Tradeoff Analysis using an Integrated Data-Driven and Model-Based Approach for the Design of Autonomous Robots

Sai Sandeep Damera, Praveen Kumar M.S., John S. Baras, Daniel R. Hunter

Institute for Systems Research
University of Maryland College Park

September 18, 2024
U.S. Army DEVCOM and SERC AI4SE & SE4AI Workshop 2024
Arlington, VA

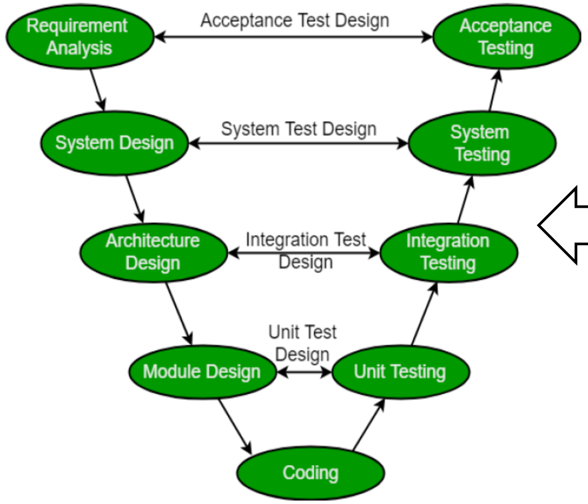
PROBLEM ADDRESSED AND SIGNIFICANCE

Systematic Methodology and Software Tool Suite for Trusted Autonomous Systems

Critical need for many US Army and DoD missions, and also many commercial applications

HOW

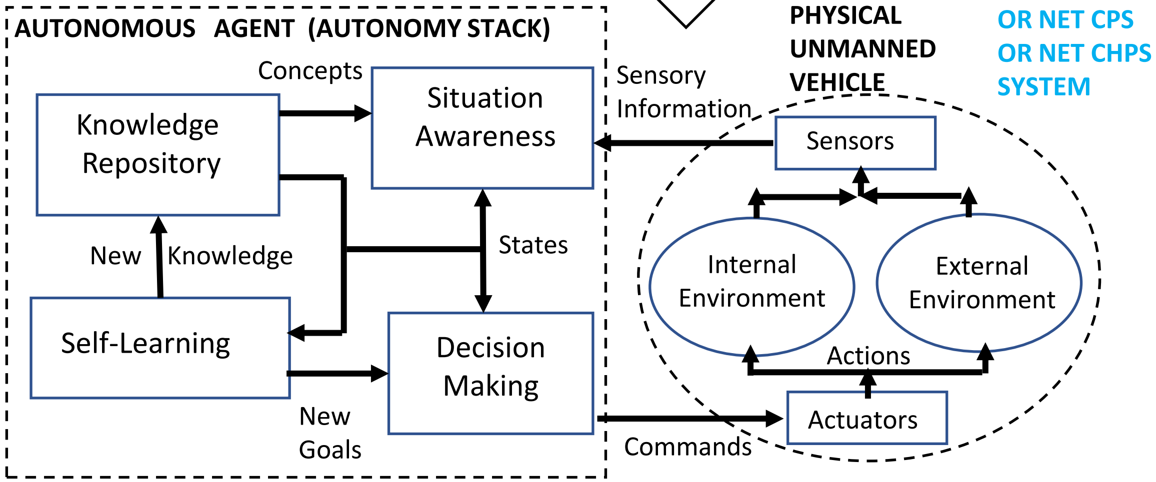
MBSE – Static, known Requirements, Verifiable



MBSE defeated by complexity and diversity of autonomous systems

Large Data Sets Driven ML - AI to the rescue: simulations, experiments, operations

Digital Twin or Testbed System



Design space exploration via tradeoffs to prioritize potential investments from portfolio of modules: sensors, actuators, cyber chips, materials, engines, architectures, algorithms, new technologies, etc.

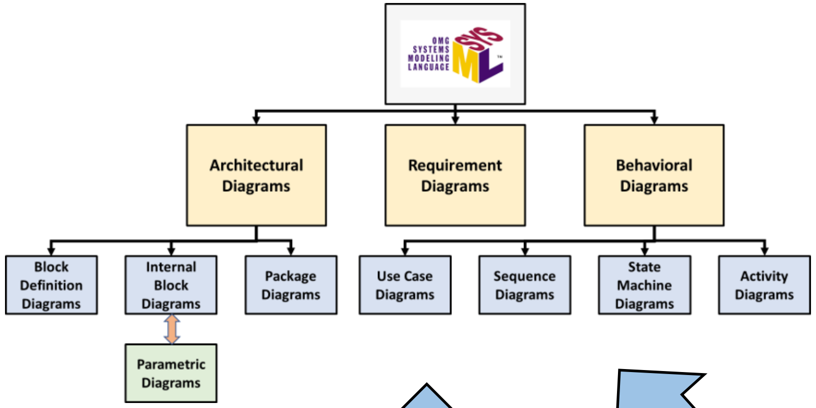
NOVELTY and VALUE

Integrating large data sets makes feasible the design of **high performance trustworthy autonomous systems** through empirical (DD) **and** formal (MBSE) validation, with changing requirements and scenarios.

Not possible otherwise. Currently major open problem.

Our Innovative Approach

SysML Models and Diagrams



VERITAS

LINK TO Formal Model Tools (UPPAAL, PRISM)
for Correct Task Execution, Timing analysis, Safety, Specification satisfaction, Robustness, Autonomy, Learning, Intelligence …

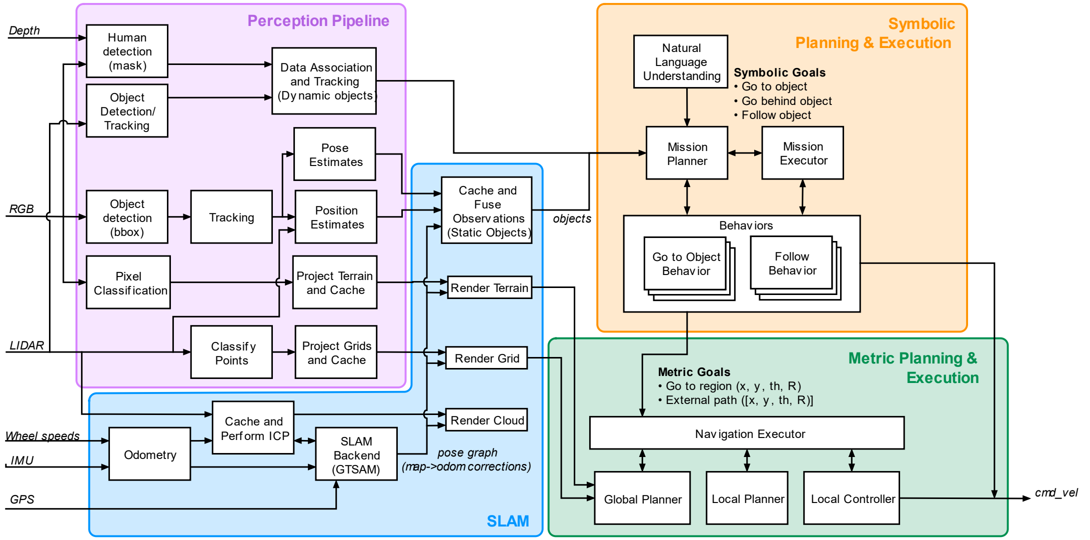
Mapping AS components to SysML models

PERFECT

TRADES-X

Design space exploration via tradeoffs to prioritize design decisions, investments, from portfolio of modules:
sensors, actuators, cyber chips, materials, engines,
algorithms, architectures, and new technologies.

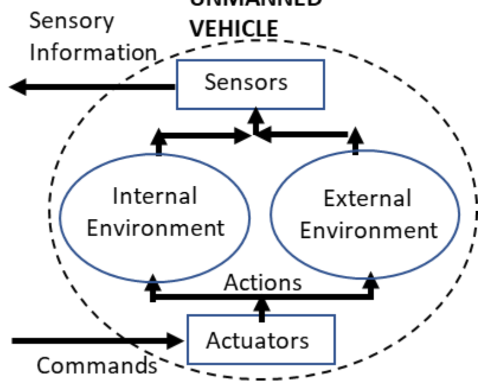
Autonomy Stack (AS)



LINK TO simulations, experiments, operations
for data generation, ML, AI

OR NET CPS
OR NET CHPS
SYSTEM

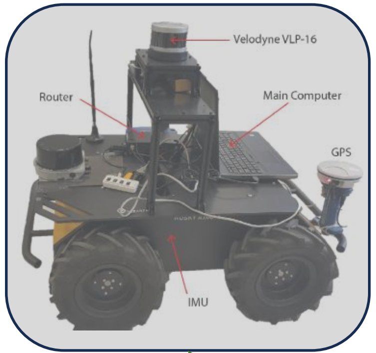
PHYSICAL UNMANNED VEHICLE

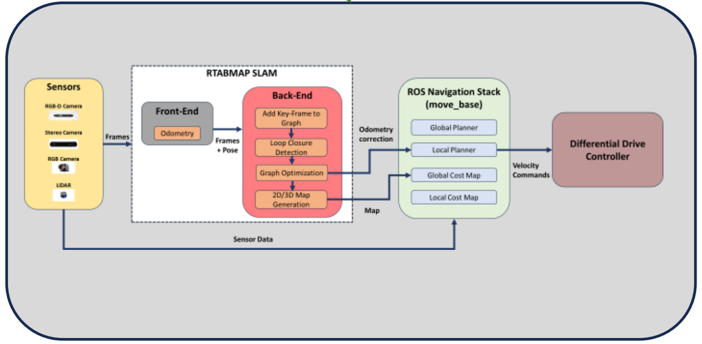


Design Space Exploration for Robotics

- Robotic Autonomous Systems are complex.
- Design involves structural and behavioral components .
- Involve a combination of model-based and data-driven techniques.
- System Development is often distributed and ad-hoc, without product lifecycle management.
- *“What-if?”* questions can arise in 3 cases
- Core tenets of Systems Engineering:

Case 1: Structural Change without Behavioral Change
Case 2: Behavioral Change without Structural Change
Case 3: Coupled Structural and Behavioral Changes





Architecture
- Structural and Behavioral Modeling
- Requirements engineering
- MBSE performed using SysML

Implementation
- Autonomy Stack for Navigation
- Perception, SLAM, Planning and Control
- Implemented using ROS/ROS2

Analysis
- Design Tradeoff and Sensitivity Analysis
- Monitoring System Performance
- Performed using Python, MATLAB, Julia.

