Astrobee Robot Simulator Profiling

Jonathan Rojas González
jonathan.a.rojasgonzalez@nasa.gov

Universities Space Research Association (USRA)
Instituto Tecnológico de Costa Rica (TEC)
NASA Ames Research Center

October 30th, 2018

Outline

- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Outline

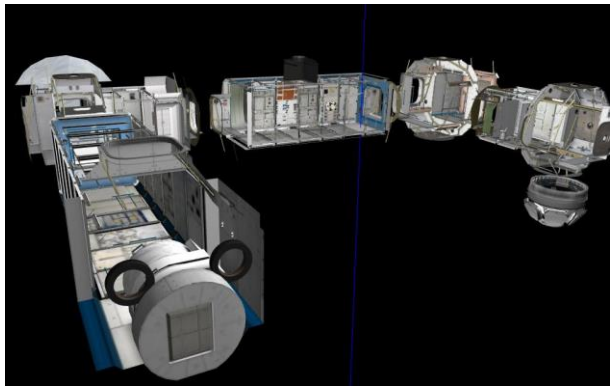
- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Background

Astrobee: a Research Platform

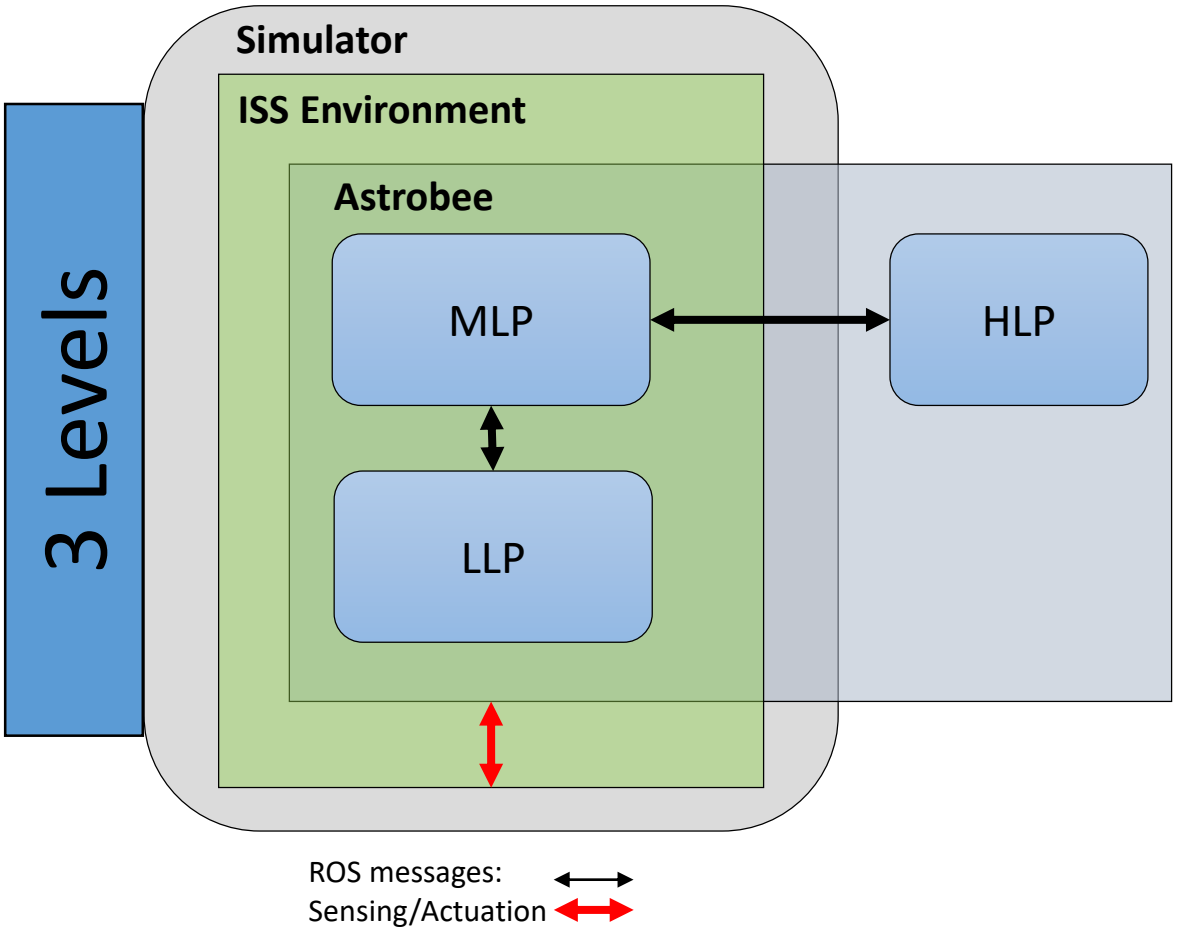
Simulator includes:

