Simulation-Based Uncertainty Propagation in Geometric Networks for Surgical Robotics

X.M. Christine Zhu

Mentor: Dr. Russell H. Taylor

JOHNS HOPKINS UNIVERSITY

MOTIVATION

- Surgical robotic systems rely on **multiple geometric components**:
  - Robots, tracking systems, anatomy, sensors, etc.
- Each component introduces **uncertainty**:
  - Calibration error
  - Measurement noise
  - Kinematic modeling error
- These uncertainties **interact** through geometric composition

Core problem

There is no unified, general framework to **model and query uncertainty propagation across an arbitrary geometric network**.

This project bridges **geometry, probability, and simulation**, providing a principled way to understand uncertainty in complex robotic systems.

SYSTEM ABSTRACTION: NETWORK VIEW

System Representation

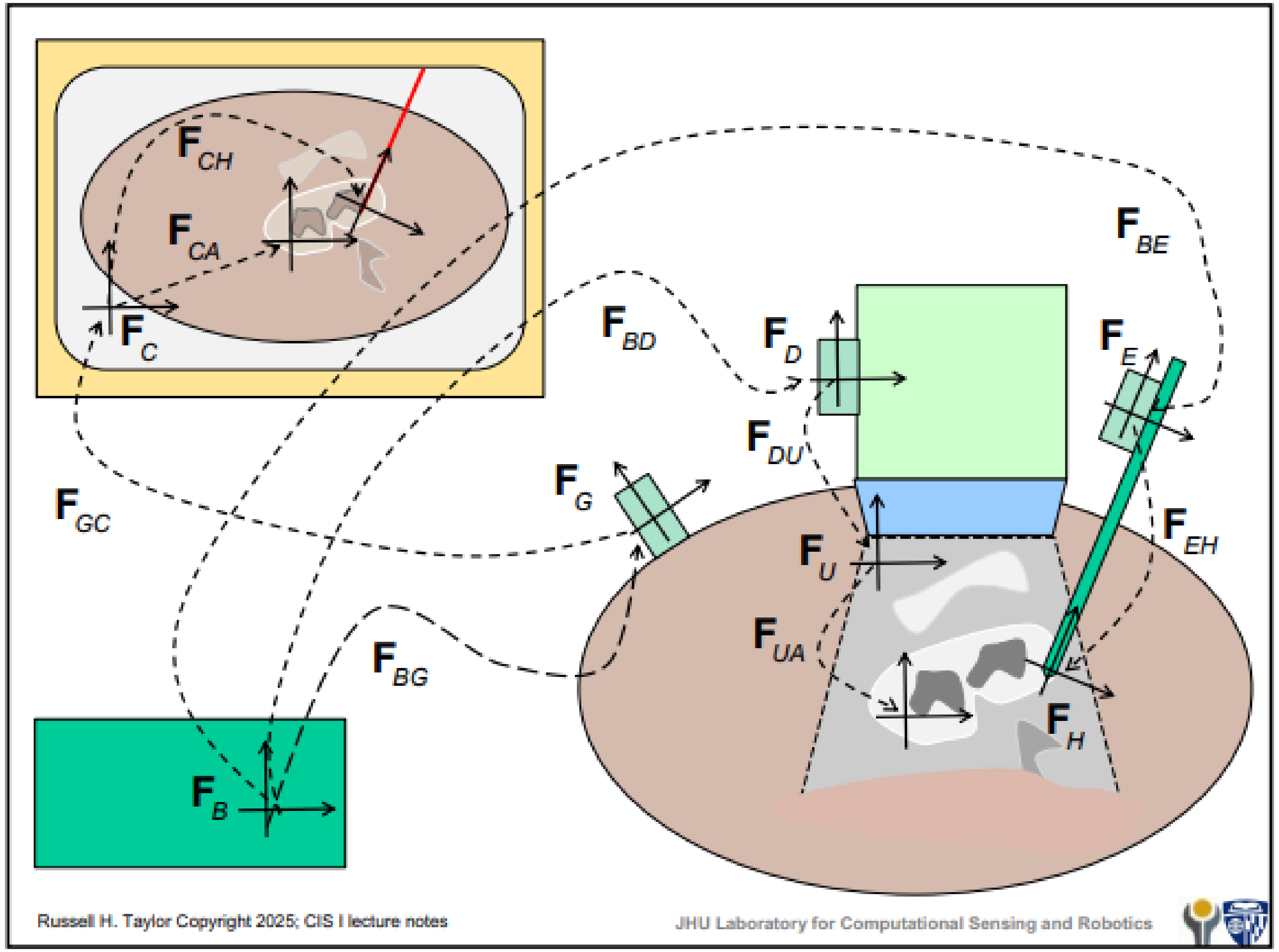
- System modeled as a **graph**:
  - **Nodes**: frames or points
  - **Edges**: geometric relationships with uncertainty