UMD-SEIL Autonomy Stacks: ROS1 and ROS2

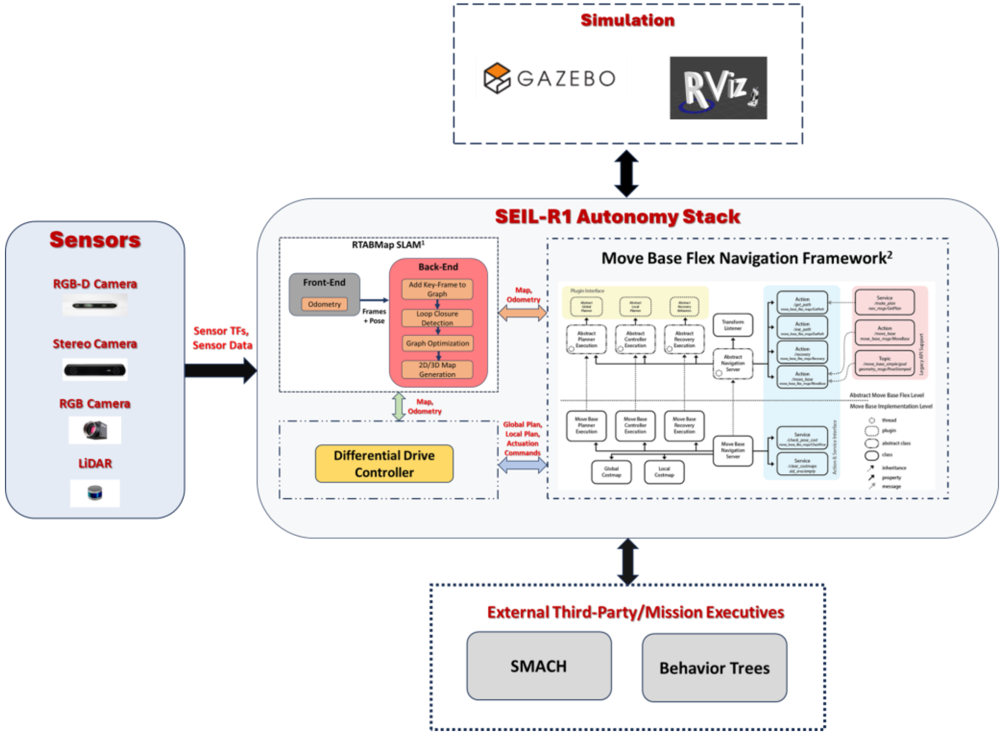


Fig: Architecture of the ROS-1 Navigation Stack

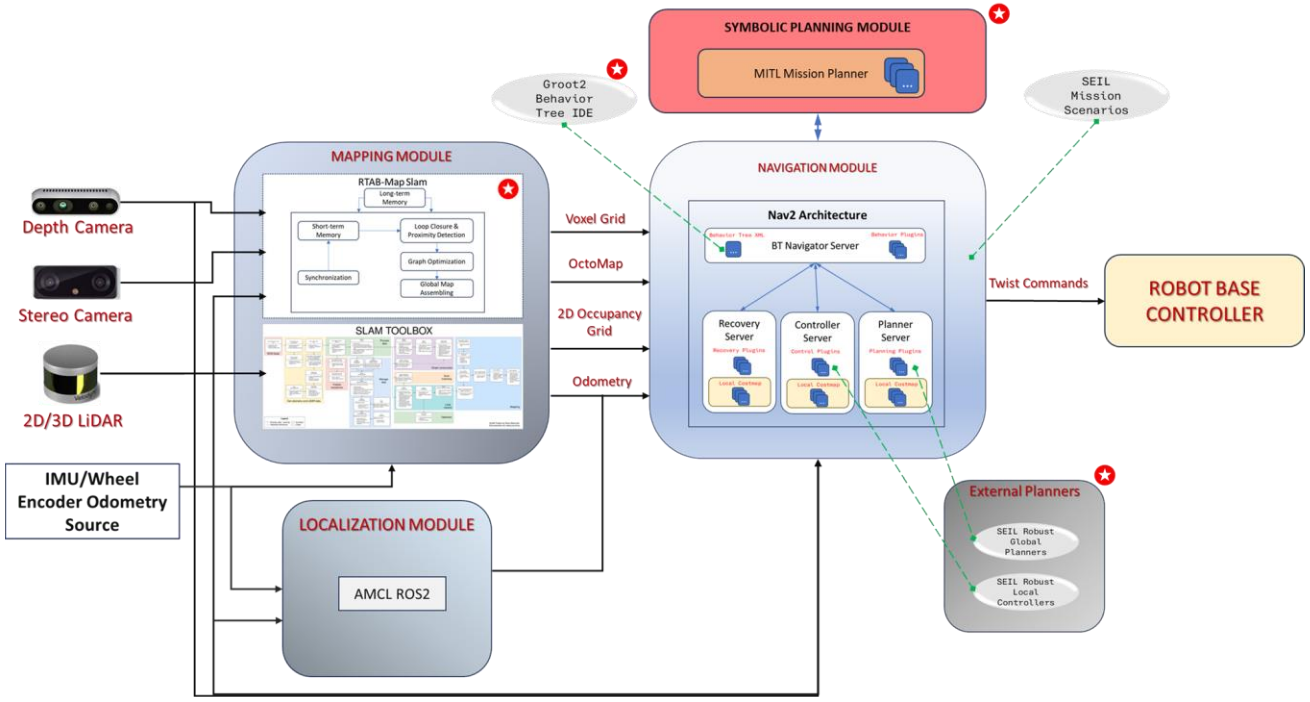


Fig: Architecture of the ROS-2 Navigation Stack

Simulation Frameworks:
- Isaac Sim
- Gazebo

Global Planning Algorithms:
- NavFn
- Smac Hybrid-A*
- Smac State Lattice

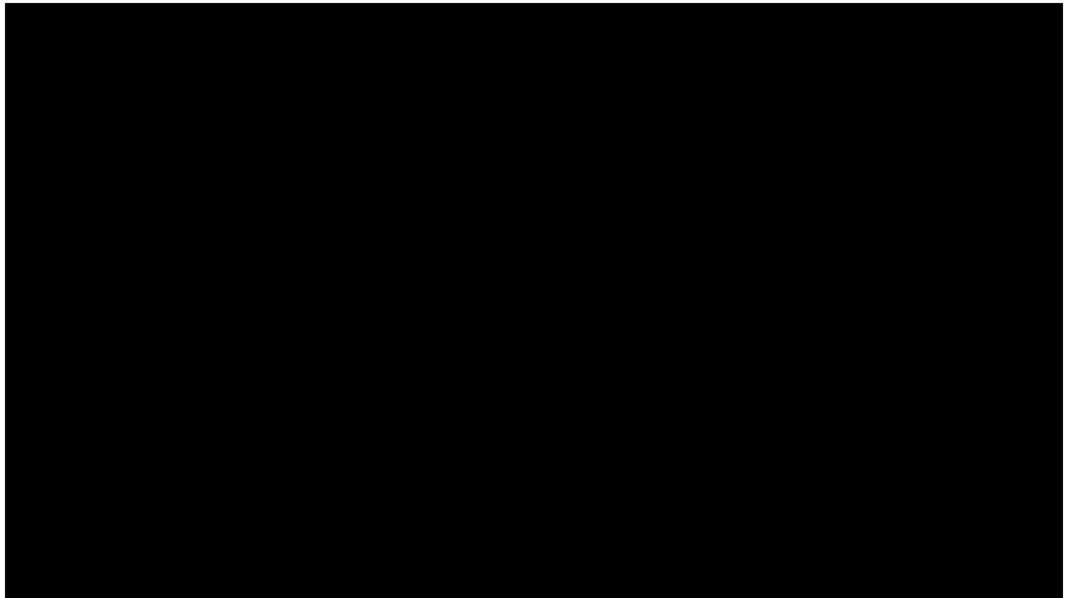
Local Controller Algorithms:
- Model Predictive Path Integral (MPPI)
- Timed Elastic Band (TEB)
- Dynamic Window Approach (DWA)

