SBAMP: Sampling Based Adaptive Motion Planning

Shreyas Raorane*, Kabir Ram Puri*, Anh-Quan Pham†

***Abstract*—Autonomous robots operating in dynamic environments must balance global path optimality with real-time responsiveness to disturbances. This requires addressing a fundamental trade-off between computationally expensive global planning and fast local adaptation. Sampling-based planners such as RRT* produce near-optimal paths but struggle under perturbations, while dynamical systems approaches like SEDS enable smooth reactive behavior but rely on offline data-driven optimization. We introduce Sampling-Based Adaptive Motion Planning (SBAMP), a hybrid framework that combines RRT*-based global planning with an online, Lyapunov-stable SEDS-inspired controller that requires no pre-trained data. By integrating lightweight constrained optimization into the control loop, SBAMP enables stable, real-time adaptation while preserving global path structure. Experiments in simulation and on RoboRacer hardware demonstrate robust recovery from disturbances, reliable obstacle handling, and consistent performance under dynamic conditions.**

***Index Terms*—Motion Planning, Dynamical Systems, Lyapunov Stability, Real-Time Adaptation**

I. Introduction

Autonomous robots must navigate geometrically complex and dynamically changing environments, including dodging pedestrians, rerouting around debris, and recovering from sudden collisions, demanding two competing capabilities: *global path quality* (near-optimal, collision-free trajectories over long horizons) and *local reactivity* (instantaneous adaptation to new obstacles or perturbations).

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

We present *Sampling-Based Adaptive Motion Planning* (SBAMP), a hybrid framework that fits a Lyapunov-stable vector field online to each RRT* waypoint segment, requiring no pre-collected data, and interleaves high-rate local control with lower-frequency global replanning to avoid expensive full replanning. Our main contributions are:

- *A bi-level SBAMP architecture* combining RRT* global planning with an online, Lyapunov-stable SEDS-inspired controller that rescues RRT* under severe perturbations.
- *An efficient interleaving scheme* minimizing global replanning while preserving provable local stability.
- *Extensive evaluation on RoboRacer [4]* hardware and simulation, showcasing rapid disturbance recovery and robust obstacle resilience.

The authors are with the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory, University of Pennsylvania, Philadelphia, PA, 19104, USA.

* Equal contribution.

† Corresponding author: quanpham@seas.upenn.edu.

Code & videos available at: https://github.com/anhquanpham/SBAMP.

II. Related Work

Sampling-Based Motion Planning (SBMP) algorithms such as RRT* enable efficient, collision-free path planning in high-dimensional spaces and converge asymptotically to optimal solutions [5]–[7]. Variants including bi-directional RRT* and heuristic-enhanced methods further accelerate convergence [8]. However, classical SBMP methods are inherently static: once a path is generated, they lack mechanisms for real-time adaptation to environmental changes, and extensions handling kinodynamic or dynamic constraints often sacrifice online practicality.

Learning-based Dynamical Systems (DS) address adaptability by modeling robot motion as stable attractor systems. SEDS fits a Gaussian mixture model to demonstration data under Lyapunov constraints, ensuring global asymptotic stability [2], while LPV-DS generalizes this via state-dependent linear models with stability certificates across operating regimes [3]. GP-MDS enables online refinement through Gaussian Process Regression without batch training, though it requires careful kernel tuning and sparse-data management [9]. All three paradigms share a core limitation: dependence on offline demonstrations, extensive dataset collection, or nontrivial model tuning, which hinders integration with global planners in real-time, unstructured environments.

Hybrid frameworks attempt to combine global exploration with local adaptability. Recent work couples RRT* with Lyapunov-certified, demonstration-driven controllers to funnel around nominal waypoints [7], but relies on pre-collected data and lacks a unified global stability guarantee. Robust samplers incorporating forward reachability analysis [10] similarly omit Lyapunov-style proofs, while chance-constrained RRT variants using tube-based LPV-MPC [11] and LPV-embedded nonlinear MPC [12] achieve probabilistic robustness but incur significant per-step optimization overhead. None of these methods simultaneously eliminate demonstration dependence, guarantee unified stability, and maintain real-time tractability.

SBAMP addresses all three gaps by fitting its SEDS-style Gaussian mixture model on-the-fly from each newly planned RRT* segment—requiring no offline data—and synthesizing every local controller under a common Lyapunov-style local stability constraint. By decoupling RRT* planning from vector-field evaluation (a weighted sum of linear maps at

control rate), SBAMP sustains real-time performance without per-step optimization solves.

III. Sampling Based Adaptive Motion Planning

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 refits the dynamical system whenever the planner produces a new path.

