Mobile Robotics for Computer Architects

Vijay Janapa Reddi, Ph. D.
Associate Professor

John A. Paulson School of Engineering and Applied Sciences
Domain Specific System Architecture (DoSSA) Workshop
Berkley campus (Kiwi)
Berkley campus (Kiwi)
20 Commercial Drone Use Cases

- Insurance Claim Validation
- Wind Turbine Inspection
- First Aid
- Agriculture
- Security
- Gas Flare Inspection
- Flash Flood Warning
- Organ Transplant Delivery
- Wildlife Conservation
- Railway Safety
- Construction Site Management
- Pipeline Leak Detection
- Oil Spill Monitoring
- Preventing Shark Attacks
- Shipping Emission Monitoring
- Cargo Delivery
- Cinematography
- Reforestation
- Journalism
- Search and Rescue
Global Market Size for Robotics

“… revenue generated from the robotics market globally, both industrial and non-industrial, from 2016 to 2022. In 2017, the robotics market is estimated to be worth 40 billion U.S. dollars globally. The industrial robotics market, which has traditionally represented the robotics industry … is giving way to non-industrial robots, such as personal assistant robots, customer service robots, autonomous vehicles, and unmanned aerial vehicles (UAVs).”

Source: [Statista 2018]
Global Market Size for Robotics

“... revenue generated from the robotics market globally, both industrial and non-industrial, from 2016 to 2022. In 2017, the robotics market is estimated to be worth 40 billion U.S. dollars globally. The industrial robotics market, which has traditionally represented the robotics industry ... is giving way to non-industrial robots, such as personal assistant robots, customer service robots, autonomous vehicles, and unmanned aerial vehicles (UAVs).”

Source: [Statista 2018]
What are the **challenges** facing aerial robots?
Performance (Real-time)

Reliability (Safety)

Power (Endurance)
Performance Depends on the Environment

Max Allowed Velocity (m/s)

<table>
<thead>
<tr>
<th>Sensor to Control</th>
<th>Process Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>
Performance Depends on the Environment

<table>
<thead>
<tr>
<th>Max Allowed Velocity (m/s)</th>
<th>Sensor to Control</th>
<th>Process Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Power Depends on the Physical Form Factor

87 kWh
Power Depends on the Physical Form Factor

87 kWh
Power Depends on the Physical Form Factor

87 kWh

0.08 kWh
Power Depends on the Physical Form Factor

87 kWh

1000X

0.08 kWh
Power Depends on the Physical Form Factor
Power Depends on the Physical Form Factor

![Graph showing battery capacity vs. size and endurance for drones.](image)
Power Depends on the Physical Form Factor
Safety Depends on Robustly Accounting for Failures
Performance (Real-time)

Reliability (Safety)

Power (Endurance)
What **architectural tools** are needed to enable research on aerial robots?
Traditional Computer Architecture Toolkit
Challenge with the Traditional Toolkit for Robotics
Challenge with the Traditional Toolkit for Robotics

There is a continuous feedback loop.
Simulation Infrastructure
Simulation Infrastructure

End-to-end Benchmarks

Evaluation Metrics
Simulation Infrastructure

End-to-end Benchmarks

Evaluation Metrics
The “World” of an Aerial Agent

- Application
- Algorithm
- Runtime
- Hardware

Environment
Understanding and Capturing the Interactions

Application

Algorithm

Runtime

Hardware

Environment
Understanding and Capturing the Interactions

- Ideally, each level should be readily explorable...
  - Independently
  - In relation to one another
Understanding and Capturing the Interactions

- Ideally, each level should be readily explorable...
  - Independently
  - In relation to one another

- Need development platforms
  - Enable design, prototyping, development and evaluation
  - Fast, accurate, reproducible, …
Closed-loop Simulation Setup
Closed-loop Simulation Setup

AirSim
Closed-loop Simulation Setup

AirSim

Unreal Engine
Closed-loop Simulation Setup

AirSim

Unreal Engine

Sensory Data (RGBD, GPS)
Closed-loop Simulation Setup

- AirSim
- Unreal Engine

Sensory Data (RGBD, GPS) → Workload
Closed-loop Simulation Setup

AirSim

Unreal Engine

Sensory Data (RGBD, GPS)