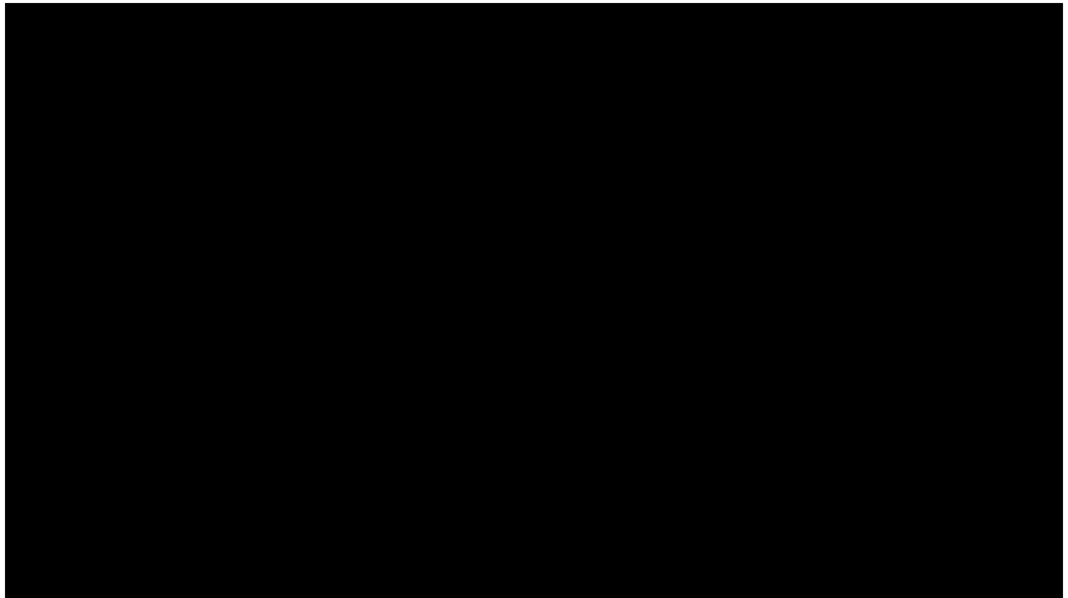
Standardized Test Suite for TRADES-X

- Some key features of the simulation world:
  - Very large map area with distinct offroad terrain zones.
  - Procedural domain randomization tools to improve Sim2Real gap.
  - Large collection of simulation assets, materials and textures.
  - Robust synthetic data generation capabilities built-in.
  - Decoupled execution of autonomy stack and simulation world.
  - Large collection of high-fidelity sensors available: ray-traced, vision-based and contact-based
  - Support for dynamic obstacles including models of people walking.
- High-fidelity collision mechanics and physics for a wide variety of assets to build external and internal world simulations.

High Fidelity Terrain Environment with Vegetation (Nvidia Isaac Sim)

Simulation Environments: Physics





Combining Model-Based & Data-Driven Trade Study

Basic approach to integrating model-based and data-driven optimization techniques.

Data-based metrics:
Can only be computed based on data collected from simulation runs.

Sensor Options

- 3D Lidar – 4 choices
- Laser Range Finder- 4 choices
- RGB Camera - 2 choices
- RGB Depth Cameras – 3 choices

2^13= 8192
Combinatorial
Sensor Suite Options

Model-Based Multi-Objective Optimization

Functional Optimization

$$\min_{x_1,\ldots,x_k \in \mathcal{S}} \bar{J}(\bar{x})$$

$$\text{s.t. } |C(\bar{x})| > c$$

where,

$$\bar{x} = [x_1, \ldots, x_k]^T$$

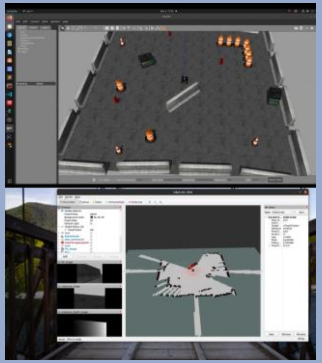
$$\bar{J}(\bar{x}) = [-\mathscr{E}(\bar{x}), \mathscr{P}(\bar{x}), \mathscr{R}(\bar{x}), \mathscr{C}(\bar{x})]^T$$

- *Effective Sensor Coverage* $\mathscr{E}(\bar{x})$
- *Sensor Suite Cost* $\mathscr{C}(\bar{x})$
- *Memory Usage* $\mathscr{R}(\bar{x})$
- *Power Consumption* $\mathscr{P}(\bar{x})$

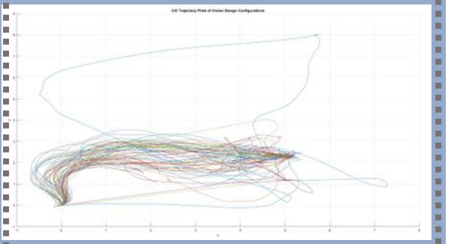
Pruning Design Configurations

Data-Driven Multi-Objective Optimization

ROS-Gazebo Simulation



Data Analysis



- *Time to Completion* $\mathcal{T}_c(\bar{x})$
- *Number of Human Interventions* $\mathcal{I}(\bar{x})$
- *Navigation Optimality* $\mathcal{N}(\bar{x})$

Statistical Studies of Configurations

Combining Model-Based & Data-Driven Trade Study

Basic approach to integrating model-based and data-driven optimization techniques.

Data-based metrics:
Can only be computed based on data collected from simulation runs.

Sensor Options

- 3D Lidar – 4 choices
- Laser Range Finder- 4 choices
- RGB Camera - 2 choices
- RGB Depth Cameras – 3 choices

2^13= 8192
Sensor Suite Options

Model-Based Multi-Objective Optimization

Functional Optimization

$$\min_{x_1,\ldots,x_k \in \mathcal{S}} \bar{J}(\bar{x})$$

$$\text{s.t.} \quad |C(\bar{x})| > c$$

where,

$$\bar{x} = [x_1, \ldots, x_k]^T$$

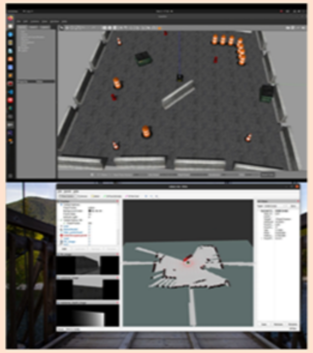
$$\bar{J}(\bar{x}) = [-\mathscr{E}(\bar{x}), \mathscr{P}(\bar{x}), \mathscr{R}(\bar{x}), \mathscr{C}(\bar{x})]^T$$

- *Effective Sensor Coverage* $\mathscr{E}(\bar{x})$
- *Sensor Suite Cost* $\mathscr{C}(\bar{x})$
- *Memory Usage* $\mathscr{R}(\bar{x})$
- *Power Consumption* $\mathscr{P}(\bar{x})$

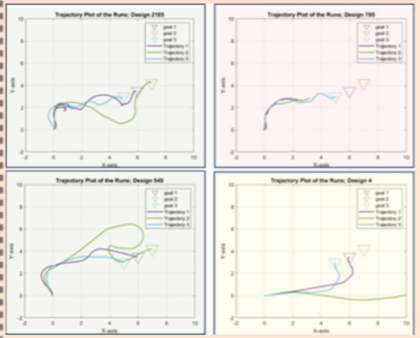
Pruning Design Configurations

Data-Driven Multi-Objective Optimization

ROS-Gazebo Simulation



Data-Driven Performance Evaluation



- *Time to Completion*
- *Number of Human Interventions*
- *Navigation Optimality*

Statistical Studies of Configurations

Model-Based Sensor Trade Study –Problem

- We consider a library of sensors of *4 types: Lidar, Laser, RGB Depth Camera, and RGB Camera*.
- We impose some limits on the number of sensors of each type that can be chosen. We call these the cardinality constraints $|C(\bar{x})|$
- We define a sensor configuration as: $\bar{x} = [x_1, \ldots, x_k]^T$

  where k potential sensors are available,

  and $\{x_i\}$ is a binary variable representing the selection of sensor i.
- The multi-objective optimization problem is formulated as:

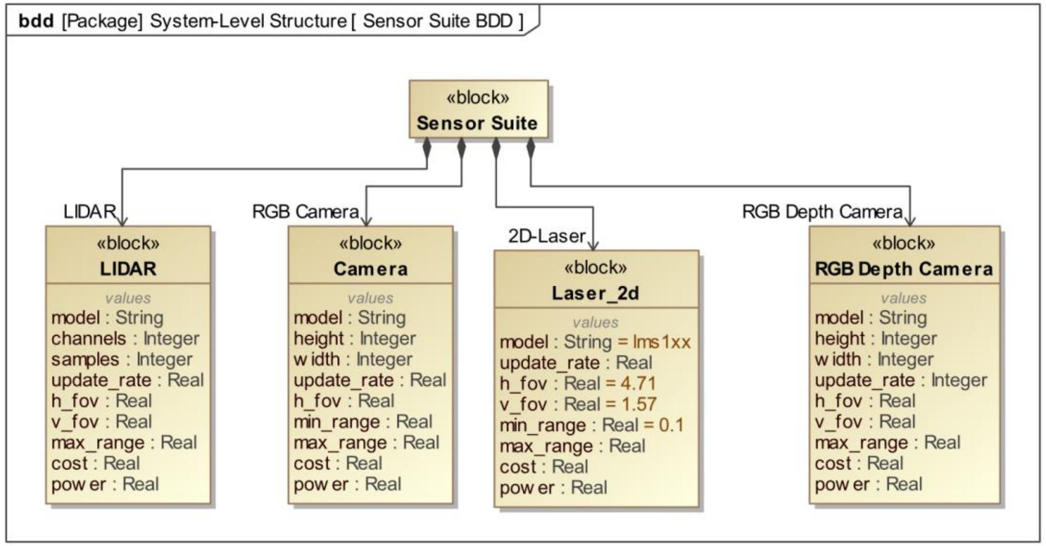
$$\min_{x_1,\ldots,x_k \in \mathcal{S}} \bar{J}(\bar{x})$$ Vector Objective Function

s.t. $|C(\bar{x})| < c$ Cardinality Constraint

where, $\bar{x} = [x_1, \ldots, x_k]^T$ Decision Variable

$$\bar{J}(\bar{x}) = [-\mathscr{E}(\bar{x}), \mathscr{P}(\bar{x}), \mathscr{R}(\bar{x}), \mathscr{C}(\bar{x})]^T$$

- The *combinatorial* problem as stated above is NP-hard.
- We exploit the nature of our objective functions and use heuristic algorithms to solve the problem.