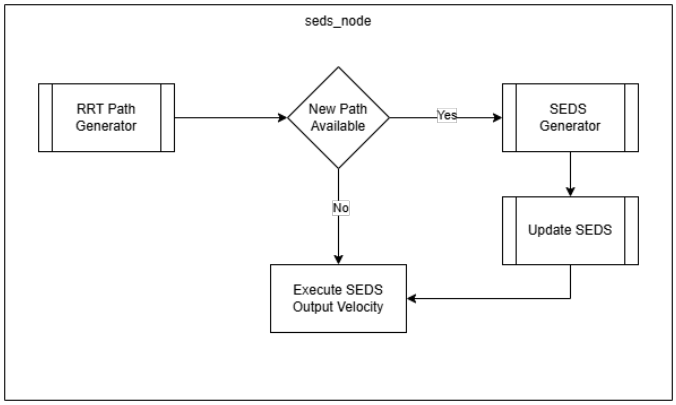


Fig. 1: Flowchart of the SBAMP theoretical framework. When *New Path Available?* is true, the SEDS generator refits the dynamical system to the latest RRT* segment; otherwise, the existing SEDS velocity command is executed.

SBAMP is structured as a bi-level optimization framework comprising three interacting components.

A. Global Path Planning via RRT*

We incrementally grow and rewire a tree $\mathcal{T} \subset \mathcal{C}_{\text{free}}$ using a lightweight RRT*-inspired planner [6], yielding a waypoint sequence

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

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

B. 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]:

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

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

$$b_k = -A_k\, x_i \quad \Longrightarrow \quad f(x_i) = 0$$

at the active waypoint x_i. We pose local controller synthesis as a real-time constrained optimization problem: given waypoints $\{x_i, x_{i+1}\}$, find a K-component mixture of linear systems that (1) produces smooth vector fields via GMM fitting, and (2) guarantees local asymptotic stability via Hurwitz projection. Formally, for each component k:

$$\min_{A_k} \|A_k - \hat{A}_k\|_F \quad \text{s.t.} \quad A_k + A_k^\top \prec 0,$$

This per-cycle fit requires no stored dataset and completes within the replanning loop. The attractor is then recentered at x_{i+1} by updating $\{b_k\}$ accordingly.

C. Real-Time Integration and Stability

Upon receiving a new path, the SEDS generator refits $\{b_k\}$ and updates $\dot{\xi}$; otherwise, the current model is used, with attractor shifts preserving velocity continuity. 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 each fitted subsystem remains locally stable to its active waypoint, with empirically observed recovery to the final goal x_g. Together, these three modules realize a real-time adaptive planner with Lyapunov-stable local subsystems and no offline training data required. Full implementation details on the RoboRacer [4] hardware are provided in Appendix A.

IV. Experiments

We evaluate SBAMP against standard RRT* across three complementary studies designed to stress-test the two core claims of the framework: that the DS attractor preserves control authority during replanning gaps, and that Lyapunov-stable local control enables recovery from disturbances that exceed RRT*'s planning capacity.

A. Computational Efficiency Under Disturbance

A fundamental vulnerability of purely sampling-based planners is that replanning latency grows with perturbation magnitude: as the vehicle is displaced farther from its prior tree, RRT* must explore a larger region of $\mathcal{C}_{\text{free}}$ before recovering a feasible path, and if that latency exceeds the time to exhaust the current waypoint buffer, the controller loses its reference entirely. We quantify this degradation by teleporting the vehicle laterally by $\Delta d \in [2.25, 2.75]$ m at a fixed straightaway immediately before each planning cycle, repeating $N = 20$ trials per displacement in the ROS2 simulator at $v = 1$ m/s.

Figure 2 reports replanning frequency f_{plan} as a function of Δd. RRT* degrades sharply across this range, falling below the 2 Hz minimum required to ensure the vehicle advances no more than 0.5 m between updates at nominal speed. SBAMP, by contrast, sustains approximately 60 Hz throughout, because the DS attractor provides a well-defined velocity command toward the last known waypoint irrespective of planner latency, and transitions to each new RRT* solution without discontinuity in the control signal. This 30× margin over the stability threshold directly validates the dwell-time argument of Section III.