- Propulsion system and perching arm.
- Color and depth cameras.
- Inertial sensors and environments (ISS, Granite Lab)



Background

Astrobee Simulator



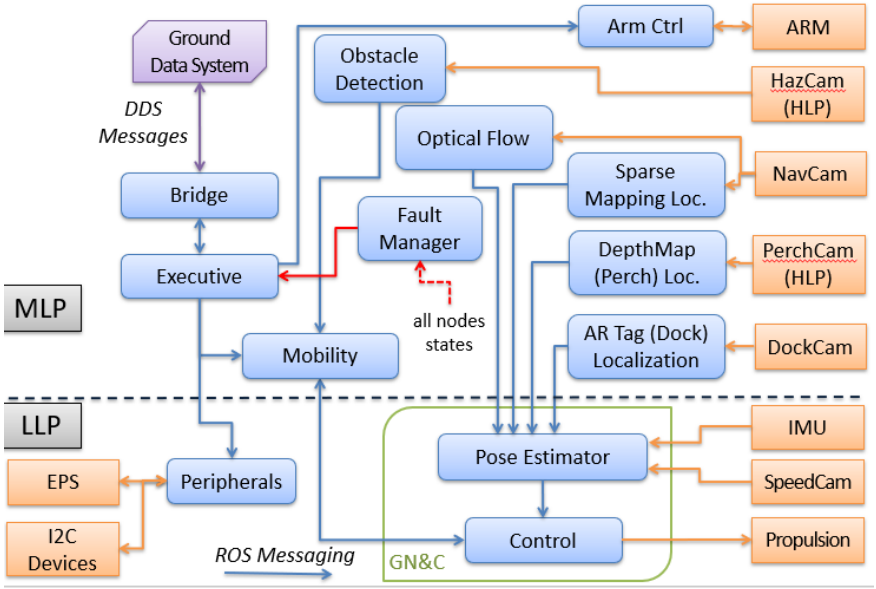
Three processors:

- High Level Processor
- Mid Level Processor
- Low Level Processor

Background

Astrobee Robot Software

- Relies extensively on the Robot Operating System (ROS)
- Reutilizes ROS messages to communicate all its nodes.
- Uses Gazebo Server and Viewer.



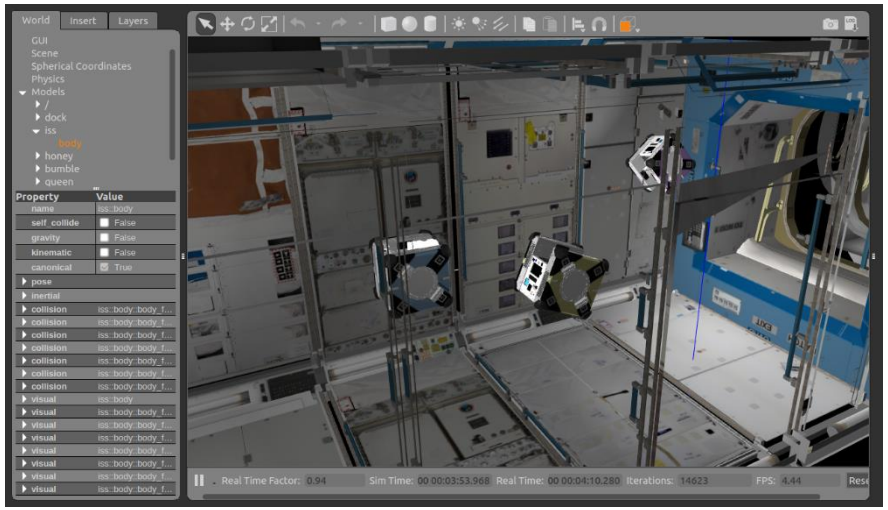
Outline

- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Problem Statement

Simulator Configurations

- Number of Robots
- Real Time Factor Speedup
- Instances of the simulator running in parallel.



