CrisisCritters: Autonomous Swarm of SAR Robots

E3: Casper Wong, Kobe Zhang, Akash Arun

18-500 Capstone Design, Spring 2024

Electrical and Computer Engineering Department

Carnegie Mellon University

Product Pitch

Search and Rescue teams struggle with various challenges when dealing with areas affected by natural disasters. These challenges include:

- A lack of communication infrastructure
- Understaffing and fatigue
- Harsh and unknown conditions

With our project, we explore the integration of sophisticated autonomous swarm robotics to enhance human Search and Rescue operations. Our autonomous robot swarm:

- Communicates locally with **0% packet loss** using TCP over LAN
- **Scales in numbers** and runs **collaborative search algorithm**
- Lasts **1 hour on battery** life with compute units lasting **over 6 hours**
- **Operates autonomously** and identifies **survivors with a 0.83 average precision**
- Records **VSLAM data** to visualize unknown terrain

System Architecture

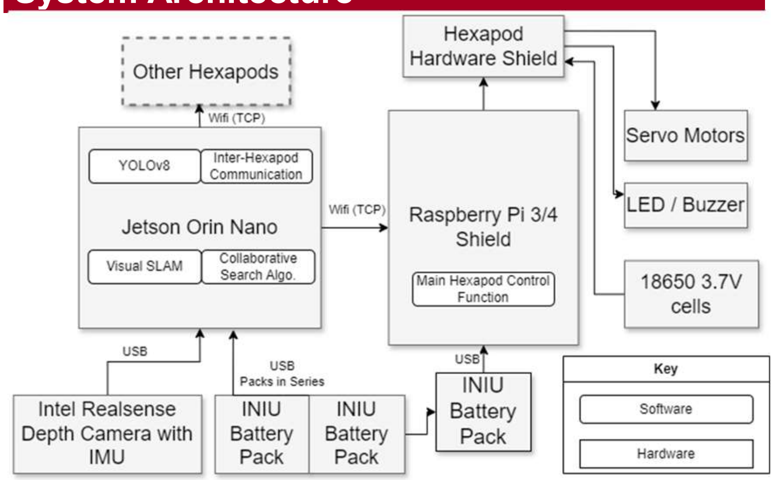

Figure 1. Hardware Block Diagram

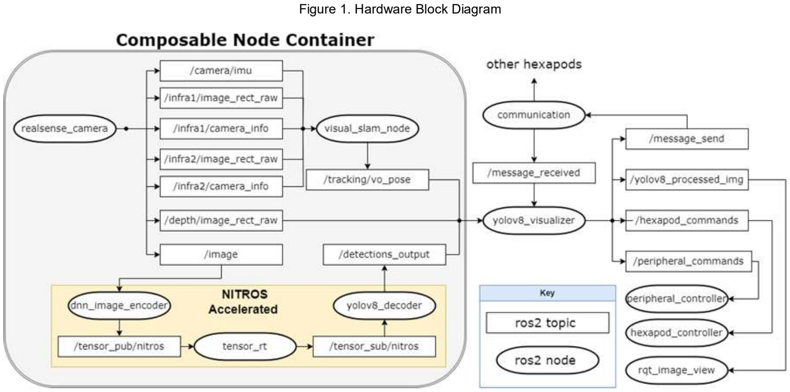

Figure 2. Software Block Diagram

Conclusions & Additional Information

Link to our website

SCAN ME

https://course.ece.cmu.edu/~ece500/projects/s24-teame3/

We believe that the integration of sophisticated autonomous robots like our project with current Search and Rescue operations can greatly enhance the success of SAR missions. By giving each robot object detection, SLAM, and local communication abilities, they can collaborate and search inaccessible locations efficiently. We hope to refine our robot behaviors so they can search with more speed and collect more data.

Through this project we learned the importance of **adaptability**, **deep understanding**, and leaving more **time for integration**.

System Description

On the hardware side, we are outfitting an off-the-shelf FreeNove hexapod robot, which is controlled via a Raspberry Pi and powered through 18650 3.7V battery cells, with a **Jetson Orin Nano** as the main computational unit. The Jetson Orin Nano will interface with an **Intel RealSense D435i Depth Camera** to perform perception tasks like object detection and SLAM. The Jetson Orin Nano is powered by two INIU battery packs wired in series and mounted on top of the Raspberry Pi Shield.

On the software side, we use **Isaac ROS** to leverage the Jetson Orin Nano's hardware acceleration capabilities. Our main software subsystems are **VSLAM** and **YOLOv8 Object Detection** and they are fed camera data through a RealSense camera node. The perception software are all contained with a multi-threaded composable node container and the tensor portion of **YOLOv8 is GPU accelerated through NITROS**. The yolov8_visualizer visualizes object detection and based on the object detection data combined with VSLAM orientation data, we control peripherals, the hexapod, and communicate with other hexapods.

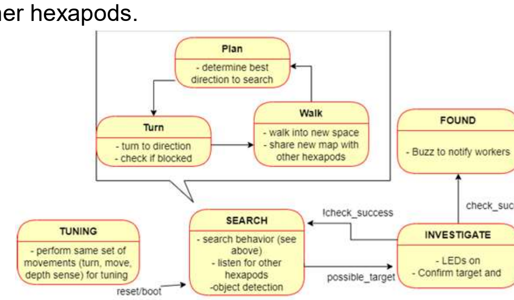

Figure 3. Software State Chart

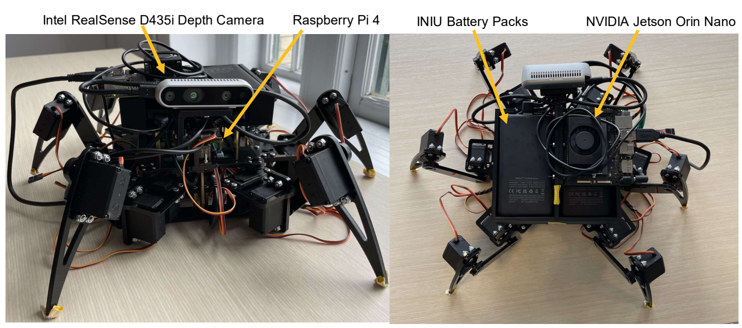

Figure 4 and Figure 5. Single finished hexapod, from the front and from above.

System Evaluation

- Communication Requirement: Met through 0% packet loss
- Identification Requirement: Met through >90% accuracy
- Scalability Requirement: Met through 2x speedup with 3 hexapods via simulation
- Battery Life: Met through ~ 1 hour battery life for full hexapod.

Design Tradeoffs

- **Object Detection Algo.**
  - YOLOv7: easier, less accurate
  - Isaac ROS YOLOv8: hardware accelerated, more accurate
- **Camera**
  - IMX219: cheaper, no depth
  - Intel RealSense D435i: depth

Battery Life Analysis

Number of Servos	18
Voltage (V)	4.8
Idle Current (A)	0.01
% Idling	60%
Stall Current (A)	1.2
% Stall	40%
Total Power (W)	41.990
Total Power (A at 7.4V)	**5.674**

Electrical & Computer ENGINEERING

Carnegie Mellon