(a)

#	Name	channels	samples	update_rate	h_fov	v_fov	max_range	cost	power
1	velodyne HDL 32E-A	32	2187	15	6.283	0.526	120	15000	100
2	velodyne HDL 32E-B	32	2187	20	6.283	0.526	120	16000	100
3	velodyne VLP-16-A	16	1875	15	6.283	0.526	100	10000	80
4	velodyne VLP-16-B	16	1875	20	6.283	0.526	100	12000	80

(b)

#	Name	update_rate	h_fov	v_fov	min_range	max_range	cost	power
1	lms111-a1	25	4.71	1.57	0.1	20	1000	30
2	lms111-b1	50	4.71	1.57	0.1	20	1500	30
3	lms151-a2	25	4.71	1.57	0.1	50	1500	50
4	lms151-b2	50	4.71	1.57	0.1	50	2000	50

(c)

#	Name	width	height	update_rate	h_fov	v_fov	max_range	cost	power
1	realSense d415	540	360	60	1.1345	0.6981	6	2000	8
2	realSense d435	1280	720	30	1.5184	1.0122	6	2500	8
3	realSense d455	1920	1080	15	1.501	0.9948	6	3000	10

(d)

#	Name	height	width	update_rate	h_fov	min_range	max_range	cost	power
1	backflyA	540	720	30	1.047	0.5	50	1500	3
2	backflyB	1080	1540	15	1.047	0.5	60	3000	6

Fig: (Top) SysMLL Block Definition Diagram of the Sensor Suite
(Bottom) Instance Tables for (a) Lidars (b) Lasers (c) RGB Depth Cameras (d) RGB Cameras

Model-Based Sensor Trade Study: Metrics

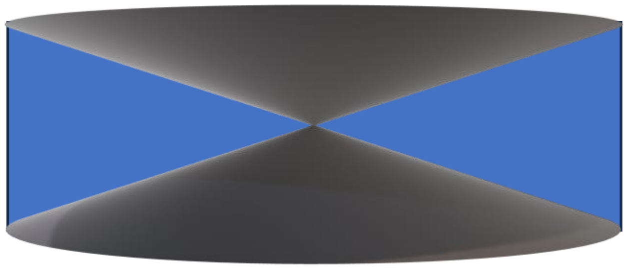
Effective Sensor Coverage $\mathscr{E}(\bar{x})$

- The Lidar is centrally mounted and has a 360-deg. horizontal Field of View (FoV). Camera, Depth Camera and Laser Range Finder are all pointed in the forward direction and have their own FoV limits.
- Each sensor type has its own *volumetric* coverage. We approximate the footprints as:
  - Lidar $\mathscr{E}(x_i) = \frac{4\pi}{3} r_{max}^3 . \cos^2\theta_{vfov} \sin\theta_{vfov}$
  - Camera and Depth Camera $\mathscr{E}(x_i) = \frac{\pi}{3} r_{max}^3 . \tan(\theta_{hfov}/2) \tan(\theta_{vfov}/2)$
  - 2D Laser $\mathscr{E}(x_i) = \frac{\theta_{hfov}}{2} r_{max}^2 . h_{box}$
- The bounding boxes are applied on top of these to account for ground and ceiling.
- The total effective sensor coverage of a given sensor configuration can be computed as:

$$\mathscr{E}(\bar{x}) = \bigcup_{i=1,\ldots,k} \mathscr{E}(x_i)$$

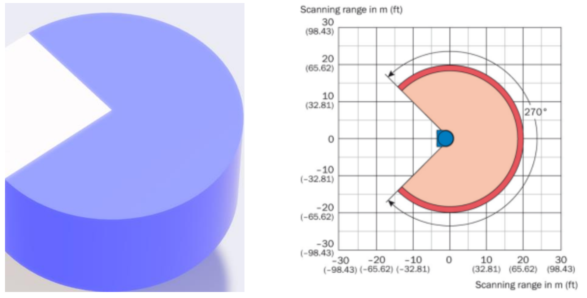
  - The metric is **sub-modular** and **monotonic**- A fact that will be leveraged for the solution.

Similarly, we have functional response models for RAM, power and cost metrics



(a) Volumetric Coverage (blue) of a 3d Lidar

(b) Volumetric Coverage (blue) of Cameras



(c) Volumetric Footprint of the Laser Scanner

Model-Based Sensor Trade Study: Method

Greedy Multi-Objective Sub-Modular Optimization (G-MOSMO)

- The design variable $\bar{x} = [x_1, x_2, \ldots, x_k]^T$ is a Boolean Vector with each element representing the selection of a sensor in the given configuration. We consider 4 Lidars, 4 Lasers, 3 RGB-Depth Cameras and 2 Cameras. Therefore, we have k=13.
- We impose a sensor configuration cardinality of c= 6. Therefore, the total no. of sensors in any configuration cannot be more than 6.
- Therefore, we have 8192 (the power set of 13 sensors) potential configurations to equip the robot. Evaluating all configurations is hard.

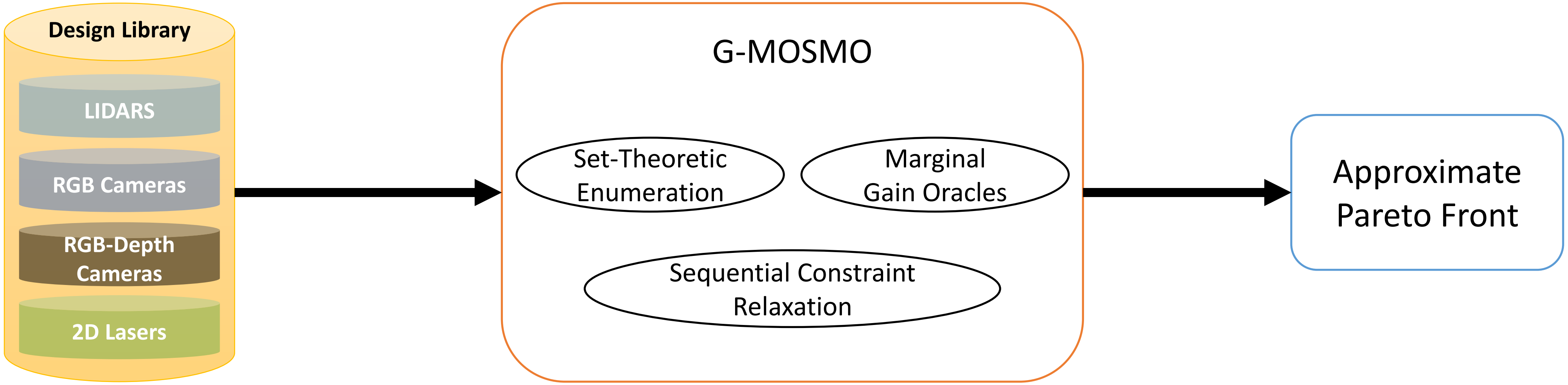


Fig: Schematic of the Greedy Multi-Objective Sub-Modular Optimization (G-MOSMO) Routine. Adapted from [1].

[1] Collin, Anne, et al. "A multiobjective systems architecture model for sensor selection in autonomous vehicle navigation." Complex Systems Design & Management: Proceedings of the Tenth International Conference on Complex Systems Design & Management, CSD&M Paris 2019. Springer International Publishing, 2020.

Model-Based Sensor Trade Study: Results

- We show that the MBO routine allows us to efficiently construct the approximate pareto front for the model-based metrics for the sensor selection problem.
- In the figures below, we show how we can use G-MOSMO to select 7 design candidates out of the 96 pareto optimal configurations for the given set of model-based metrics. **These candidates will now be evaluated using simulation runs.**

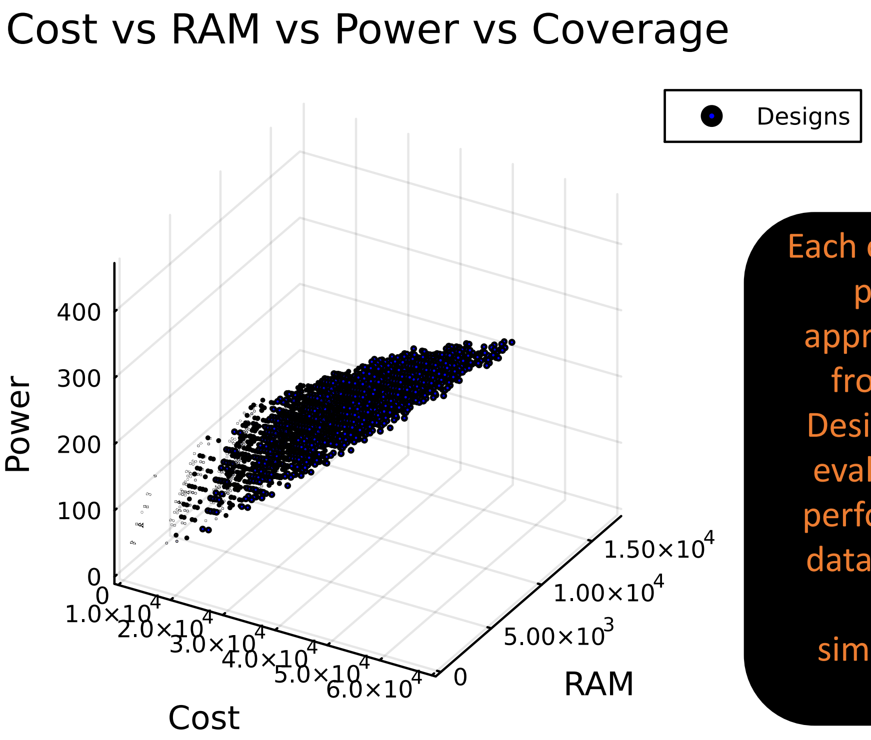


Fig: Evaluations of all feasible designs (# Designs = 4095).
We want to minimize RAM, Cost and Power while maximizing Sensor Coverage
(represented here by the marker size)

Each of the 7 (orange) points on the approximate Pareto front represents Designs that will be evaluated for their performance on the data-driven metrics by running simulations of the stack.