Problem Statement

Beneficiaries:

MIT Zero Robotics (ZR) Competition:

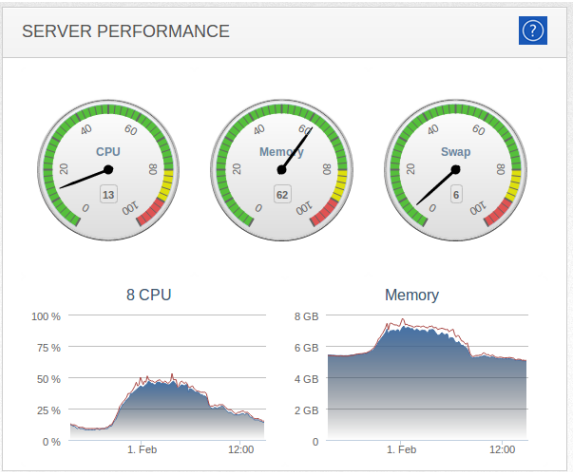
- Aims at inspiring middle/high school students solve code challenges using SPHERES in order to win a game and starting from 2020, using Astrobee.

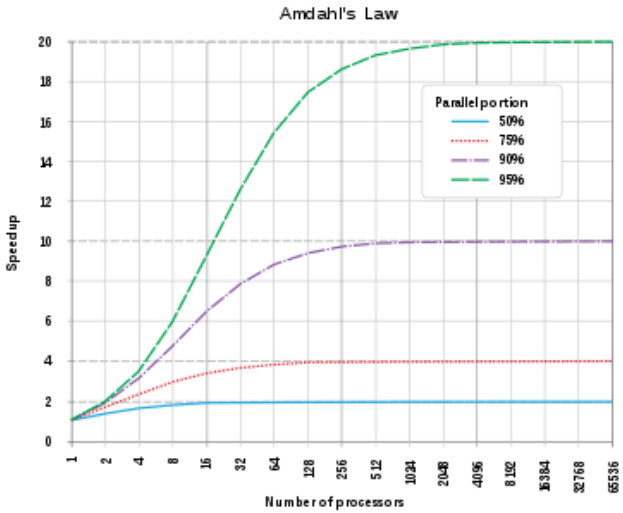
ZERO ROBOTICS
ISS PROGRAMING CHALLENGE

Problem Statement

It is required to adapt the current full-fledged simulator to have one that can be run with only essential components.

The simulator should be profiled in such way that users know the constraints of the simulations´ configurations as a requirement of the organization.



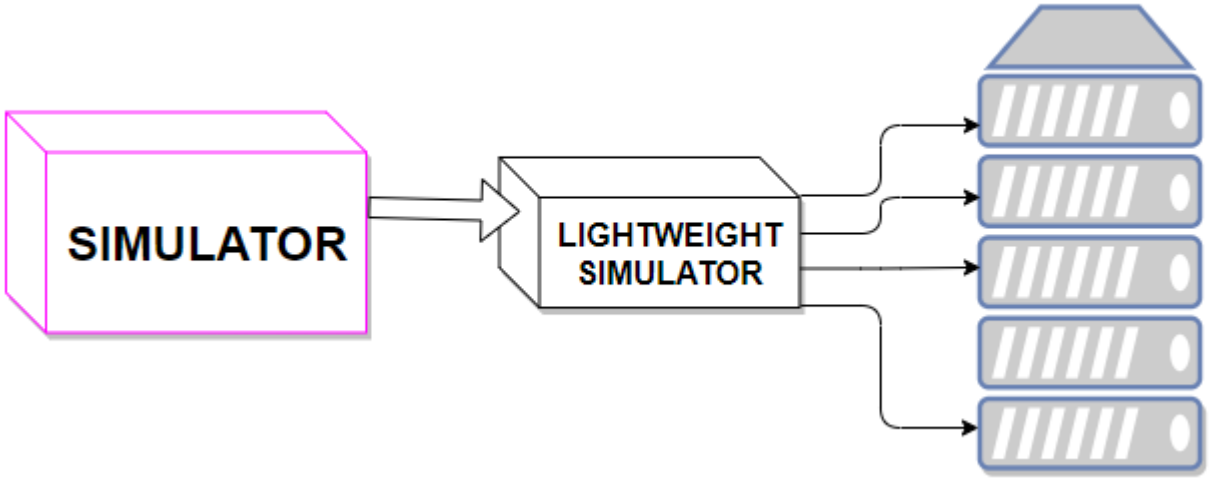


Outline

- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Goal

Adapt the Astrobee simulator software to provide a more flexible version through the simplification of components and the profiling of their behavior.