Examples of Nodes

- Robot base, links
- Optical tracker, EM tracker
- Ultrasound Sensors
- Video Cameras
- Anatomy
- Fiducial or tool to tip points, 3-DOF points

Examples of Edges

- Rigid transformations
- Kinematic chains
- Sensor measurements



Cited from 2025 CIS I lecture notes

PROJECT MOTIVATION & GOAL

Project Goal:

Develop a **general simulation framework** that models and propagates uncertainty through a network of geometric relationships, enabling uncertainty queries between any two nodes.

Significance:

- Enables **quantitative uncertainty reasoning** in:
  - Surgical navigation
  - Registration
  - Tracking–robot fusion
- Models systems as **networks of uncertain frames and points**
- Treats uncertainty as a **first-class geometric quantity**
- Enables **end-to-end uncertainty queries** between any two nodes
- Supports Monte Carlo validation
- Supports both 1-time uncertainty (e.g. tolerance) and multiple time uncertainty (e.g. noise)

TECHNICAL APPROACH & DEVELOPMENT PLAN

Technical approach summary

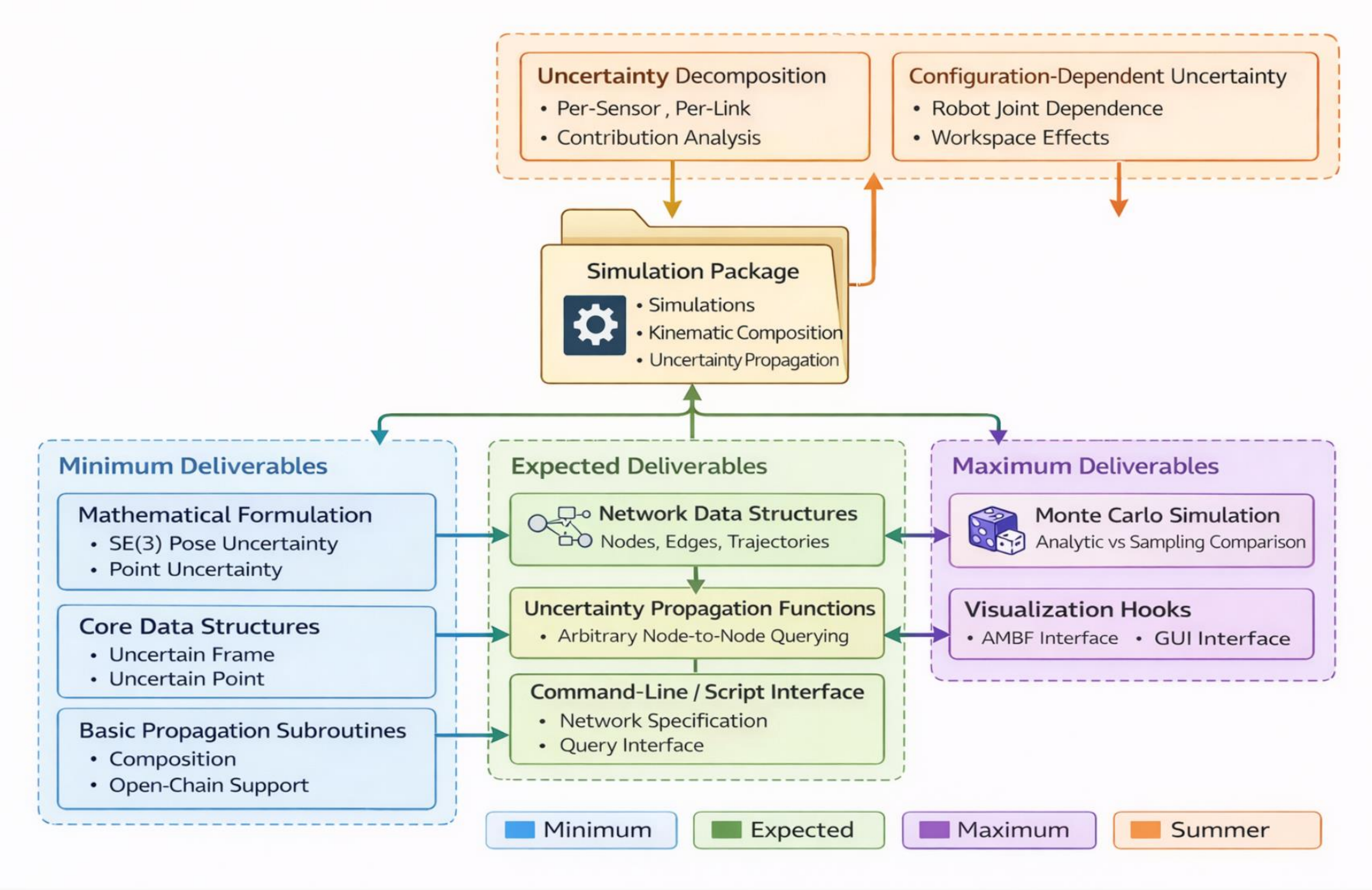
- Define uncertainty-aware geometric primitives
- Build composable operators
- Represent system as a graph
- Support Monte Carlo propagation
- Validate and visualize in AMBF

