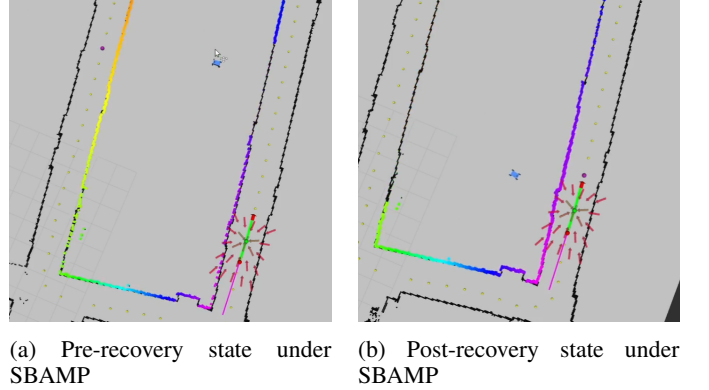


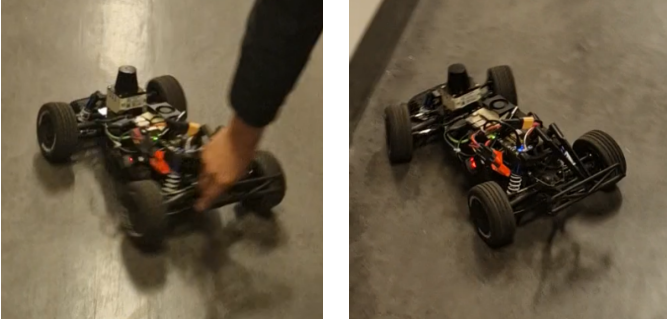
Fig. 9: SBAMP recovery behavior before and after large translational and rotational perturbations.

C. Experiment 3: Human-Induced Perturbations on Hardware

To validate SBAMP’s robustness on real hardware, we manually applied large translational and rotational disturbances to the F1/10 vehicle during closed-loop operation. Figure 10 shows the pre- and post-disturbance states for a human-induced rotation, and Figure 11 shows the corresponding translational perturbation results.

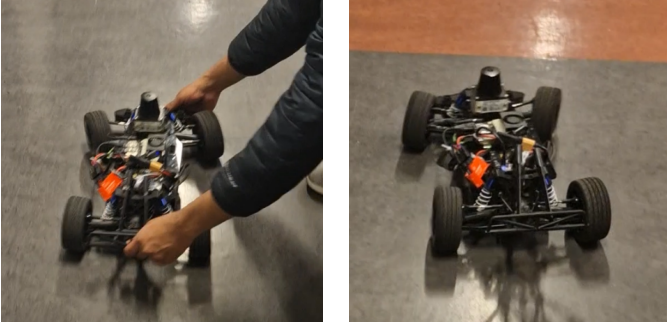
In both scenarios, immediately after the disturbance the vehicle deviates from its nominal path (left panels), after which the SEDS attractor generates control commands that drive the vehicle back toward the last RRT* waypoint (right panels). Once re-entry is achieved, SBAMP seamlessly switches

to the newly computed global plan. Over 20 randomized human-applied perturbations, SBAMP attained a near-100% recovery rate; failures rarely occurred only when rotational displacements were so large that the subsequent waypoint was misidentified, although the vehicle still returned to the most recent waypoint. These hardware trials demonstrate that SBAMP’s combined sampling-based planning and DS control maintains stable, safe operation under realistic disturbances.



(a) Pre-perturbation state under human-induced rotation (b) Post-perturbation recovery under human-induced rotation

Fig. 10: SBAMP performance before and after a human-applied rotational disturbance.



(a) Pre-perturbation state under human-induced translation (b) Post-perturbation recovery under human-induced translation

Fig. 11: SBAMP performance before and after a human-applied translational disturbance.

Figures 12–13 illustrate SBAMP’s real-world obstacle-avoidance behavior under two drift scenarios. In scenario A (Figure 12), the vehicle is perturbed toward a large box; SBAMP immediately generates a left-side avoidance trajectory around the obstacle and then rejoins the nominal corridor. In the opposite scenario B (Figure 13), SBAMP executes a right-side avoidance maneuver before recovering the original plan. SBAMP’s obstacle avoidance mechanism is non-invasively enhanced by RRT* for global path generation: when the sampling-based planner produces timely, collision-free waypoints, SBAMP does not interfere negatively. When planning or perception challenges or external disturbances cause RRT* to falter, the DS attractor seamlessly intervenes to recover a safe trajectory without modifying the underlying planner.