Objectives

- Beta-test and improve the setup process for Astrobee's Guest Science Android APKs.
- Establish a set of metrics to profile the Astrobee simulator.
- Create a mechanism for measuring these metrics.
- Analyze the data obtained to determine upper bounds of the simulator operation under different configurations.

Outline

- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Lightweight Simulator (I)

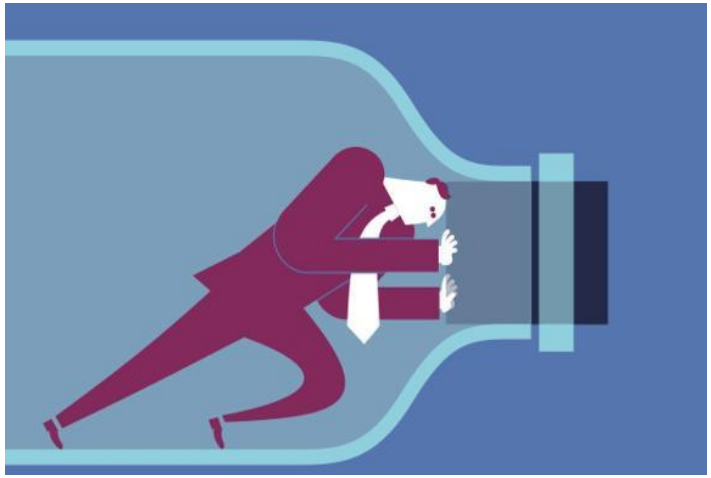
1. Familiarization and study of the simulator.
2. Identify components that can be removed and verify it runs correctly.
3. Evaluate the change in the speedup of the simulator's real time factor.

Lightweight Simulator (II)

Disable the Extended Kalman Filter (EKF)

+

Run simulator headless, (no Graphical User Interface)



Lightweight Simulator (III)

Before modifications:

- Maximum Real Time Factor Speedup: 2x

After modifications:

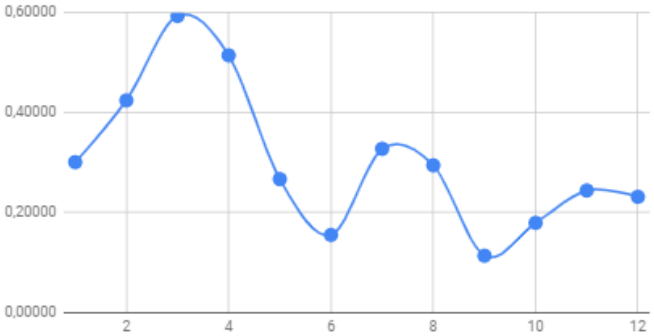
- Maximum Real Time Factor Speedup: 4x

Outline

- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Profiling

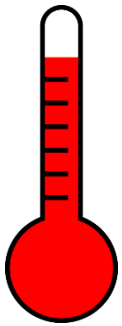
- Identify variables of interest.
- Develop scripts to gather metrics data.
- Analyze the obtained data and prepare graphics from the results.