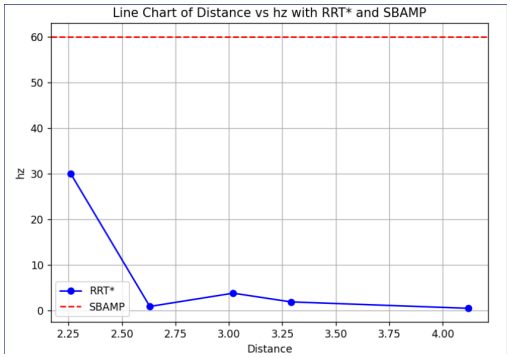
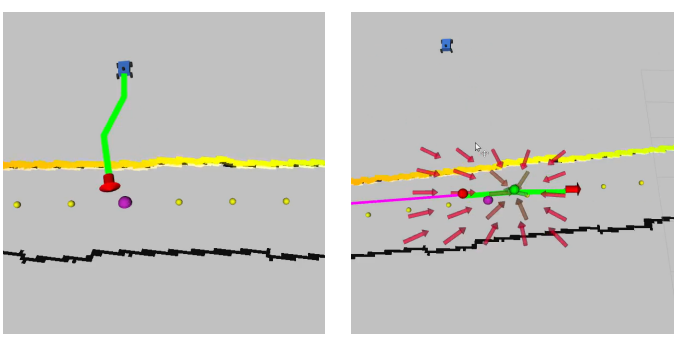


Fig. 2: Replanning frequency vs. lateral perturbation. RRT* falls below the 2 Hz stability threshold as Δd grows; SBAMP maintains approximately 60 Hz throughout.

B. Recovery from Extreme Planner Failures

To characterize the boundary of RRT*'s recovery envelope and demonstrate SBAMP's behavior beyond it, we subjected both planners to three qualitatively distinct failure modes in a $5\,\mathrm{m} \times 2\,\mathrm{m}$ corridor: large translational jumps, rotational offsets up to 90°, and corner entrapment. In each case we increased disturbance magnitude until the planner either collided or failed to produce a feasible path within the planning budget.

Under large lateral displacement (Figure 3), RRT* exhausts its planning budget before reconnecting to the corridor, leaving the vehicle with no valid reference. SBAMP's attractor immediately redirects the vehicle toward the last RRT* waypoint, maintaining bounded tracking error until the planner recovers and a new path is handed off without discontinuity.

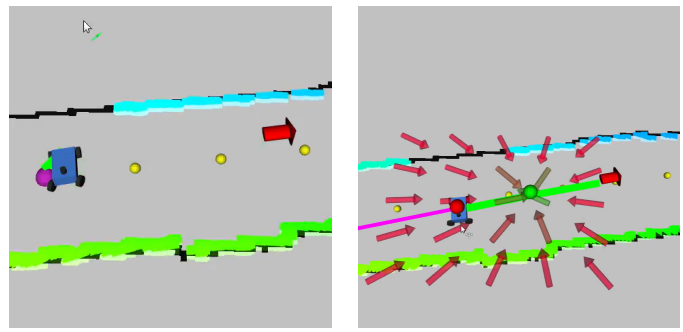


(a) RRT* under large translation (b) SBAMP under large translation

Fig. 3: Planner recovery under large translational perturbation. RRT* loses its reference; SBAMP maintains a stable attractor throughout.

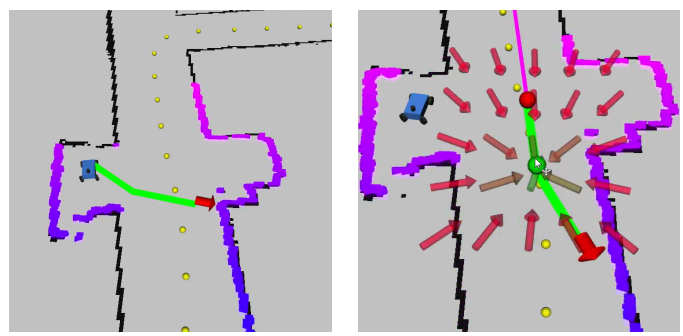
Rotational offsets exceeding 60° expose a second failure mode: RRT* either times out or produces paths that, when executed, direct the vehicle into obstacles before a corrective replan can arrive (Figure 4). SBAMP is unaffected because the DS controller operates on the error to the current waypoint in Cartesian space, not on heading, and continues issuing stable commands regardless of orientation.

In tight-corner scenarios (Figure 5), RRT*'s sparse sampling produces waypoints that, under execution, bring the vehicle within collision range of opposing walls. SBAMP resolves this by committing only to the immediately reachable waypoint via the SEDS vector field and withholding progression until the next global plan is available, yielding smooth, collision-free negotiation of the corner.