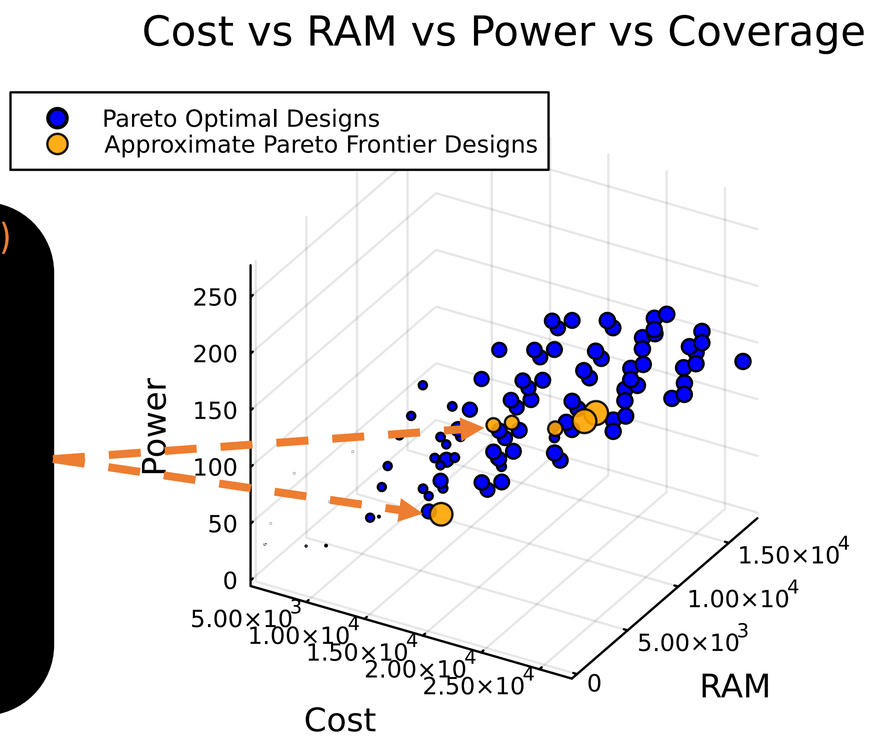


Fig: Evaluation of the Pareto Frontier (**# Pareto Optimal Designs = 96**).
We want to minimize RAM, Cost and Power while maximizing Sensor Coverage
Candidate solutions of G-MOSMO (**# Approximate Pareto Front= 7**).

Model-Based Sensor Trade Study: Method

Architecture

Architecture
Modeling in SysML:

Structure, Behavior, Requirements and Trade Study Profiles

Implementation

Simulations of
Selected Designs
for Data Collection

Analysis

Model-based Pruning of Designs

Greedy MOdel-based Sub-Modular Optimization (G-MOSMO)

Finally, the data-driven metrics from the simulation runs are evaluated and combined with the model-based metrics to perform the MAVF analysis and provide design recommendations.

The selected designs are evaluated through data collected from multiple simulation runs for on a standardized set of evaluation tasks.

Data-Driven Analysis–Results

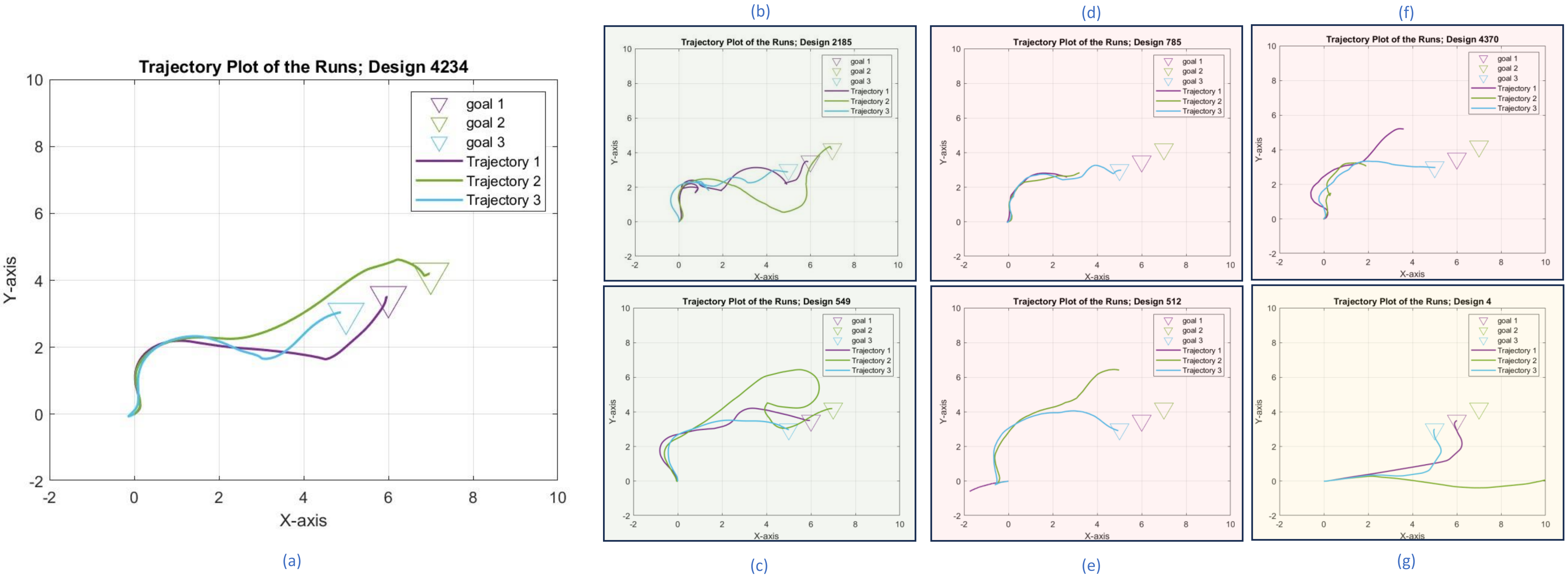


Fig: Trajectory Plots of the 7 different (pareto optimal, MBO-recommended) sensor suite design configurations across 3 Test Goals in a Gazebo Test Environment

(a), (b), (c)- Mission Success 100% [N_{hi}=0] | (d), (e), (f)- Mission Success 33% (N_{hi}=2) | (g)- Mission Success 50% (N_{hi}=1)

Encoding Scheme used for Sensor Suite Design IDs: Bool(Vector) $\bar{x} = [x_1, \ldots, x_k]^T$ ←→ Decimal

Ex: Design 4234→ [1 0 0 0 0 1 0 0 0 1 0 1 0] → { `VLP-16-A (lidar) + lms111-b1 (laser) + realSense d455 (rgb depth cam) + blackflyA (rgb cam)`}

Multi-Attribute Value Function Analysis

- The Table below describes the evaluation of the optimal design using the Multi-Attribute Value Function Analysis.
- For runs that resulted in failure to reach goal were assigned large penalties for Tc and Pl metrics were assigned.
- The relative weighting used for the metrics have also been shown. The design recommendation depends on these weight assignments.

			Human Interventions (# out of n=3)	Mean (n=3) Path Length (m)	Mean (n=3) Completion Time (s)	Effective Coverage (m^3)	Cost ($)	Memory Usage (MB)	Power Consumption (W)	Multi-Attribute Value Function
S.no	Design ID	Design	I w=0.24	P_l w=0.15	T_c w=0.15	E w=0.06	C w=0.24	R w=0.09	P w=0.06	MAVF
1	4234	{VLP-16-A, lms111-b1, realSense 455, blackflyA }	0.00	8.75	15.00	541802.00	15500.00	7698.00	121.00	**0.70**
2	2185	{VLP-16-B, lms111-b1, realSense 415, blackflyA }	0.00	12.15	44.00	562462.00	19000.00	7967.40	124.00	**0.49**
3	549	{HDL-32E-B, lms151-b2, realSense 455, blackflyB }	0.00	12.34	22.67	936683.00	24000.00	6233.00	166.00	**0.53**
4	785	{HDL-32E-B, lms111-a1, realSense 415, blackflyB }	2.00	16.02	55.33	934174.00	22000.00	7385.86	144.00	**0.12**
5	512	{HDL-32E-B }	2.00	16.60	57.33	884326.00	16000.00	140.00	100.00	**0.27**
6	4370	{VLP-16-A, lms111-a1, realSense 415, blackflyA }	2.00	15.90	54.33	541763.00	14500.00	7036.65	121.00	**0.20**
7	4	{realSense 455}	1.00	11.93	35.33	148.75	3000.00	2332.00	10.00	**0.66**

Improving Design Space Exploration

- Sensitivity Analysis via Automatic Differentiation (AD)
  - Our MBO module was designed to be amenable to integration with powerful AD tools.
  - We seek to use these tools to understand how sensitive the metrics are with respect to infinitesimal variations in the input (design) parameters.
  - Can be used to synthesize a notion of hierarchy in the system requirements– ***ranking of requirements according to their sensitivities.***
- Uncertainty Quantification via Variational Inference
  - The DDE module cannot be used to study sensitivities using gradients.
  - Instead, we propose the use of variational inference to quantify the (probabilistic) uncertainty in the data-driven metrics caused by the input parameters being chosen from a known distribution.
  - The input distribution captures the known uncertainty in the input space.