Workload

Robot Operating System
Closed-loop Simulation Setup

- AirSim
- Unreal Engine

Sensory Data (RGBD, GPS)

- Workload
- Robot Operating System
- Companion Computer
Closed-loop Simulation Setup

AirSim

Unreal Engine

Sensory Data (RGBD, GPS)

Workload

Robot Operating System

Companion Computer
Closed-loop Simulation Setup

AirSim
Unreal Engine

Sensory Data (RGBD, GPS)

Workload
Robot Operating System
Companion Computer

Flight Commands
Closed-loop Simulation Setup

AirSim
Unreal Engine

Sensory Data (RGBD, GPS)

Workload
Robot Operating System
Companion Computer

Flight Stack

Flight Commands
Closed-loop Simulation Setup

AirSim

Unreal Engine

Sensory Data (RGBD, GPS)

Workload
- Robot Operating System
- Companion Computer

Flight Stack
- Autopilot Controller

Flight Commands
Closed-loop Simulation **Setup**

AirSim

Unreal Engine

Sensory Data (RGBD, GPS)

Workload

Robot Operating System

Companion Computer

Flight Stack

Autopilot Controller

Flight Commands
Closed-loop Simulation Setup

AirSim
Unreal Engine

Sensory Data (RGBD, GPS)
Flight Control
Sensory Data (IMU)

Workload
Robot Operating System
Companion Computer

Flight Commands

Flight Stack
Autopilot Controller
Closed-loop Simulation **Knobs**

- AirSim
- Unreal Engine

**Sensory Data (RGBD, GPS)**

**Flight Control**

**Flight Stack**

- Workload
- Robot Operating System
- Companion Computer
- Flight Commands

- Autopilot Controller

**Autopilot Hardware**

**Flight Stack**

**Sensory Data (IMU)**
Closed-loop Simulation Knobs
Closed-loop Simulation **Knobs**

- **Sensory Data (RGBD, GPS)**
- **Flight Control**
- **Flight Stack**
- **Flight Commands**
- **Workload**
- **Robot Operating System**
- **Companion Computer**
- **Autopilot Controller**
- **Drones, batteries, …**
- **Environments (forest, urban, …)**

**Tools**
- AirSim
- Unreal Engine

**Software**
- ROS
Closed-loop Simulation **Knobs**

**Workload**
- Robot Operating System (ROS)
- Companion Computer
- Applications, kernels, ...

**Flight Stack**
- Autopilot Hardware
- Flight Control
- Sensory Data (IMU)

**Sensory Data**
- RGBD, GPS
- Environments (forest, urban, …)

**Flight Commands**
- Drones, batteries, …
Closed-loop Simulation **Knobs**

```
Drones, batteries, ...
Environments (forest, urban, ...)

Sensory Data (RGBD, GPS) → Flight Control

Applications, kernels, ...
Scheduling, mapping, ...
Companion Computer

Flight Commands

Flight Stack
Autopilot Controller

Sensory Data (IMU) → Flight Control
```
Closed-loop Simulation **Knobs**

- Drones, batteries, …
- Environments (forest, urban, …)
- Sensory Data (RGBD, GPS)
- Flight Control
- Flight Stack
- Autopilot Controller
- Applications, kernels, …
- Scheduling, mapping, …
- Platforms (TX2, Snapdragon, …)
- Flight Commands
- Sensory Data (IMU)
Closed-loop Simulation **Knobs**

- **Sensory Data** (RGBD, GPS)
- **Flight Control**
- **Flight Stack** (PX4, ArduPilot, …)
- **Autopilot Controller**
- **Applications, kernels, …**
- **Scheduling, mapping, …**
- **Platforms (TX2, Snapdragon, …)**
- **Drones, batteries, …**
- **Environments (forest, urban, …)**