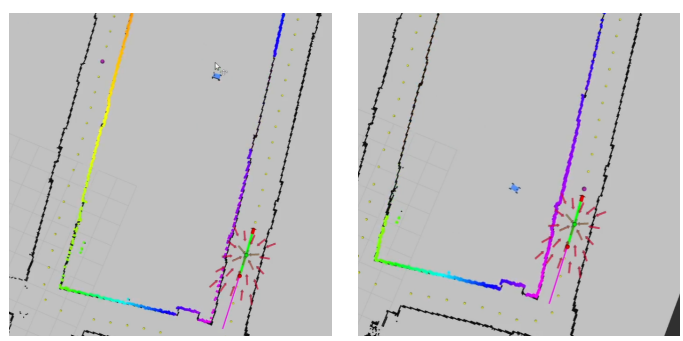
(a) RRT* under large rotation (b) SBAMP under large rotation

Fig. 4: Planner recovery under large rotational perturbation. RRT* produces unsafe paths; SBAMP maintains stable convergence to the last waypoint.



(a) RRT* in tight corners (b) SBAMP in tight corners

Fig. 5: Performance in tight-corner scenarios. RRT* produces unsafe waypoints; SBAMP commits only to the immediately reachable target.



(a) Pre-recovery state (b) Post-recovery state

Fig. 6: SBAMP recovery under combined large translational and rotational displacement.

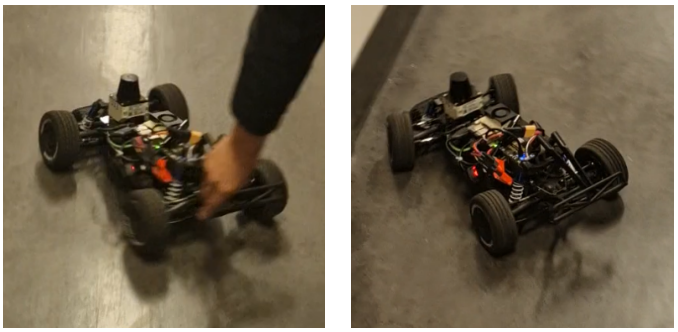
SBAMP's recovery is not limited to small perturbations: even under large translational and rotational displacements

(Figure 6), the SEDS attractor guides the vehicle back into the connected free-space corridor, at which point a new global trajectory is computed and followed without discontinuity.

C. Real-Time Performance Validation on Hardware

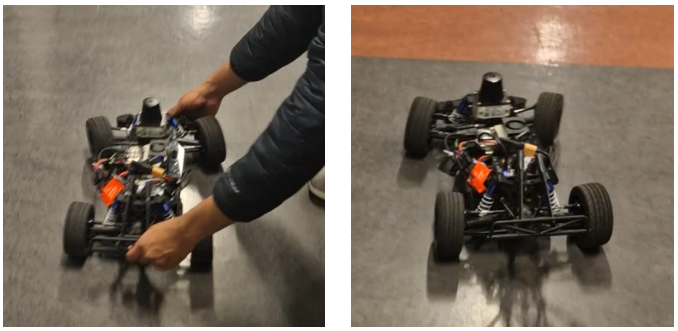
Simulation results establish SBAMP's theoretical properties; hardware trials establish that these properties transfer to a physical platform subject to sensor noise, actuation lag, and unmodeled dynamics. We operated the vehicle on an indoor loop course featuring straightaways, tight turns, and cluttered corridors, and manually applied 20 randomized translational and rotational disturbances during closed-loop operation.

In all trials, the vehicle deviated from its nominal path immediately following the disturbance, after which the SEDS attractor generated commands that returned it to the vicinity of the last RRT* waypoint before transitioning seamlessly to the newly computed global plan (Figures 7–8). SBAMP achieved a near-100% recovery rate across all 20 perturbations; the isolated failures arose only under extreme rotational displacements sufficient to cause waypoint misidentification, and even in these cases the vehicle converged to the most recently valid waypoint rather than diverging.



(a) Post-disturbance deviation (b) Post-recovery trajectory

Fig. 7: SBAMP response to human-applied rotational disturbance on hardware.



(a) Post-disturbance deviation (b) Post-recovery trajectory

Fig. 8: SBAMP response to human-applied translational disturbance on hardware.

Finally, Figures 9–10 demonstrate real-world obstacle avoidance under two drift scenarios. In each case SBAMP generated a collision-free avoidance trajectory around an unexpected object and rejoined the nominal corridor, without any modification to the underlying RRT* planner. This non-invasive augmentation is a key property of the framework: when RRT* operates correctly, SBAMP defers to it; when the planner falters, the DS attractor intervenes autonomously.