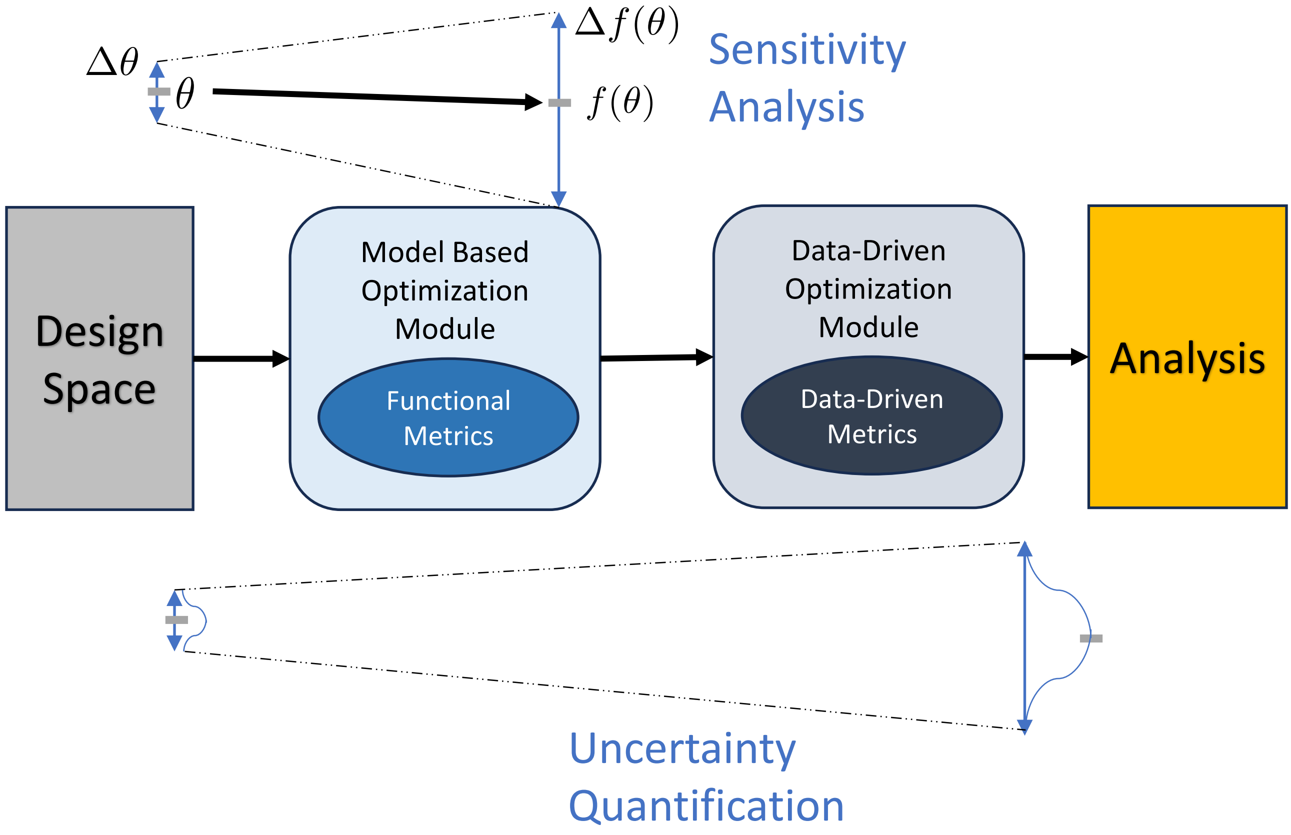


Fig: (top) The proposed framework for an improved TRADES-X tool. (bottom) Schematic of the autonomy pipeline from an Automatic Differentiation Perspective

Local Sensitivity Analysis

- Local Sensitivity Analysis using gradient information can be used to determine the uncertainty for each design instance.
- The Sensor Suite Selection problem, as formulated, has the following model-based metrics:
  - ***Effective Sensor Coverage*** $\mathscr{E}(\bar{x})$
  - ***Sensor Suite Cost*** $\mathscr{C}(\bar{x})$
  - ***Memory Usage*** $\mathscr{R}(\bar{x})$
  - ***Power Consumption*** $\mathscr{P}(\bar{x})$
- The Cost metric is just the sum of the cost parameters of the sensors. The cost parameters also do not impact any other metric.
- The Memory and Power Consumption metrics are also simple, and the gradients can be easily computed.

#	Name	Text	Satisfied By	Margin
1	SSR Functional Requirements	All sub-systems shall meet all the minimal specifications of the stakeholder.		
2	SSR.1 Sensor Suite Functional Requirements	The Sensor Suite sub-system shall meet all the minimal specifications of the stakeholder.		
3	SSR.1.1 Lidar Functional Requirements			
4	SSR.1.1.1 Lidar Channels FR	The lidar sensor shall have no less than 16 channels.	channels : Integer = 16	0
5	SSR.1.1.2 Lidar samples FR	The lidar sensor shall capture more than 1800 laser rays in a single sweep.	samples : Integer = 1875	75
6	SSR.1.1.3 Lidar sampling rate FR	The lidar sensor shall have a sampling rate of greater than 10 Hz.	update_rate : Real = 20.0	10
7	SSR.1.1.4 Lidar Range FR	The lidar sensor shall have a max range of greater than 90 m.	max_range : Real = 100.0	10
8	SSR.1.1.5 Lidar cost FR	The lidar sensor shall cost less than 18000 USD.	cost : Real = 12000.0	6000
9	SSR.1.1.6 Lidar Power Draw FR	The lidar sensor shall have an operational power draw of less than 120 W.	power : Real = 80.0	40
10	SSR.1.1.7 Lidar Vertical FoV FR	The lidar sensor shall have a vertical Field of View greater than 0.5 radians	v_fov : Real = 0.526	0.026
11	SSR.1.2 RGB Camera Functional Requiremer			
12	SSR.1.2.1 RGB horizontal resolution FR	The RGB camera sensor shall have a horizontal resolution greater than 540 pixels.	width : Integer = 1540	1000
13	SSR.1.2.2 RGB Vertical resolution FR	The RGB camera sensor shall have a vertical resolution greater than or equal to 1080 pixels.	height : Integer = 1080	0
14	SSR.1.2.3 RGB sampling rate FR	The RGB camera sensor shall have a sampling rate of greater than 10 Hz.	update_rate : Real = 15.0	5
15	SSR.1.2.4 RGB min_range FR	The RGB Camera sensor shall have a min range of less than 1 m.	min_range : Real = 0.5	0.5
16	SSR.1.2.5 RGB max_range FR	The RGB Camera sensor shall have a max range of greater than 40 m.	max_range : Real = 60.0	20
17	SSR.1.2.6 RGB Cost FR	The RGB camera sensor shall cost less than 4000 USD.	cost : Real = 3000.0	1000
18	SSR.1.2.7 RGB Power FR	The RGB camera sensor shall have an operational power draw of less than 8 W.	power : Real = 6.0	2
19	SSR.1.3 RGB Depth Camera Functional Requ			
20	SSR.1.3.1 RGBD horizontal resolution FR	The RGB Depth camera sensor shall have a horizontal resolution greater than or equal to 540 pixels.	width : Integer = 540	0
21	SSR.1.3.2 RGBD vertical resolution FR	The RGB Depth camera sensor shall have a vertical resolution greater than or equal to 360 pixels.	height : Integer = 360	0
22	SSR.1.3.3 RGBD horizontal FoV FR	The RGB Depth camera sensor shall have a horizontal FoV greater than 1.13 radians.	h_fov : Real = 1.1345	0.0045
23	SSR.1.3.4 RGBD vertical FoV FR	The RGB Depth camera sensor shall have a verical FoV greater than 0.69 radians.	v_fov : Real = 0.6981	0.0081
24	SSR.1.3.5 RGBD sampling rate FR	The RGB Depth camera sensor shall have a sampling rate of greater than 10 Hz.	update_rate : Integer = 60	50
25	SSR.1.3.6 RGBD Power FR	The RGB Depth camera sensor shall have an operational power draw of less than 12 W.	power : Real = 8.0	4
26	SSR.1.3.7 RGBD Cost FR	The RGB Depth camera sensor shall cost less than 4000 USD.	cost : Real = 2000.0	2000
27	SSR.1.4 2D Laser Functional Requirements			
28	SSR.1.4.1 laser sampling rate FR	The laser sensor shall have a sampling rate of greater than 20 Hz.	update_rate : Real = 50.0	30
29	SSR.1.4.2 Laser max_range FR	The laser sensor shall have a max range of greater than 15 m.	max_range : Real = 50.0	35
30	SSR.1.4.3 Laser cost FR	The 2D laser sensor shall cost less than 2500 USD.	cost : Real = 2000.0	500
31	SSR.1.4.4 Laser Power FR	The 2D laser sensor shall have an operational power draw of less than 60 W.	power : Real = 50.0	10

Fig: List of lowest-level component requirements [specifications].