**Flight Commands**
Closed-loop Simulation Knobs

- Drones, batteries, …
- Environments (forest, urban, …)
- Sensory Data (RGBD, GPS)
- Flight Control
- Flight Commands
- Flight Stack (PX4, ArduPilot, …)
- MAVLink ISA
- Applications, kernels, …
- Scheduling, mapping, …
- Platforms (TX2, Snapdragon, …)
Closed-loop Simulation Profiles

- Drones, batteries, …
- Environments (forest, urban, …)
- Sensory Data (RGBD, GPS)
- Flight Control
- Flight Commands
- Flight Stack (PX4, ArduPilot, …)
- MAVLink ISA
- Applications, kernels, …
- Scheduling, mapping, …
- Platforms (TX2, Snapdragon, …)
Closed-loop Simulation Profiles

Rotor energy, state of charge, ...
Environments (forest, urban, ...)

Sensory Data (RGBD, GPS)

Flight Control

Applications, kernels, ...
Scheduling, mapping, ...
Platforms (TX2, Snapdragon, ...)

Flight Commands

Flight Stack (PX4, ArduPilot, ...)

MAVLink ISA

Sensory Data (IMU)
Closed-loop Simulation Profiles

Rotor energy, state of charge, …
Collisions, Proximity misses, …

Sensory Data (RGBD, GPS)

Flight Control

Flight Commands

Applications, kernels, …
Scheduling, mapping, …
Platforms (TX2, Snapdragon, …)

Flight Stack (PX4, ArduPilot, …)

MAVLink ISA

Sensory Data (IMU)
Closed-loop Simulation Profiles

Rotor energy, state of charge, ...
Collisions, Proximity misses, ...

Sensory Data (RGBD, GPS)
Flight Control
Sensory Data (IMU)

Profile information (hot code, …)
Scheduling, mapping, …
Platforms (TX2, Snapdragon, …)

Flight Stack (PX4, ArduPilot, …)
MAVLink ISA
Closed-loop Simulation Profiles

- Rotor energy, state of charge, …
- Collisions, Proximity misses, …

Sensory Data (RGBD, GPS) → Flight Control

- Profile information (hot code, …)
- Throughput, bandwidth, …
- Platforms (TX2, Snapdragon, …)

Flight Stack (PX4, ArduPilot, …) → MAVLink ISA
Closed-loop Simulation Profiles
Closed-loop Simulation Profiles

Rotor energy, state of charge, …
Collisions, Proximity misses, …

Sensory Data (RGBD, GPS)

Profile information (hot code, …)
Throughput, bandwidth, …
Perf. Counters, Energy, …

Flight Control

Sensory Data (IMU)

Profile information
MAVLink ISA

Flight Commands
Closed-loop Simulation Profiles

Rotor energy, state of charge, …
Collisions, Proximity misses, …

Sensory Data (RGBD, GPS)

Flight Control

Profile information (hot code, …)
Throughput, bandwidth, …
Perf. Counters, Energy, …

Flight Commands

Profile information
Frequent commands, …

Sensory Data (IMU)
End-to-end Benchmarks

Simulation Infrastructure

Evaluation Metrics
Autonomous Package Delivery
Autonomous Package Delivery

Read depth sensors
Autonomous Package Delivery

- Read depth sensors
- Build 3D map of environment
Autonomous Package Delivery

- Read depth sensors
- Build 3D map of environment
- Sample environment
Autonomous Package Delivery

1. Read depth sensors
2. Build 3D map of environment
3. Sample environment
4. Try to find piecewise path
Autonomous Package Delivery

1. Read depth sensors
2. Build 3D map of environment
3. Sample environment
4. Try to find piecewise path
5. Post-process path
Autonomous Package Delivery

1. Read depth sensors
2. Build 3D map of environment
3. Sample environment
4. Try to find piecewise path
5. Smoothen piecewise path
6. Post-process path
Autonomous Package Delivery

1. Read depth sensors
2. Build 3D map of environment
3. Sample environment
4. Smoothen piecewise path
5. Try to find piecewise path
6. Post-process path
7. Fly!
   (SLAM, Obstacle Avoidance)
What the World Sees

