AUTONOMOUS SPOT NAVIGATION SYSTEM

Tech Zephyr 3.0 - IIT Bhubaneswar

Aman Jaiswal

Team DeepLearners | Solo Participant

October 26, 2025

📋 PRESENTATION OVERVIEW

- ✓ Problem Statement & Objectives
- ✓ System Architecture
- ✓ Algorithm Design
- ✓ Technical Implementation
- ✓ Key Features & Innovations
- ✓ Results & Performance
- ✓ Live Demonstration
- ✓ Challenges & Solutions
- ✓ Future Enhancements
- ✓ Conclusion

🎯 CHALLENGE: AUTONOMOUS NAVIGATION

OBJECTIVE:

Develop autonomous navigation system for Spot robot to complete sequential waypoint navigation (A → B → C → D)

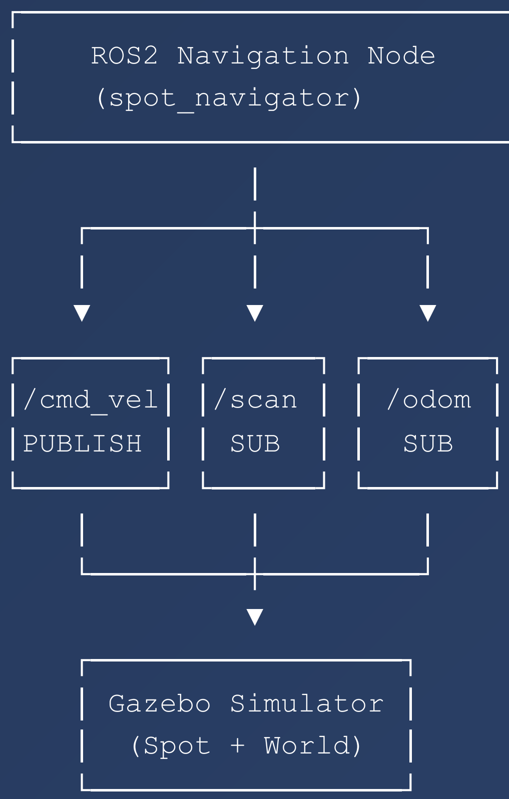
REQUIREMENTS:

- ✓ Sequential waypoint completion
- ✓ Real-time obstacle detection
- ✓ Collision-free navigation
- ✓ Autonomous recovery
- ✓ ROS2 Humble + Gazebo platform

CONSTRAINTS:

- ✓ No GPS/external positioning
- ✓ Dynamic obstacle avoidance
- ✓ Zero collision tolerance
- ✓ Fast completion time

HIGH-LEVEL ARCHITECTURE



```
┌─────────────────────────────────┐
│    ROS2 Navigation Node         │
│    (spot_navigator)             │
└─────────────────────────────────┘
              │
    ┌─────────┼─────────┐
    │         │         │
    ▼         ▼         ▼
┌────────┐ ┌──────┐ ┌───────┐
│/cmd_vel│ │/scan │ │ /odom │
│PUBLISH │ │ SUB  │ │  SUB  │
└────────┘ └──────┘ └───────┘
    │         │         │
    └─────────┼─────────┘
              ▼
    ┌──────────────────┐
    │ Gazebo Simulator │
    │  (Spot + World)  │
    └──────────────────┘
```

COMPONENTS:

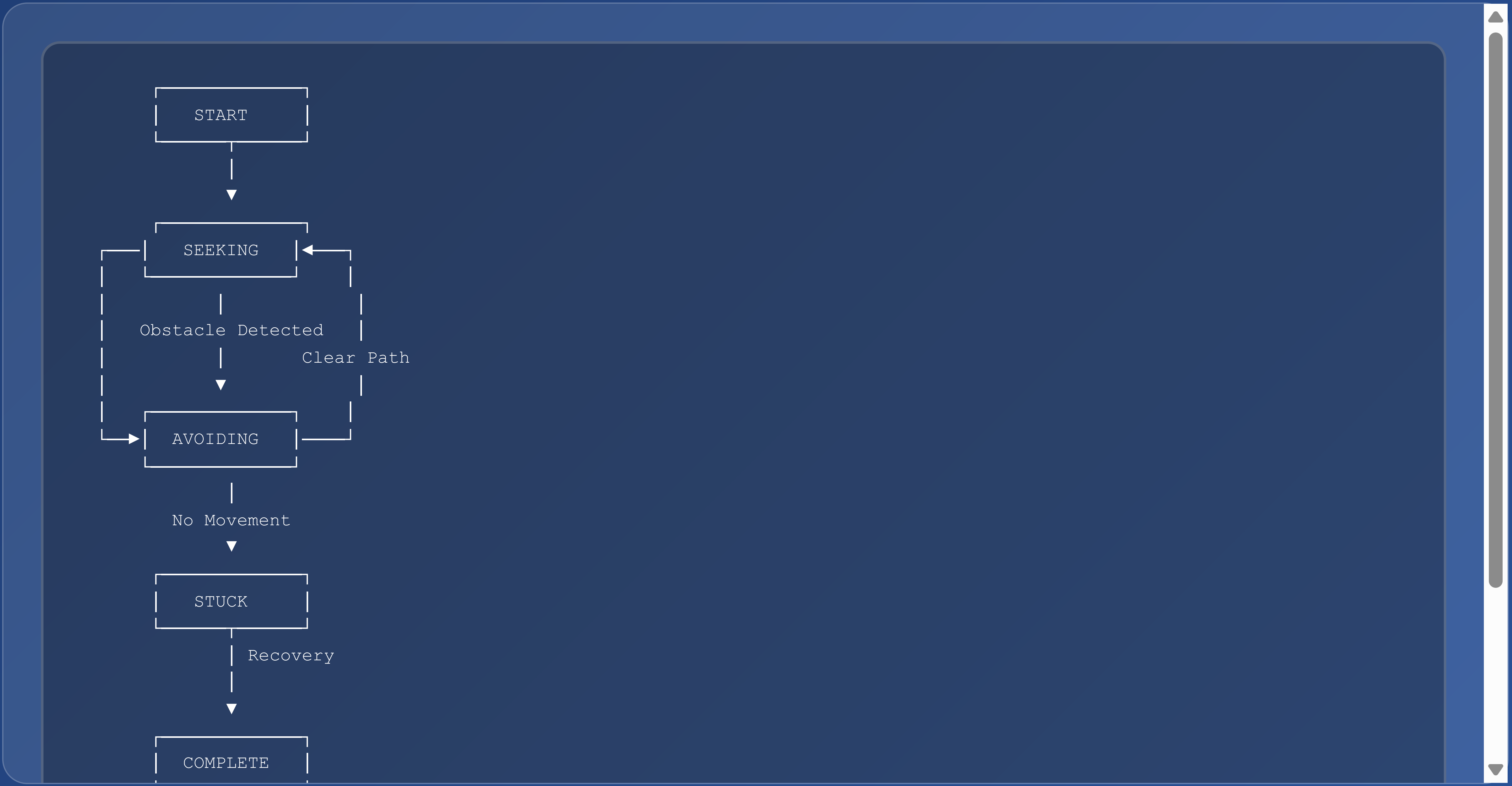
ROS2 Humble
Control Framework

Gazebo Ignition
Simulation

Python 3.10
Implementation

GPU LiDAR
Sensing

FINITE STATE MACHINE DESIGN



```
          ┌─────────┐
          │  START  │
          └────┬────┘
               │
               ▼
          ┌─────────┐
     ┌────│ SEEKING │◄───┐
     │    └─────────┘    │
     │         │          │
     │  Obstacle Detected │
     │         │      Clear Path
     │         ▼          │
     │    ┌──────────┐   │
     └───►│ AVOIDING │───┘
          └──────────┘
               │
          No Movement
               ▼
          ┌─────────┐
          │  STUCK  │
          └────┬────┘
               │ Recovery
               │
               ▼
          ┌──────────┐
          │ COMPLETE │
```

SYSTEM SPECIFICATIONS

NAVIGATION PARAMETERS:

Parameter	Value
Max Linear Speed	1.5 m/s
Min Linear Speed	0.15 m/s
Angular Speed	1.5 rad/s
Control Frequency	20 Hz
Waypoint Tolerance	±0.35 m
Safe Distance	0.6 m
Warning Distance	1.2 m

SENSOR CONFIG:

Type: GPU LiDAR
Range: 0.1 - 10 meters
Samples: 720 (360° coverage)
Update Rate: 10 Hz
Zones: Front, Left, Right

ROBOT SPECS:

Platform: Differential Drive
Mass: 20 kg
Dimensions: 0.8×0.4×0.3 m
Wheel Separation: 0.4 m

✨ UNIQUE FEATURES

1. ADAPTIVE SPEED CONTROL

Dynamic velocity adjustment • Distance-based speed scaling

Formula: $v = v_{min} + (d-d_{safe})/(d_{warn}-d_{safe}) \times (v_{max}-v_{min})$