Profiling

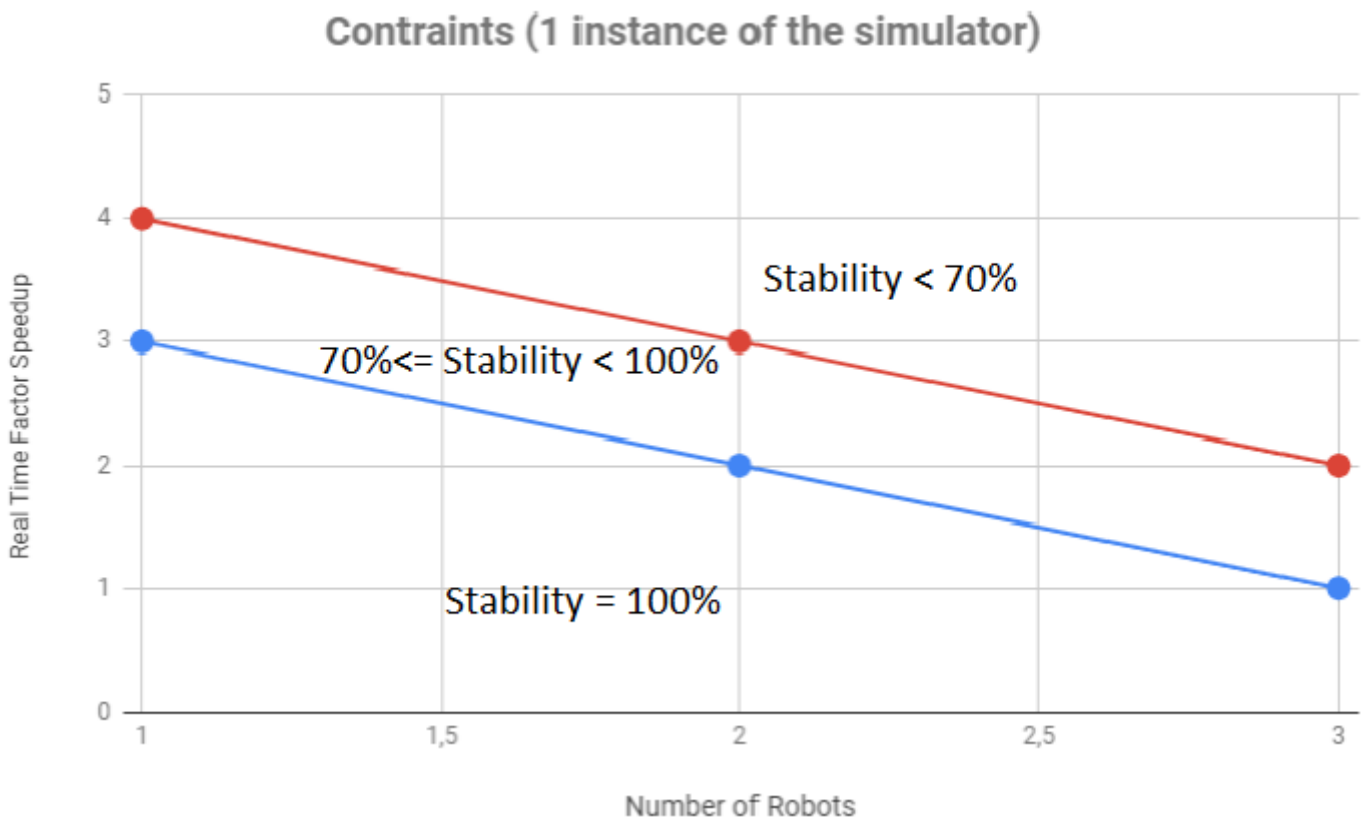
Metrics:

- Performance: CPU and RAM consumption.
- Accuracy: Error in Astrobee's final position.
- Real Time Factor Speed-up.
- CPU Temperature.
- Stability: an experiment that fails less than 30% of its total runs.



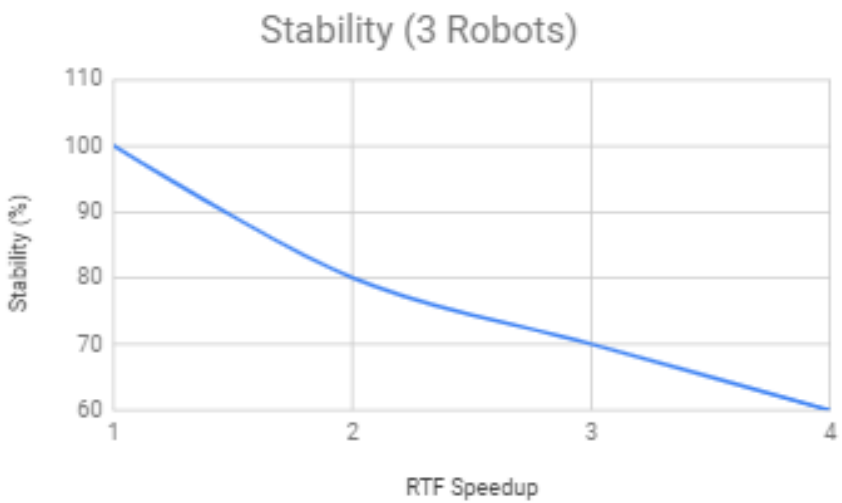
Profiling

Constraints:

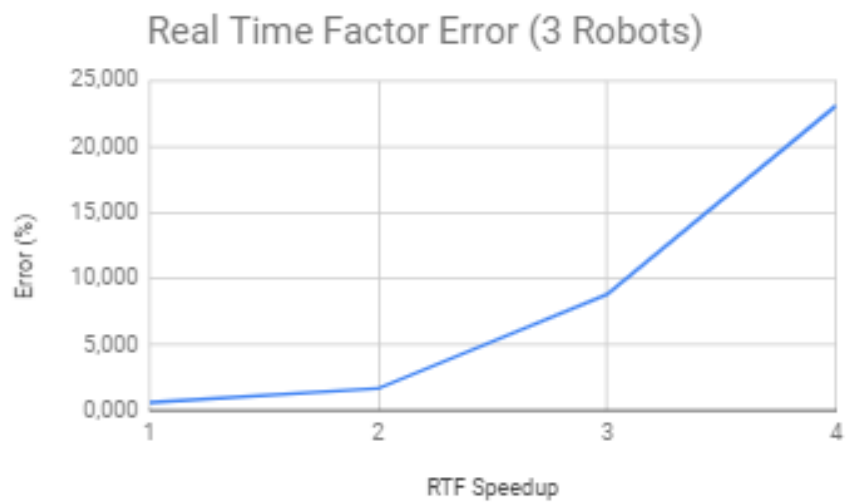


Profiling

Stability vs Real Time Factor (for 3 robots)



RTF Error vs RTF Speedup (for 3 robots)



Profiling

Cost Evaluation Function

- Data are normalized so they take values between 0-1 with 1 being the best.
- Assign weights to the Cost Evaluation Function.

Metrics	Weight
CPU	Cpu_w
RAM	RAM_w
Accuracy	Acc_w
Temperature	Temp_w
RTF	RTF_w

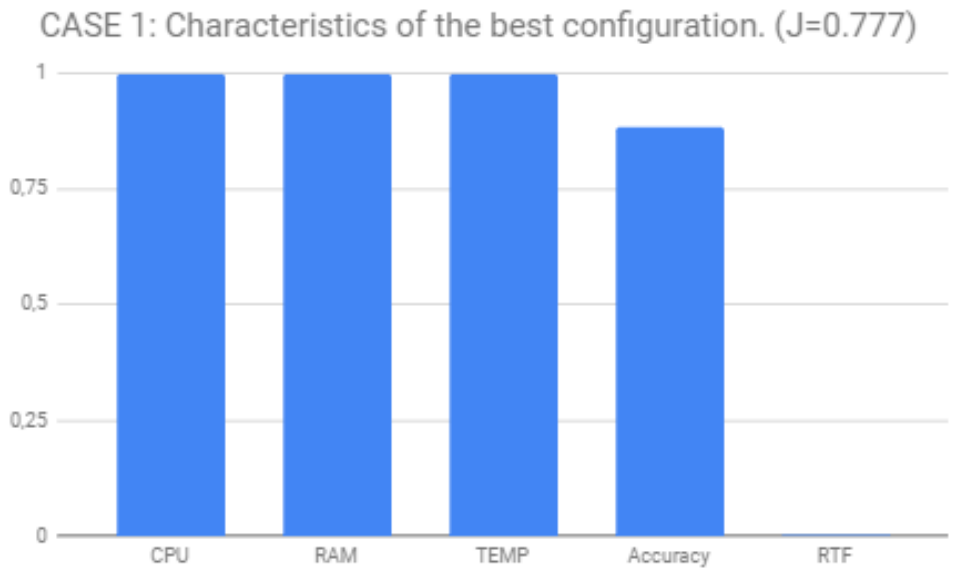
$$\mathrm{CEF}(\mathrm{x_{cpu}}, \mathrm{x_{ram}}, \mathrm{x_{ac}}, \mathrm{x_{temp}}, \mathrm{x_{rtf}}) = x_{cpu} * cpu_w + x_{ram} * ram_w + x_{ac} * ac_w + x_{temp} * temp_w + x_{rtf} * rtf_w$$

Profiling

Case 1: All metrics matter the same.