What the Drone Sees
What the World Sees

Depth Camera

What the Drone Sees
What the World Sees

Depth Camera

What the Drone Sees

3D Map
What the World Sees

Depth Camera

Up to 3x speed up

3D Map

What the Drone Sees
What the World Sees

What the Drone Sees
What the World Sees

What the Drone Sees
MAVBench: Kernel Decomposition
# MAVBench: Kernel Decomposition

<table>
<thead>
<tr>
<th>Perception</th>
<th>Planning</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point Cloud Generation</strong></td>
<td><strong>Occupancy Map Generation</strong></td>
<td><strong>Collision Check</strong></td>
</tr>
<tr>
<td>Buffered</td>
<td>Real Time</td>
<td>GPS</td>
</tr>
<tr>
<td><strong>Scanning</strong></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Aerial Photography</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Package Delivery</strong></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>3D Mapping</strong></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Search and Rescue</strong></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
## MAVBench: Kernel Decomposition

### Perceptron

<table>
<thead>
<tr>
<th>Category</th>
<th>Scanning</th>
<th>Aerial Photography</th>
<th>Package Delivery</th>
<th>3D Mapping</th>
<th>Search and Rescue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Cloud Generation</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Occupancy Map Generation</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Collision Check</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Object Detection</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Tracking</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Localization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smoothened</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shortest Path</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frontier Exploration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lawn Mowing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Path Tracking/Command Issue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Planning

- Buffered
- Real Time

### Control

- Buffered
- Real Time
“Quality of Flight” Metrics

Start

End

Y(m)

−40
−20
0

X (m)

0
20
40
“Quality of Flight” Metrics
“Quality of Flight” Metrics

Trajectory
“Quality of Flight” Metrics

Trajectory

Energy
“Quality of Flight” Metrics

Trajectory
Energy
Mission time
“Quality of Flight” Metrics

- Trajectory
- Energy
- Mission time
- Velocity
“Quality of Flight” Metrics

Start

End

Trajectory
Energy
Mission time
Velocity
Jerk
“Quality of Flight” Metrics

Trajectory
Energy
Mission time
Velocity
Jerk
...
A Case Study of Power Efficiency

Performance

Reliability

Power
Where have my Joules gone?

![Graph showing power consumption over time for different modes: Arming Motor, Hovering, Flying, Landing.](chart)

- **Power (W) @ 5 m/s**
- **Power (W) @ 10 m/s**

- Time Step: 0, 2000, 4000, 6000
Where have my Joules gone?

Most of the energy is burned in the mechanical subsystem.
## Core/Frequency Sensitivity Analysis

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Average Velocity (m/s)</th>
<th>Mission Time (s)</th>
<th>Energy (kJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Frequency (GHz)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8 1.5 2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3D Mapping
Core/Frequency Sensitivity Analysis

<table>
<thead>
<tr>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Cores 2</td>
<td>0.34</td>
<td>0.65</td>
<td>0.75</td>
</tr>
<tr>
<td># of Cores 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Cores 4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average Velocity (m/s)

<table>
<thead>
<tr>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Time (s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Cores 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Cores 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Cores 4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Energy (kJ)

# of Cores

3D Mapping
Core/Frequency Sensitivity Analysis

### Average Velocity (m/s)

<table>
<thead>
<tr>
<th>Frequency (GHz)</th>
<th># of Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>4</td>
</tr>
<tr>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>2</td>
</tr>
</tbody>
</table>

### Mission Time (s)

<table>
<thead>
<tr>
<th>Frequency (GHz)</th>
<th># of Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>4</td>
</tr>
<tr>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>2</td>
</tr>
</tbody>
</table>

### Energy (kJ)

<table>
<thead>
<tr>
<th>Frequency (GHz)</th>
<th># of Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>4</td>
</tr>
<tr>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>2</td>
</tr>
</tbody>
</table>

3D Mapping
Core/Frequency Sensitivity Analysis

### Average Velocity (m/s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.34 0.65 0.75</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

### Mission Time (s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1388.1 676.8 530.5</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