Development Plan

- Phase 1: Mathematical & Software Foundations
- Phase 2: Network & Simulation
- Phase 3: AMBF & Visualization

Figure: Illustration of Uncertainty Propagation

SYSTEM BLOCK DIAGRAM



DELIVERABLES

Minimum

- Mathematical formulation for uncertainty propagation ✅
- Core data structures ✅
  - Uncertain **Frame**
  - Uncertain **Point**
- Basic subroutines for composing and propagating uncertainty ✅
- Open-chain network support ✅
- Simple test cases demonstrating correctness ✅

Expected

- Fully functional **network system** for uncertainty propagation ✅
- Support for querying uncertainty between arbitrary nodes 🚩
- Command-line or script-based network specification 🚩
- Documentation and examples ✅

Maximum

- Monte Carlo simulation for validation and comparison 🚩
- AMBF-based visualization ❌
- GUI for interactive network specification and querying ❌

TIMELINE & MILESTONES

WEEKS	GOALS	MILESTONES
Weeks 1-2	• Establish correct uncertainty modeling. • Define covariance representations. • Validate with simple example.	Documents for math framework and implementation plan. Expected: 02/15/2026 ✅
Weeks 3-4	• Implement uncertainty-aware geometric primitives. • Working uncertainty propagation for kinematic chains.	Core function library validated on basic examples. Expected: 03/01/2026 ✅
Weeks 5-6	• Support general geometric networks. • Enable uncertainty queries between any two nodes.	General geometric network support implemented. Expected: 03/15/2026 🚩
Weeks 7-8	• Validate propagation. • Implement Monte Carlo sampling.	Monte Carlo validation completed. Expected: 03/29/2026 🚩
Weeks 9-10	• Usable interface. • Implement GUI.	User workflow validated on representative examples. Expected: 04/05/2026 ❌
Weeks 11-12	• Prepare evaluation-ready output. • Integrate to AMBF for visualization. • Finalize documentation.	Visualization integrated. Finalize documentation. Expected: 04/19/2026 ❌