Configuration			
No.	**Instances**	**Robots**	**RTF Speedup**
1	1	1	1

Metrics	Weight
CPU	0,2
RAM	0,2
Accuracy	0,2
Temperature	0,2
RTF	0,2

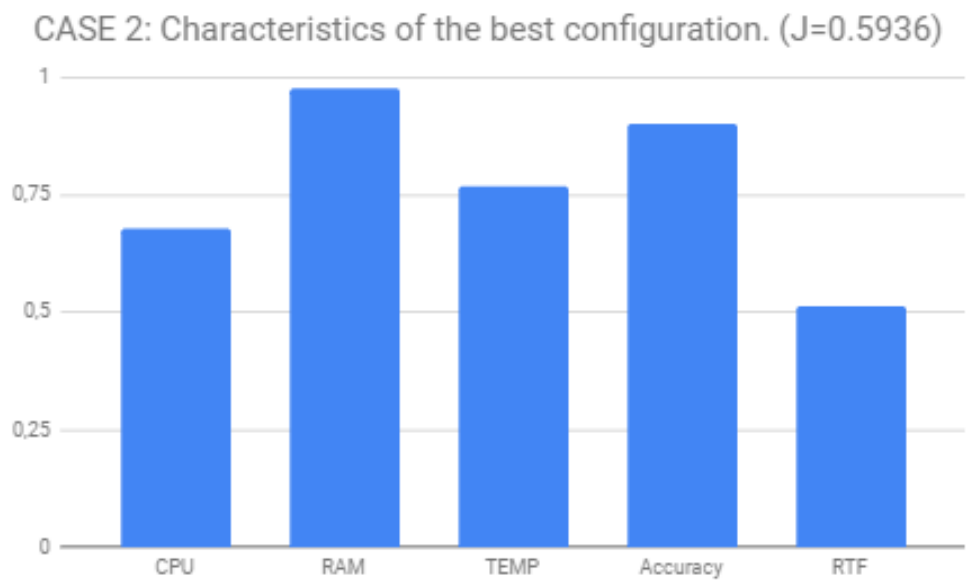


Profiling

Case 2: Focus on Real Time Factor.

Configuration			
No.	**Instances**	**Robots**	**RTF Speedup**
1	1	1	3

Metrics	Weight
CPU	0,1
RAM	0,1
Accuracy	0,3
Temperature	0,1
RTF	0,4



Outline

- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Conclusions

- Independently of the number of robots running, the tendency is that the stability decreases as the Real Time Factor speedup increases.
- Independently of the number of robots running, the tendency is that the Real Time Factor error increases as the Real Time Factor speedup increases.
- Most cases simulator users will have to pay for desired features by losing some other characteristics.
  - E.g. The greater RTF speedup is used, the more CPU consumption is needed.

Outline

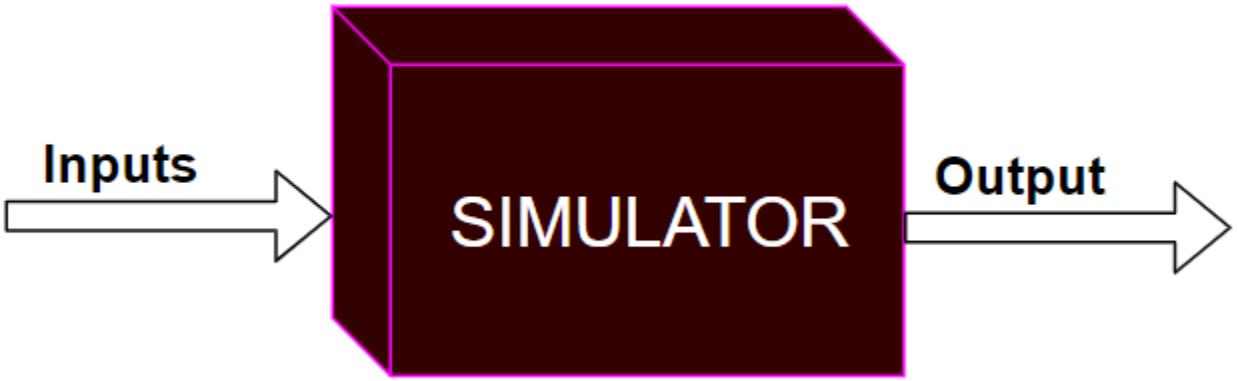
- Background
- Problem Statement
- Goal and Objectives
- Methodology and Results
  - Lightweight Simulator
- Methodology and Results
  - Profiling
- Conclusions
- Next Steps

Next Steps

- Do tests on computers with different hardware specifications so that it is possible to predict the constraints on other computers (Astrobeast server).

Next Steps

- Develop a wrapper application for providing the simulator to Guest Scientists as a black box.