### Energy (kJ)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>625.4 370 308.1</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

3D Mapping
## Core/Frequency Sensitivity Analysis

### Average Velocity (m/s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>0.34</td>
<td>0.65</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.49</td>
<td>0.86</td>
<td>1.07</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.72</td>
<td>0.92</td>
<td>1.12</td>
</tr>
</tbody>
</table>

### Mission Time (s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>1388.1</td>
<td>676.8</td>
<td>530.5</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>980.1</td>
<td>477.1</td>
<td>393.1</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>716</td>
<td>440.7</td>
<td>326.7</td>
</tr>
</tbody>
</table>

### Energy (kJ)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>625.4</td>
<td>370</td>
<td>308.1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>554.3</td>
<td>297.5</td>
<td>260.3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>438.1</td>
<td>269.4</td>
<td>223.9</td>
</tr>
</tbody>
</table>

3D Mapping
# Core/Frequency Sensitivity Analysis

## Average Velocity (m/s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>0.34</td>
<td>0.65</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.49</td>
<td>0.75</td>
<td>1.07</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.72</td>
<td>0.92</td>
<td>1.12</td>
</tr>
</tbody>
</table>

## Mission Time (s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>1388.1</td>
<td>676.8</td>
<td>530.5</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>980.1</td>
<td>477.1</td>
<td>393.1</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>716</td>
<td>440.7</td>
<td>326.7</td>
</tr>
</tbody>
</table>

## Energy (kJ)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>625.4</td>
<td>370</td>
<td>308.1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>554.3</td>
<td>297.5</td>
<td>260.3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>438.1</td>
<td>269.4</td>
<td>223.9</td>
</tr>
</tbody>
</table>

---

**3D Mapping**

---
# Core/Frequency Sensitivity Analysis

## Average Velocity (m/s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.34</td>
<td>0.65</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.49</td>
<td>0.75</td>
<td>1.07</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.72</td>
<td>0.92</td>
<td>1.12</td>
<td></td>
</tr>
</tbody>
</table>

## Mission Time (s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1388.1</td>
<td>676.8</td>
<td>530.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>980.1</td>
<td>471.1</td>
<td>393.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>716</td>
<td>440.7</td>
<td>326.7</td>
<td></td>
</tr>
</tbody>
</table>

## Energy (kJ)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>625.4</td>
<td>370</td>
<td>308.1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>554.3</td>
<td>297.5</td>
<td>260.3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>438.1</td>
<td>269.4</td>
<td>223.9</td>
<td></td>
</tr>
</tbody>
</table>

3D Mapping
## Core/Frequency Sensitivity Analysis

### Average Velocity (m/s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.34</td>
<td>0.65</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.49</td>
<td>0.75</td>
<td>1.07</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.72</td>
<td>0.92</td>
<td>1.12</td>
<td></td>
</tr>
</tbody>
</table>

### Mission Time (s)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1388.1</td>
<td>676.8</td>
<td>530.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>980.1</td>
<td>477.1</td>
<td>393.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>716</td>
<td>440.7</td>
<td>326.7</td>
<td></td>
</tr>
</tbody>
</table>

### Energy (kJ)

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>0.8</th>
<th>1.5</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>625.4</td>
<td>370</td>
<td>308.1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>554.3</td>
<td>297.5</td>
<td>260.3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>438.1</td>
<td>269.4</td>
<td>223.9</td>
<td></td>
</tr>
</tbody>
</table>

3D Mapping
## Core/Frequency Sensitivity Analysis

<table>
<thead>
<tr>
<th># of Cores</th>
<th>Frequency (GHz)</th>
<th>Average Velocity (m/s)</th>
<th>Mission Time (s)</th>
<th>Energy (kJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.8</td>
<td>0.34</td>
<td>1388.1</td>
<td>625.4</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>0.65</td>
<td>676.8</td>
<td>370.0</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>0.75</td>
<td>530.5</td>
<td>308.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
<td>0.34</td>
<td>980.1</td>
<td>554.3</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>0.65</td>
<td>493.1</td>
<td>297.5</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>0.75</td>
<td>393.1</td>
<td>260.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>0.34</td>
<td>716</td>
<td>438.1</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>0.65</td>
<td>440.7</td>
<td>269.4</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>0.75</td>
<td>326.7</td>
<td>223.9</td>
</tr>
</tbody>
</table>