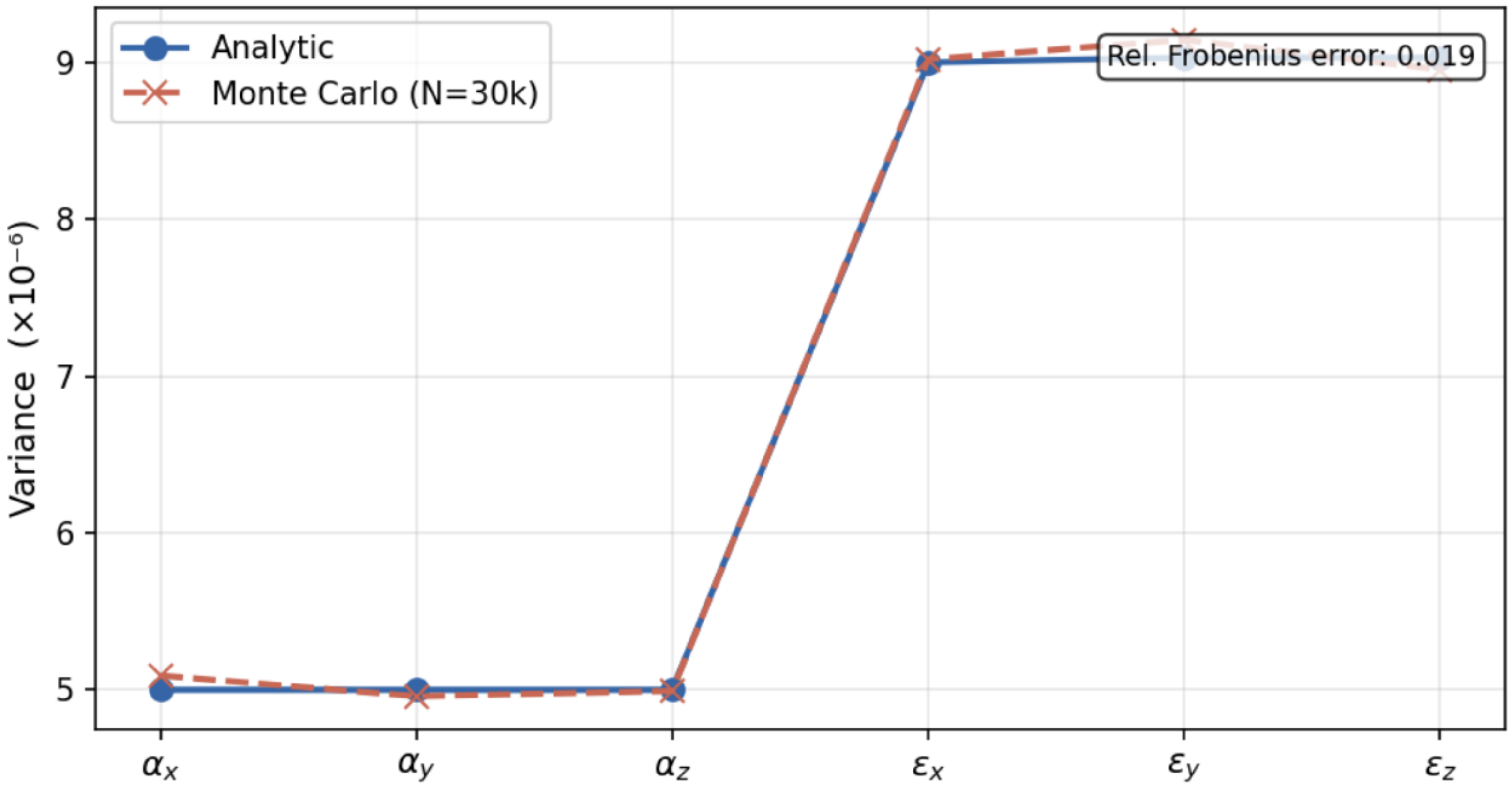
What Implemented So Far

- **SE(3) Lie group math:** exponential/logarithm maps, adjoint matrices, SO(3) left Jacobians with small-angle numerical stability
- **Uncertain transform primitive:** nominal pose + 6×6 covariance; composition, inversion, and point transformation via first-order linearization (CIS I convention)
- **Geometric network:** directed graph of frames connected by uncertain transforms; supports full SE(3), rotation-only, and translation-only edge types
- **Path queries:** BFS shortest path and DFS all-paths, each with full covariance propagation along the chain
- **Point uncertainty queries:** propagate 3D points across frames; correlation-aware point-to-point covariance that accounts for shared edges
- **Closed-loop Bayesian inference:** condition posterior covariance on loop constraints via information-form update; supports subspace (rotation/translation-only) conditioning
- ***Multi-loop conditioning(on going):*** simultaneously apply N loop constraints with stacked Jacobians and joint posterior
- ***Automatic loop discovery(on going):*** spanning tree cycle basis extracts all independent loops; auto-applies posterior conditioning
- ***Gaussian fusion(on going):*** information-form optimal fusion of multiple independent path estimates
- ***Monte Carlo validation(on going):*** 11 scripts validating all major features against MC sampling (open chains, closed loops, shared-edge correlations, surgical scene)