(a) SBAMP approaching obstacle region (scenario A). (b) SBAMP path adaptation around obstacle (scenario A).

Fig. 12: SBAMP obstacle-avoidance performance in scenario A - turning left around obstacle.



(a) SBAMP approaching obstacle region (scenario B). (b) SBAMP path adaptation around obstacle (scenario B).

Fig. 13: SBAMP obstacle-avoidance performance in scenario B-turning right around obstacle.

Across three complementary studies, SBAMP consistently outperforms standard RRT* in both responsiveness and robustness to external disturbances and perturbations. In simulation, SBAMP maintained control-loop rates above 60Hz under perturbations that reduced RRT* replanning below the 2Hz safety threshold. In extreme-failure scenarios, the SEDS-based local attractor guaranteed safe re-entry where RRT* alone stalled or collided. Finally, on physical F1/10 hardware, SBAMP achieved near-100% recovery from large translational and rotational disturbances and executed reliable left- and right-side obstacle avoidance without modifying the underlying planner. These results collectively demonstrate that integrating sampling-based global planning with online, Lyapunov-stable dynamical-systems control yields a motion-planning framework capable of real-time adaptation, formal stability guarantees, and seamless recovery in complex, dynamic environments.

VI. CONCLUSION

We have introduced SBAMP, a bi-level motion-planning framework that non-invasively augments RRT* with a Lyapunov-stable dynamical-systems controller to achieve true

on-the-fly adaptation with no prior training data. SBAMP continuously converts each RRT* waypoint into a locally stable attractor, ensuring a valid control reference even when global replanning lags. Our threefold evaluation, including control-loop latency analysis in simulation, qualitative recovery from extreme failures, and real-world hardware stress tests, demonstrates that SBAMP sustains high replanning frequencies, reliably recovers from large translational and rotational disturbances, and executes safe obstacle avoidance, all without any offline learning or demonstration dataset.

Future work includes integrating SBAMP with receding-horizon optimizers such as model predictive control (MPC) or MPPI to combine finite-horizon cost minimization with provably stable local attractors. Additional efforts will embed obstacle-repulsive modulation directly into the dynamical-systems layer, further reducing reliance on occupancy-grid update rates and enlarging safety margins. Finally, a systematic perturbation study will be conducted to map SBAMP's region of attraction, and the framework will be extended to high-dimensional manipulators to broaden its applicability across autonomous robotics tasks.

REFERENCES

- [1] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, June 2011. [Online]. Available: <https://arxiv.org/pdf/1105.1186>
- [2] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, Oct 2011. [Online]. Available: <https://ieeexplore.ieee.org/document/5953529>
- [3] N. Figueroa and A. Billard, "A physically-consistent bayesian non-parametric mixture model for dynamical system learning," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 87. PMLR, Oct 2018, pp. 927–946. [Online]. Available: <https://proceedings.mlr.press/v87/figueroa18a/figueroa18a.pdf>
- [4] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "Fltenth: An open-source evaluation environment for continuous control and reinforcement learning," in *NeurIPS 2019 Competition and Demonstration Track*. PMLR, 2020, pp. 77–89.
- [5] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006. [Online]. Available: <http://planning.cs.uiuc.edu/>
- [6] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt*," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1478–1483.
- [7] A. Orthey *et al.*, "Sampling-based motion planning: A comparative review," *Annual Review of Control, Robotics, and Autonomous Systems*, 2024.
- [8] B. Akgun and M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 2640–2645.
- [9] K. Kronander, S. M. Khansari-Zadeh, and A. Billard, "Incremental motion learning with locally modulated dynamical systems," *Robotics and Autonomous Systems*, vol. 70, pp. 52–62, 2015.
- [10] A. Wu, T. Lew, K. Solovey, E. Schmerling, and M. Pavone, "Robust-RRT: Probabilistically-complete motion planning for uncertain nonlinear systems," *arXiv preprint arXiv:2205.07728*, 2022.
- [11] M. Nezami, H. S. Abbas, N. T. Nguyen, and G. Schildbach, "Robust tube-based lqv-mpc for autonomous lane keeping," *arXiv preprint arXiv:2210.02971*, 2022.
- [12] D. S. Karachalios and H. S. Abbas, "Efficient nonlinear mpc by leveraging lqv embedding and sequential quadratic programming," *arXiv preprint arXiv:2403.19195*, 2025.
- [13] J. P. Hespanha and A. S. Morse, "Stability of switched systems with average dwell-time," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 2. Phoenix, AZ: IEEE, 1999, pp. 2655–2660. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=831330>
- [14] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.