Got it 👍 Here’s the report in English, structured and academic-style.

Report: Study of the Differences Between the Classic Navigation Stack and Navigation with Reinforcement Learning

1. Introduction

Autonomous navigation is a fundamental challenge in robotics and self-driving systems. Over the years, navigation techniques have evolved from classic approaches—which rely on deterministic algorithms and predefined models—to machine learning-based methods, particularly Reinforcement Learning (RL), where agents learn optimal navigation strategies through interaction with the environment.

This report highlights the key differences between the classic navigation stack and navigation with reinforcement learning, focusing on methodology, adaptability, performance, and limitations.

---

2. Classic Navigation Stack

The classic navigation stack is commonly used in robotic frameworks such as ROS (Robot Operating System). It is composed of well-defined modules, each responsible for a different aspect of navigation:

- Mapping and Localization: Algorithms such as SLAM (Simultaneous Localization and Mapping), AMCL (Adaptive Monte Carlo Localization).
- Global Planning: Graph-search or sampling-based planners (e.g., A*, Dijkstra, RRT).
- Local Planning and Control: Trajectory optimization and reactive obstacle avoidance (e.g., DWA – Dynamic Window Approach).
- Sensor Integration: Fusion of lidar, odometry, and IMU data for reliable state estimation.

Strengths:

- Deterministic, predictable behavior.
- Well-tested and robust in structured environments.
- High interpretability and ease of debugging.

Limitations:

- Requires precise environment models.
- Limited adaptability to dynamic, uncertain, or unstructured scenarios.
- Hand-crafted cost functions and parameters need extensive tuning.

---

3. Navigation with Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm where an agent learns navigation policies by trial-and-error interactions with the environment, guided by a reward function.

- Learning Framework: The robot perceives its state (e.g., position, lidar scan, goal direction) and chooses an action (move forward, turn, stop). The outcome provides a reward (e.g., closer to goal = positive, collision = negative).
- Policy Representation: Neural networks approximate the mapping from observations to actions.
- Training: Conducted in simulation (to avoid collisions) and sometimes fine-tuned in the real world.

Strengths:

- High adaptability to dynamic and complex environments.
- Can learn sophisticated navigation strategies without explicit modeling.
- Potential for emergent behaviors that outperform hand-coded planners.

Limitations:

- Requires massive training data and computational resources.
- Less predictable and harder to interpret compared to classical methods.

- Risk of poor generalization if training scenarios differ from real-world conditions.

---

4. Comparative Analysis

Aspect	Classic Navigation Stack	Reinforcement Learning Navigation
Methodology	Rule-based, model-driven	Data-driven, reward-based learning
Adaptability	Limited to predefined models	High adaptability to new environments
Performance	Stable in structured settings	Can excel in dynamic/complex environments
Data Requirement	Low (maps + sensors)	High (large training datasets)
Interpretability	Transparent, easy to debug	Black-box, harder to interpret
Computational Cost	Low to moderate	High (especially during training)
Real-world Reliability	Proven and robust	Still under research, less reliable without careful training

---

5. Conclusion

The classic navigation stack remains the most reliable and widely adopted solution in real-world robotic applications, particularly where environments are structured and predictable. On the other hand, reinforcement learning-based navigation offers exciting potential for highly adaptive and intelligent behaviors in unstructured or dynamic settings, but it still faces challenges in training cost, generalization, and reliability.

A promising direction is hybrid approaches, where classical planners provide safety and reliability, while reinforcement learning augments adaptability and decision-making in uncertain situations.