SysML Requirement Table shows that components of the specific design instance meet all low-level specifications.

`{Vel-16A Lidar, Blackfly-B Camera, RealSense435i Depth Camera, lms1xx-a Laser}`

Local Sensitivity Analysis

- Computing the gradient of the Effective Sensor Coverage Metric requires Automatic Differentiation.
- We use **Forward-Mode AD** to calculate the gradient of the metric with respect to the (20) aggregated input parameters of a design instance.
- For our analysis, the final design recommendation was:

Design 4234 `{VLP-16-A, lms111-b1, realSense 455, blackflyA}`

$$\mathscr{C}(\bar{x}) = f_{\mathscr{C}}(c^L, c^C, c^D, c^{La})$$

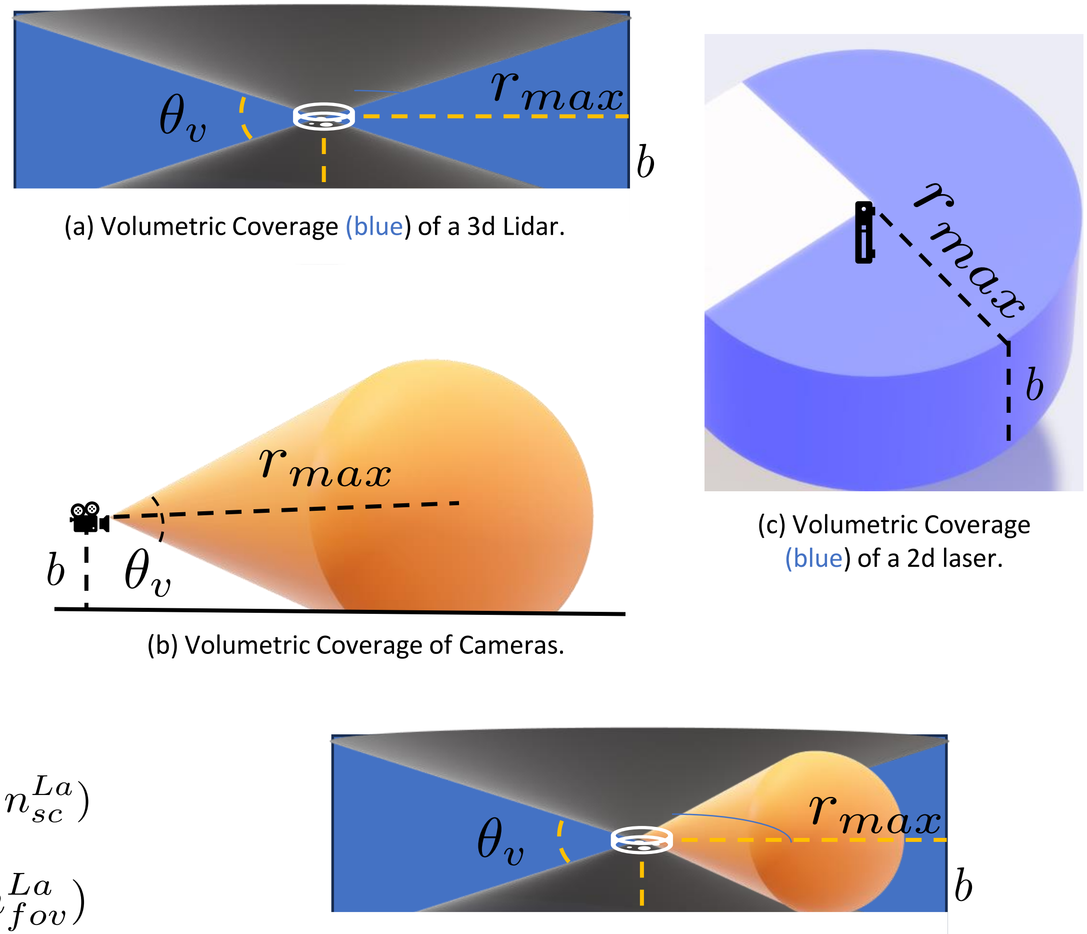
$$\mathscr{R}(\bar{x}) = f_{\mathscr{R}}(n_s^L, n_{ch}^L, n_{sc}^L, n_s^C, p_h^C, p_w^C, n_s^D, p_h^D, p_w^D, n_s^{La}, n_{sc}^{La})$$

$$\mathscr{P}(\bar{x}) = f_{\mathscr{P}}(p^L, n_s^L, n_{ch}^L, n_{sc}^L, p^L, n_s^C, p_h^C, p_w^C, p^D, n_s^D, p_h^D, p_w^D, p^{La}, n_s^{La}, n_{sc}^{La})$$

$$\mathscr{E}(\bar{x}) = f_{\mathscr{E}}(r_{max}^L, v_{fov}^L, r_{max}^C, h_{fov}^C, v_{fov}^C, r_{max}^D, h_{fov}^D, v_{fov}^D, r_{max}^{La}, h_{fov}^{La})$$

$$\nabla\mathscr{C}(\bar{x}), \nabla\mathscr{R}(\bar{x}), \nabla\mathscr{P}(\bar{x}), \nabla\mathscr{E}(\bar{x})$$

The gradients are computed using AD



(a) Volumetric Coverage (blue) of a 3d Lidar.

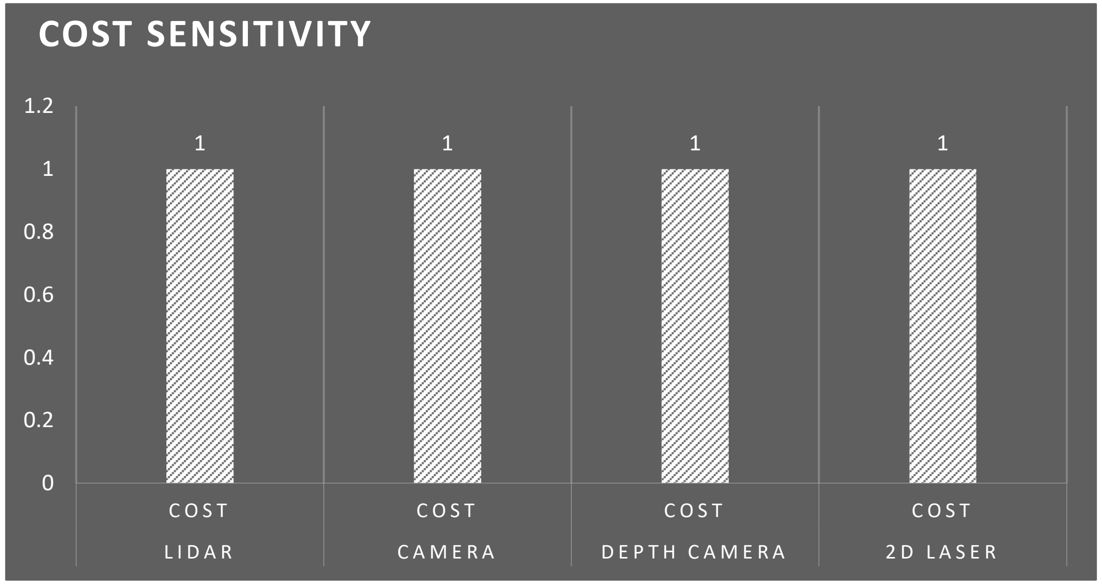
(b) Volumetric Coverage of Cameras.

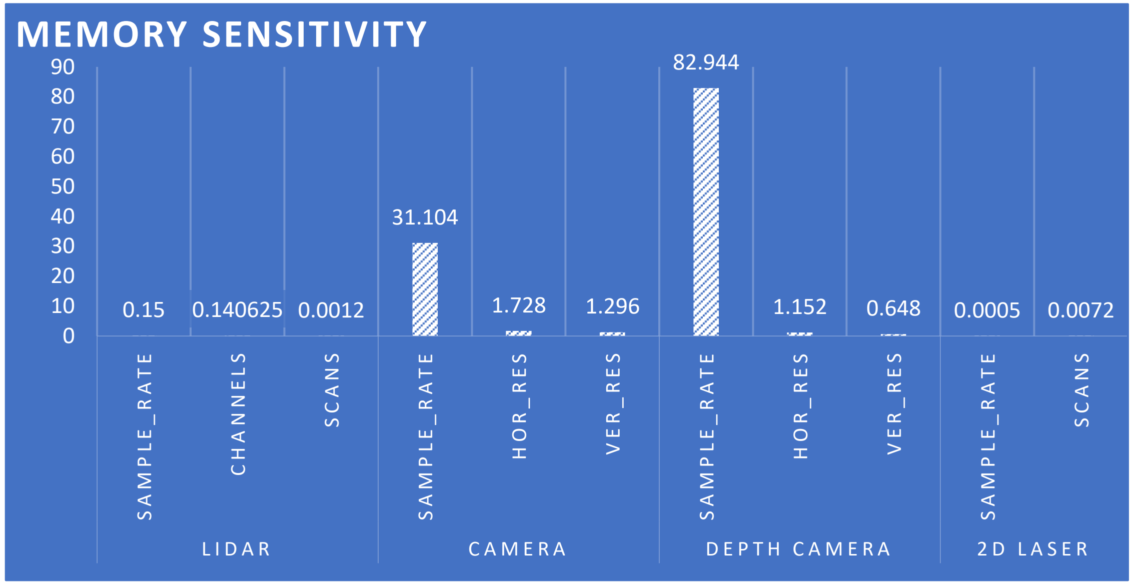
(c) Volumetric Coverage (blue) of a 2d laser.