### 3D Mapping

- 3X improvement in speed, flight time and energy as compute capability increases
Recall: **Power Depends on Physical Form Factor**

![Graph showing battery capacity and endurance vs. size for different types of UAVs.](image)
Unmanned Aerial Vehicles (UAVs) are getting closer to becoming ubiquitous in everyday life. Among them, Micro Aerial Vehicles (MAVs) have seen an outburst of attention recently, specifically in the area with a demand for autonomy. A key challenge standing in the way of making MAVs autonomous is that researchers lack the comprehensive understanding of how performance, power, and computational bottlenecks affect MAV applications. MAVs must operate under a stringent power budget, which severely limits their flight endurance time. As such, there is a need for new tools, benchmarks, and methodologies to foster the systematic development of autonomous MAVs. In this paper, we introduce the “MAVBench” framework which consists of a closed-loop simulator and an end-to-end application benchmark suite. A closed-loop simulation platform is needed to probe and understand the intra-system (application data flow) and inter-system (system and environment) interactions in MAV applications to pinpoint bottlenecks and identify opportunities for hardware and software co-design and optimization. In addition to the simulator, MAVBench provides a benchmark suite, the first of its kind, consisting of a variety of MAV applications designed to enable computer architects to perform characterization and develop future aerial computing systems. Using our open source, end-to-end experimental platform, we uncover a hidden, and thus far unexpected compute to total system energy relationship in MAVs. Furthermore, we explore the role of compute by presenting three case studies targeting performance, energy and reliability. These studies confirm that an efficient system design can improve MAV’s battery consumption by up to 1.8X.

I. Introduction

Unmanned aerial vehicles (a.k.a drones) are becoming an important part of our technological society. With myriad use cases, such as in sports photography [1], surveillance [2], disaster management, search and rescue [3, 4], transportation and package delivery [5–7], and more, these unmanned aerial vehicles are on the cusp of demonstrating their full potential. Hence, drones are rapidly increasing in number. Between 2015, when the U.S. Federal Aviation Administration (FAA) first required every owner to register their drone, and 2017, the number of drones has grown by over 200%. At the time of writing, the FAA indicates that there are over 900,000 drones registered with the FAA drone registry database (Figure 1). By 2021, the FAA expects this number will exceed 4 million units [8]. Such an upward trend can be explained by the new opportunities that unmanned aerial vehicles are enabling.

Abstract—Unmanned Aerial Vehicles (UAVs) are getting closer to becoming ubiquitous in everyday life. Among them, Micro Aerial Vehicles (MAVs) have seen an outburst of attention recently, specifically in the area with a demand for autonomy. A key challenge standing in the way of making MAVs autonomous is that researchers lack the comprehensive understanding of how performance, power, and computational bottlenecks affect MAV applications. MAVs must operate under a stringent power budget, which severely limits their flight endurance time. As such, there is a need for new tools, benchmarks, and methodologies to foster the systematic development of autonomous MAVs. In this paper, we introduce the “MAVBench” framework which consists of a closed-loop simulator and an end-to-end application benchmark suite. A closed-loop simulation platform is needed to probe and understand the intra-system (application data flow) and inter-system (system and environment) interactions in MAV applications to pinpoint bottlenecks and identify opportunities for hardware and software co-design and optimization. In addition to the simulator, MAVBench provides a benchmark suite, the first of its kind, consisting of a variety of MAV applications designed to enable computer architects to perform characterization and develop future aerial computing systems. Using our open source, end-to-end experimental platform, we uncover a hidden, and thus far unexpected compute to total system energy relationship in MAVs. Furthermore, we explore the role of compute by presenting three case studies targeting performance, energy and reliability. These studies confirm that an efficient system design can improve MAV’s battery consumption by up to 1.8X.