(a) Approaching obstacle (A) (b) Left-side avoidance (A)

Fig. 9: Obstacle avoidance scenario A.

(a) Approaching obstacle (B) (b) Right-side avoidance (B)

Fig. 10: Obstacle avoidance scenario B.

V. Conclusion

We have introduced SBAMP, a bi-level motion-planning framework that non-invasively augments RRT* with a Lyapunov-stable dynamical-systems controller, achieving on-the-fly adaptation with no prior training data. By converting each RRT* waypoint into a locally stable attractor on-the-fly, SBAMP ensures a valid control reference even when global replanning lags. Our threefold evaluation 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 MPC or MPPI, embedding obstacle-repulsive modulation directly into the dynamical-systems layer to reduce reliance on occupancy-grid update rates, and extending the framework 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, "F1tenth: 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 lpv-mpc for autonomous lane keeping," *arXiv preprint arXiv:2210.02971*, 2022.

[12] D. S. Karachalios and H. S. Abbas, "Efficient nonlinear mpc by leveraging lpv 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

APPENDIX A
SBAMP IMPLEMENTATION ON ROBORACER

SBAMP is deployed on the RoboRacer [4] platform using ROS2 Humble. Laser scans and odometry feed into a local occupancy grid; the planner produces a waypoint sequence τ; the SEDS controller issues velocity commands; and an optional visualization node renders the state in RViz2.

Five ROS2 nodes form the backbone of SBAMP: an *Occupancy Grid Node* fusing LIDAR and odometry; a *Next Waypoint Node* extracting the next feasible goal; an *RRT* Node* continuously replanning collision-free paths; an optional *Visualization Node* for RViz2 debugging; and the *SBAMP Node* fitting the SEDS model and publishing Ackermann drive commands at high frequency. The five core nodes are listed in Table I.

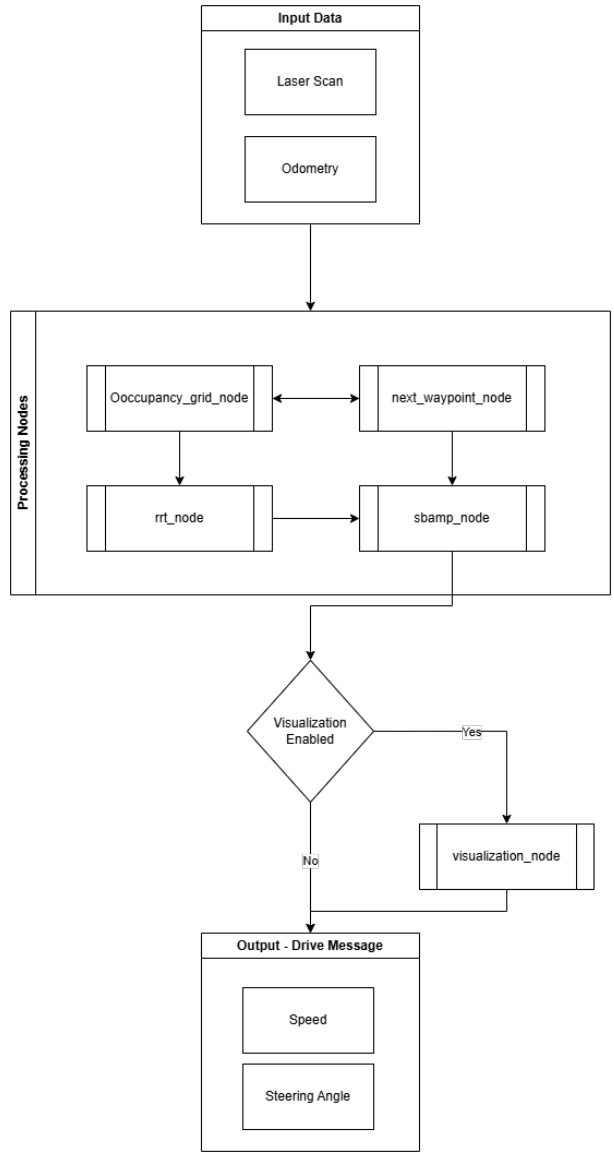


Fig. 11: ROS2 node graph for SBAMP on RoboRacer.

TABLE I: Core ROS2 Nodes in the `sbamp` Package

Node	Functionality
`occupancy_grid_node`	LIDAR/odometry fusion
`rrt_node`	Online path planning
`next_waypoint_node`	Feasible goal extraction
`sbamp_node`	SEDS fit and Ackermann control
`visualization_node`	RViz2 rendering (optional)

All experiments were performed on the RoboRacer F1/10 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 Ackermann steering commands.