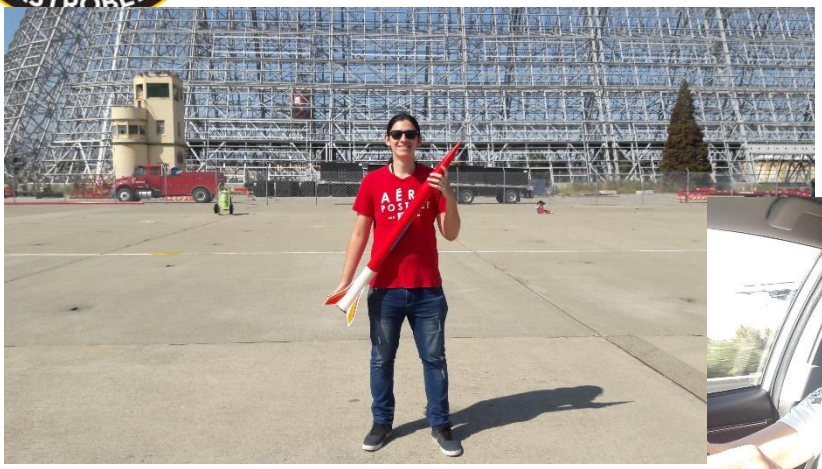
Acknowledgements

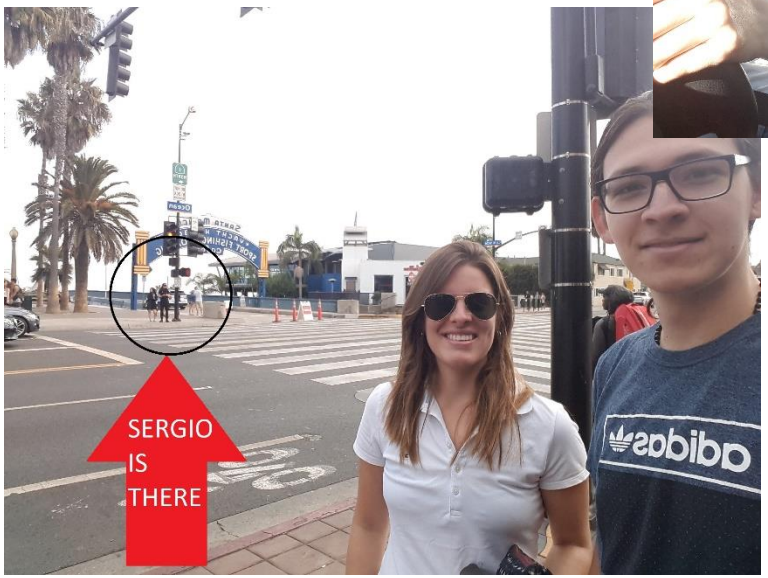
- Andres Mora Vargas
- Kathryn Hamilton
- Andrew Symington
- Brian Coltin
- Ted Morse
- Jose Benavides
- Jonathan Barlow
- All the Astrobee Facilities Group and Intelligent Robotics Group

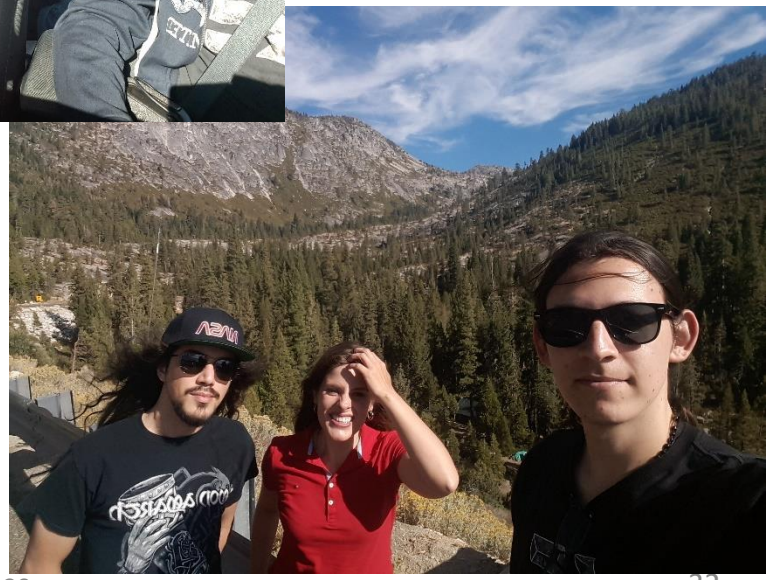
- Saba Hussain and all the USRA team.

- Milton Villegas Lemus
- Jennier Solano Cordero and TEC.

But there was more than that…







References

[1] L. Fluckiger, K. Browne, B. Coltin, J. Fusco, T. Morse and A. Symington, "Astrobee Robot Software: Enabling Mobile Autonomy on the ISS." [Online]. Available: http://brian.coltin.org/pub/fluckiger2018astrobee.pdf, iSARIAS 2018