Preliminary Result

- Validating Uncertainty Propagation(Open-chain Analytic vs Monte Carlo)

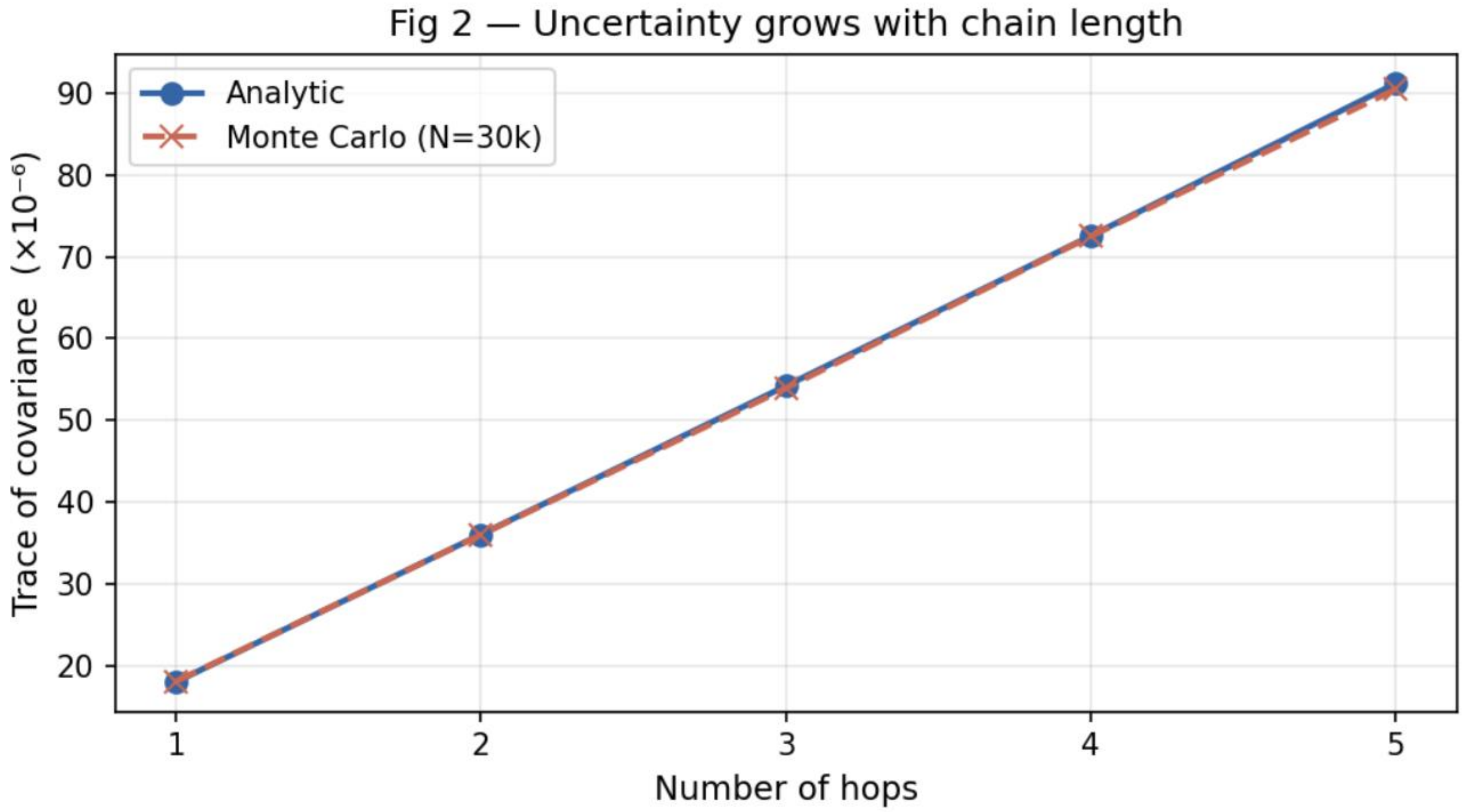
Fig 1 — Open-chain covariance: Analytic vs Monte Carlo



We verify our analytic covariance formula against Monte Carlo simulation (30,000 samples). Given two uncertain SE(3) transforms composed in sequence, the first-order propagation rule predicts the output covariance without sampling. The analytic result matches Monte Carlo with only **1.9% relative Frobenius error**, confirming the formula is correct.