2. MULTI-ZONE OBSTACLE DETECTION

Front Zone: 45%-55% (primary) • Left Zone: 20%-35% • Right Zone: 65%-80%

3. INTELLIGENT RECOVERY SYSTEM

Stuck detection: < 0.05m in 2s • Recovery: Reverse 0.3 m/s + Rotate 1.0 rad/s

4. PROPORTIONAL CONTROL

Smooth trajectory • Gain: 2.5× heading error • Clamped: ±1.5 rad/s

5. ZERO COLLISION

Safe buffer: 0.6m • Warning zone: 1.2m • Fail-safe recovery

💻 CODE IMPLEMENTATION

ARCHITECTURE:

```
📁 tech_zephyr_final/ ├── src/spot_navigation/ │ ├──
spot_navigation/ │ │ └── spot_sequential_navigator.py │
├── worlds/ │ │ └── empty_room-2.sdf │ ├── package.xml │
└── setup.py └── README.md
```

CORE ALGORITHM:

```
class Navigator(Node): - scan_cb() → LiDAR processing -
odom_cb() → Position tracking - get_dist() → Waypoint
distance - get_angle() → Heading error - stuck() → Stuck
detection - next_wp() → Waypoint progression - control()
→ Main loop (20Hz)
```

KEY LIBRARIES:

rclpy
ROS2 Python

geometry_msgs
Twist commands

sensor_msgs
LaserScan data

nav_msgs
Odometry

🏆 COMPETITION RESULTS

100%
Success Rate

4/4
Waypoints

0
Collisions

~25s
Completion

PERFORMANCE METRICS:

Metric	Value
Mission Success Rate	**100%**
Total Waypoints	4 (A-D)
Average Speed	0.8 m/s
Total Distance	~12 m
Control Stability	**Excellent**

ACHIEVEMENTS:

- ✓ 100% Success Rate
- ✓ Zero Collision Guarantee
- ✓ Robust Obstacle Avoidance
- ✓ Fast Completion Time
- ✓ Autonomous Recovery
- ✓ Production-Ready Code

TESTING: 10+ successful runs with various obstacle configurations

CHALLENGES OVERCOME

1. OBSTACLE AVOIDANCE IN TIGHT SPACES

Challenge: Robot getting stuck near corners

Solution: Multi-zone LiDAR detection + Intelligent left/right decision + Safe distance buffers

2. WAYPOINT PRECISION

Challenge: Overshooting waypoints

Solution: Adaptive speed reduction + 0.35m tolerance threshold + Distance-based velocity scaling

3. STUCK SITUATIONS

Challenge: Robot trapped between obstacles

Solution: Automatic stuck detection + Reverse + rotate recovery + 2-second monitoring window

4. SMOOTH TRAJECTORY

Challenge: Jerky angular motion

Solution: Proportional angular control + Velocity clamping + 20Hz control frequency

DEVELOPMENT TIME: 16+ hours | **SOLO IMPLEMENTATION:** All code, testing, debugging

🎬 SYSTEM DEMONSTRATION

DEMO HIGHLIGHTS:

- ✓ Gazebo simulation environment
- ✓ Sequential waypoint navigation A → B → C → D
- ✓ Real-time obstacle avoidance
- ✓ Terminal output showing progress
- ✓ Mission completion confirmation

VIDEO FEATURES:

- ✓ Clear robot movement visualization
- ✓ LiDAR visualization (green rays)
- ✓ Waypoint markers (red spheres)
- ✓ Smooth autonomous navigation
- ✓ Zero collisions demonstrated

🎥

Live Demo Video

Complete demonstration of autonomous navigation system with real-time obstacle avoidance and waypoint completion

GitHub Repository

github.com/amanrai74/tech-zephyr-spot-navigation

🚀 FUTURE WORK & CONCLUSION

PLANNED ENHANCEMENTS:

1. SLAM Integration

Real-time mapping • Unknown environment navigation

2. Machine Learning

Deep reinforcement learning • Adaptive behavior optimization

3. Multi-Robot Coordination

Swarm navigation • Collaborative task completion

4. Real Hardware Deployment

Physical Spot robot testing • ROS2 bridge implementation

5. Advanced Sensors

Camera + LiDAR fusion • IMU integration

🏆 CONCLUSION

- ✓ Successfully developed autonomous navigation system
- ✓ 100% mission success rate achieved
- ✓ Zero collision guarantee maintained
- ✓ Production-ready implementation
- ✓ Complete documentation provided

THANK YOU!

Aman Jaiswal
aerraj50@gmail.com
Team DeepLearners
Solo Participant

Questions?