Fig. 1: Rapidly growing interest in UAVs. Data mined from FAA vehicle registration. The number of FAA registrations increased by 2X over the past two years, and it is rapidly growing. The FAA projects that by 2021 the number will exceed 4M units [9].

The growth and significance of this emerging domain of autonomous agents call for architects attention. Challenges such as low endurance (how long the drone can last in the air) and small battery capacities for drones demand hardware and system architects’ attention. The limited on-board energy budget manifests itself in the limited endurance and range of drones. This can be seen in various off-the-shelf commercial drones where endurance is typically less than 20 minutes, and flight range is about 15 miles [6]. To practically deploy drones, both their endurance and range must be improved.

In this paper, we investigate and show the role of computing given the endurance and range challenges. For example, we show how a powerful compute subsystem can be deployed to mitigate the problem of limited endurance. The drone’s compute subsystem dictates how fast a drone can maneuver, fly, and efficiently finish its mission. Hence, a computing subsystem that takes a long time to do path planning while the drone is hovering in the air, results in the inefficient consumption of energy. Furthermore, a more powerful compute subsystem can lead to more intelligent decision making (e.g., shorter paths to take). It is important to note that enabling intelligence on drones is challenging because of the computational power, size, weight, and cooling limitations.

To enable research and investigation, the foremost challenge to address is the lack of systematic benchmarks and infrastructure for research. To address this shortcoming, we introduce MAVBench, the first of its kind, a platform for the holistic evaluation of aerial agents, involving a closed-loop simulation
The AI Revolution
The Traditional “PPC” Pipeline

- Perception
- Octomap Generation
- RRT
- Trajectory Smoothening
- Path Tracking
- Flight Stack (PX4)

Sensors
- IMU Data
- Camera Data

Compute
- SLAM
- Occupancy Map

Actuators
- Rotors Velocity
Deep Neural Nets
Autonomous Navigation

Autonomous navigation is a common kernel across multiple applications:

- Package Delivery
- Surveying
- 3D Mapping
- Search & Rescue
- Aerial Photography

Autonomous Navigation
Autonomous Navigation - Two Paradigms

- **Trajectory Planning**
- **Obstacle Avoidance**

**PPC**

**Pixel2Control**
AirLearning Infrastructure

AirSim

OpenAI Gym

State ($S_t$),
Reward ($R_t$)

Action ($A_t$)

Or

Ras Pi 3

Nvidia TX2
AirLearning Infrastructure

Simulation Environment

DRL Framework

Compute System

AirSim

OpenAI Gym

State ($S_t$), Reward ($R_t$)

Action ($A_t$)

Or

Ras Pi 3

Nvidia TX2
AirLearning Infrastructure

Simulation Environment

DRL Framework

Compute System

AirSim

State ($S_t$), Reward ($R_t$)

OpenAI Gym

Action ($A_t$)

Or

Ras Pi 3

Nvidia TX2

55
Simulation Environment

Sample 1
https://youtu.be/DMCj0bwHmds

Sample 1*
https://youtu.be/JRZ1lxHMWnc

Sample 2
https://youtu.be/YI2aaSSqZ0Y

Sample 2*
https://youtu.be/Oh3Hj0HsG50

Constrained Space

Increased Obstacle Density
Simulation Environment

Sample 1
https://youtu.be/DMCj0bwHmds

Sample 1*
https://youtu.be/JRZ1lxHMWnc

Sample 2
https://youtu.be/Yl2aaSSqZ0Y

Sample 2*
https://youtu.be/Oh3Hj0HsG50

Constrained Space

Increased Obstacle Density
AirLearning Infrastructure

Simulation Environment

AirSim

OpenAI Gym

State (S_t), Reward (R_t)

Action (A_t)

Ras Pi 3

or

Nvidia TX2

DRL Framework

Compute System
Deep RL Framework

UnReal + AirSim
OpenAI Gym
OpenAI Baselines
TensorFlow

Depth Image + RGB + IMU
Reinforcement Learning
Algorithms (DQN, DDQN, A3C)
Machine Learning Backend
Reinforcement learning based Drone Control