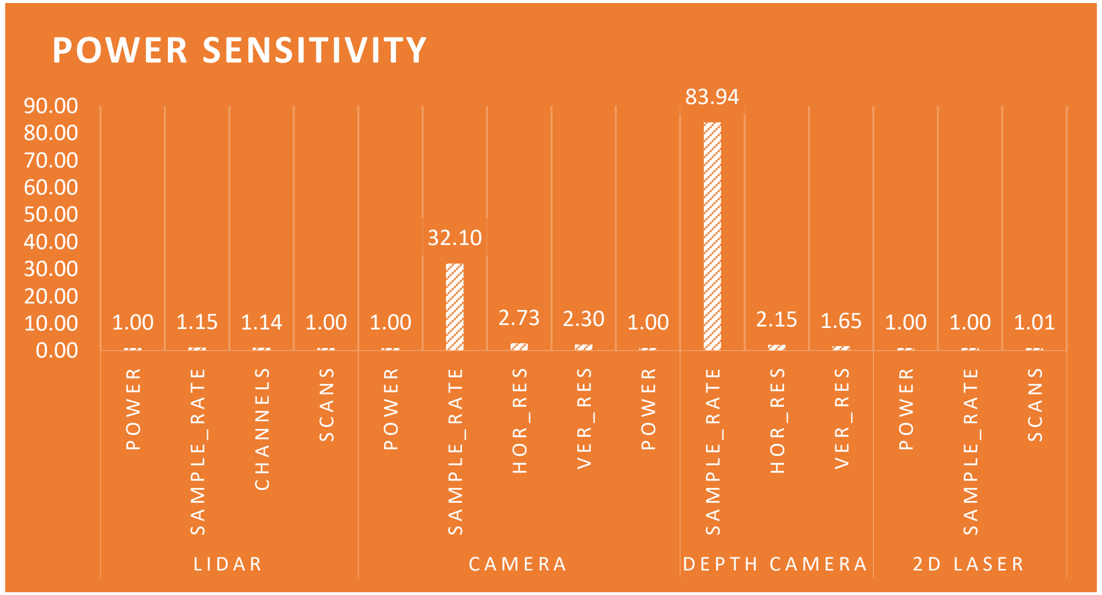
(d) Effective Cumulative Coverage

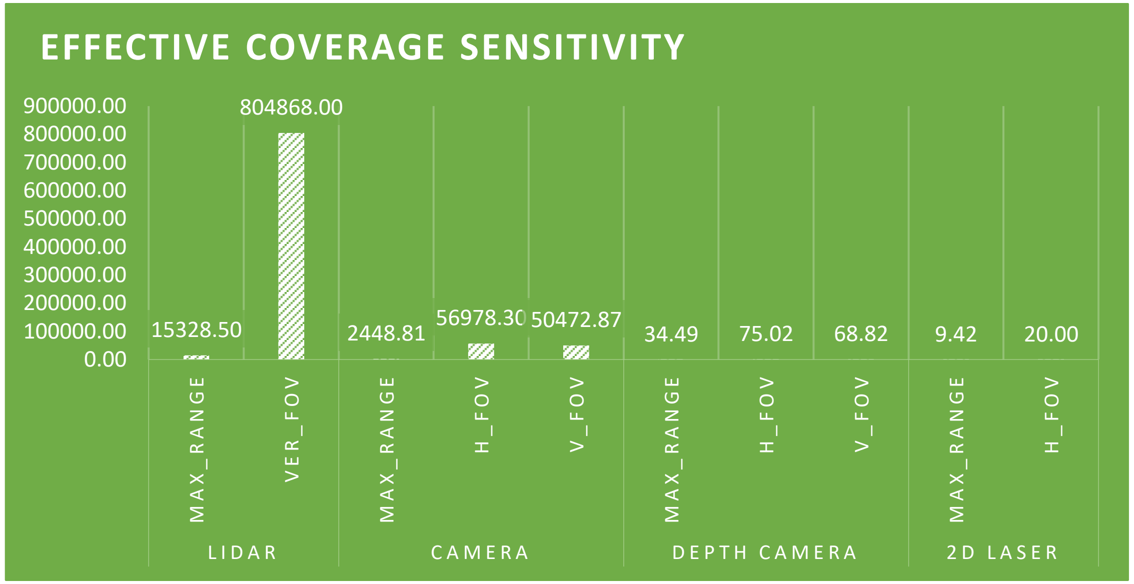
Fig: Effective sensor coverage computation is sub-modular and combinatorial. The specific closed-form expression varies for each design instance.

Local Sensitivity Analysis : Design 4234









Sensitivity Analysis for Ranking Requirements

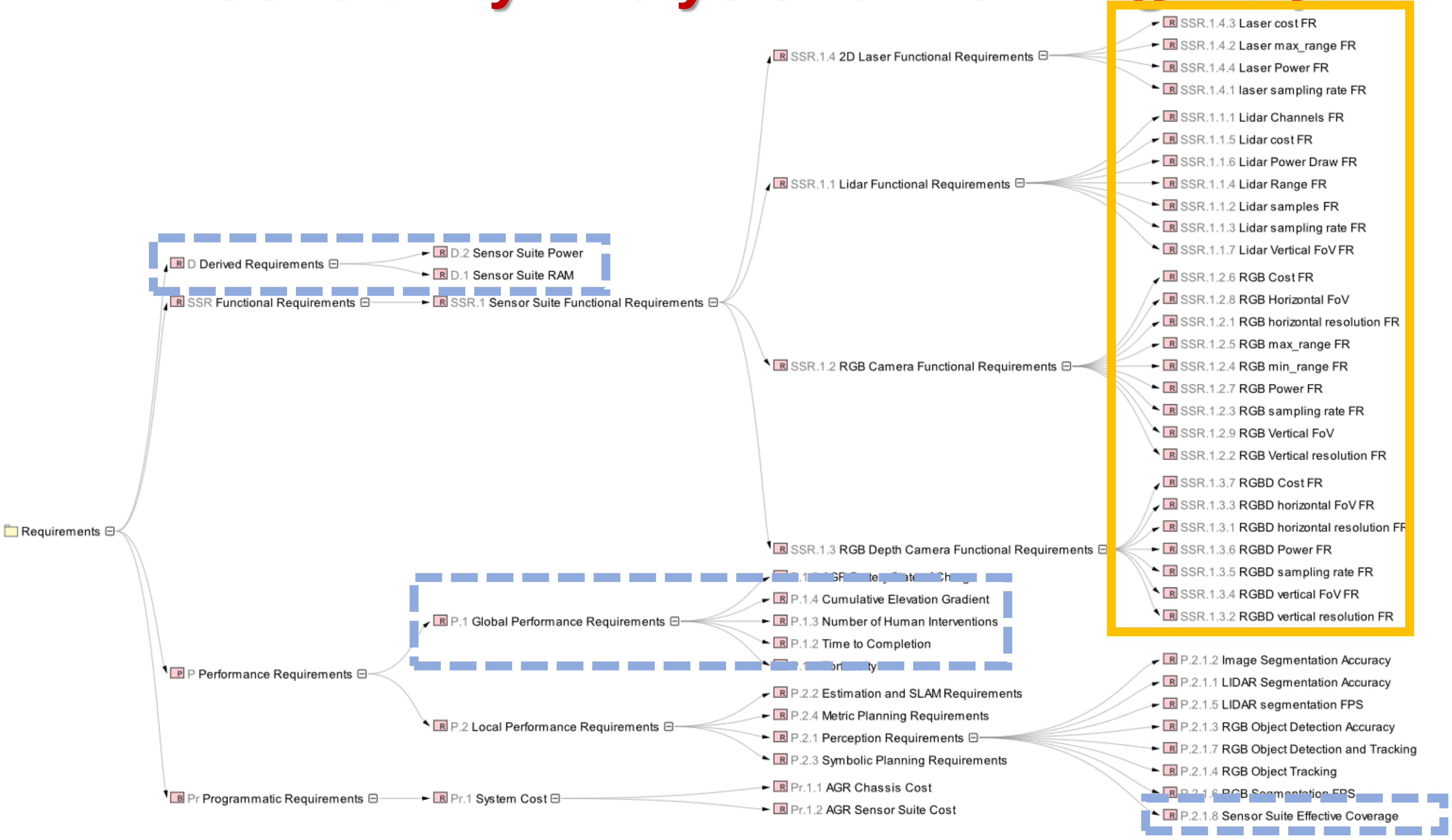


Fig: SysML Requirements Containment Map.
[Blue] High-Level Requirements Mapped to Global and Local Performance Metrics for DSE.
[Yellow] Low-Level Component Specification Requirements Allocated to Components

Coverage Metric Hierarchy:

1. SSR 1.1.7
2. SSR 1.2.8
3. SSR 1.2.9
4. SSR 1.1.4

Power and RAM Metric Hierarchy:

1. SSR 1.3.5
2. SSR 1.2.3

Cost Metric Hierarchy:

1. SSRs 1.1.5, 1.2.6, 1.3.7, 1.4.3

Thank you!

baras@umd.edu

301-405-6606

https://johnbaras.com/

Questions?