![Diagram showing the interactions between the Agent, Environment, State, Rewards, and Action Space. The Agent makes decisions based on the Environment's state, receiving rewards for actions taken. These rewards influence future actions, guiding the agent to optimize performance in the Action Space.](image-url)
AirLearning Demonstration

Demonstrates that the UAV has learned to navigate in a narrow passageway

https://www.youtube.com/watch?v=RIrkjnmXEAs&feature=youtu.be
AirLearning Demonstration

Demonstrates that the UAV has learned to navigate in a narrow passageway

https://www.youtube.com/watch?v=RlrkjmXEAs&feature=youtu.be
AirLearning Infrastructure

Simulation Environment

DRL Framework

Compute System

AirSim

OpenAI Gym

State ($S_t$), Reward ($R_t$)

Action ($A_t$)

Ras Pi 3

Nvidia TX2
Accelerator Design

ROS + Open Source Robotics Foundation = XPU
Accelerator Design

HiFive Unleashed + ROS + Open Source Robotics Foundation = XPU
Accelerator Design

Xilinx Vc707 + HiFive Unleashed + ROS = XPU
Accelerator Design

Xilinx Vc707 + PCIe Card + HiFive Unleashed + ROS = XPU
Accelerator Design

Xilinx Vc707 + PCIe Card + HiFive Unleashed = XPU

ROS (Open Source Robotics Foundation)
Algorithm Performance: Small vs. Large Policy

Small Policy

Large Policy
Algorithm Performance: Small vs. Large Policy
## MAVBench: Kernel Decomposition

<table>
<thead>
<tr>
<th>Perception</th>
<th>Planning</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Cloud Generation</td>
<td>Occupancy Map Generation</td>
<td>Collision Check</td>
</tr>
<tr>
<td>Scanning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerial Photography</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package Delivery</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3D Mapping</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Search and Rescue</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

- X indicates the presence or absence of a function in a specific category.
- The table represents the distribution of various tasks across different kernels.
- Each row corresponds to a specific task category, and each column represents a system function.
- The presence of an 'X' indicates that the task is supported by the respective function.
## MAVBench: Kernel Decomposition

<table>
<thead>
<tr>
<th>Scanning</th>
<th>Aerial Photography</th>
<th>Package Delivery</th>
<th>3D Mapping</th>
<th>Search and Rescue</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Perception</strong></td>
<td><strong>Planning</strong></td>
<td><strong>Control</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point Cloud Generation</td>
<td>Occupancy Map Generation</td>
<td>Collision Check</td>
<td>Object Detection</td>
<td>Object Tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>630</td>
<td>1</td>
<td>0</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>482</td>
<td>1</td>
<td>0</td>
<td>46</td>
</tr>
<tr>
<td>2</td>
<td>427</td>
<td>1</td>
<td>0</td>
<td>45</td>
</tr>
</tbody>
</table>
### MAVBench: Kernel Decomposition

<table>
<thead>
<tr>
<th>Perception</th>
<th>Planning</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Cloud Generation</td>
<td>Occupancy Map Generation</td>
<td>Collision Check</td>
</tr>
<tr>
<td>Scanning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerial Photography</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package Delivery</td>
<td>2</td>
<td>630</td>
</tr>
<tr>
<td>3D Mapping</td>
<td>2</td>
<td>482</td>
</tr>
<tr>
<td>Search and Rescue</td>
<td>2</td>
<td>427</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MAVBench: Kernel Decomposition

<table>
<thead>
<tr>
<th>Scanning</th>
<th>Perception</th>
<th>Planning</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerial Photography</td>
<td></td>
<td></td>
<td>89 1</td>
</tr>
<tr>
<td>Package Delivery</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>3D Mapping</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Search and Rescue</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Identify bottlenecks and accelerate them through domain specific logic.
Closed-loop simulation is important for the development of architectural solutions for autonomous machines.

End-to-end applications are needed to do hardware and software co-design to design domain specific accelerators.

RISC-V with ROS unlocks the potential of designing custom hardware accelerators that are open source to drive innovation.