Preliminary Result

- Uncertainty Accumulates with Chain Length(Uncertainty Growth Along a Chain)

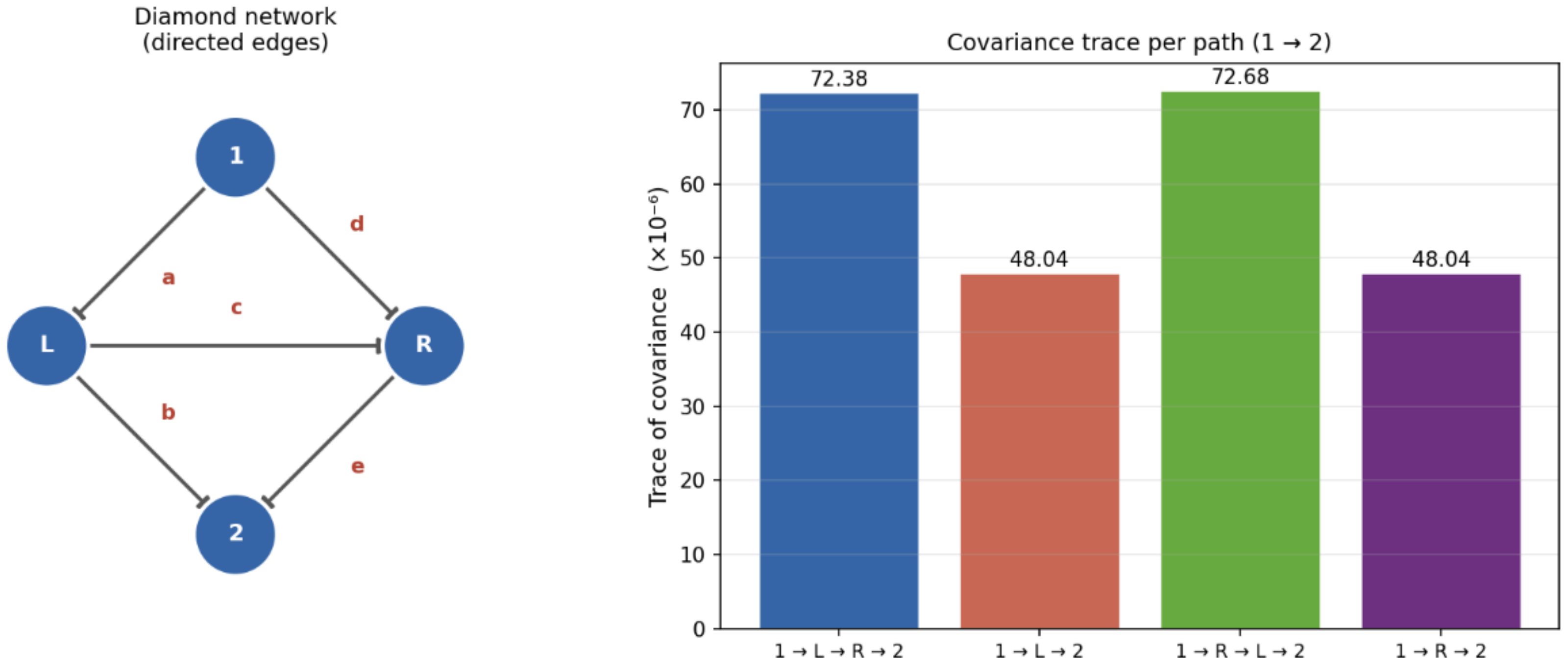


As a transform is propagated through more frames, uncertainty accumulates at each hop. Each edge contributes its own noise, and the total covariance grows with the number of steps. This motivates the need for a network-aware framework — different paths between the same two frames will accumulate different amounts of uncertainty depending on how many edges they traverse.

Preliminary Result

- Multiple Paths, Different Uncertainties(Diamond Network: All 4 Paths from Node 1 to Node 2)

Fig 3 — Diamond network: directed edges and propagated uncertainty



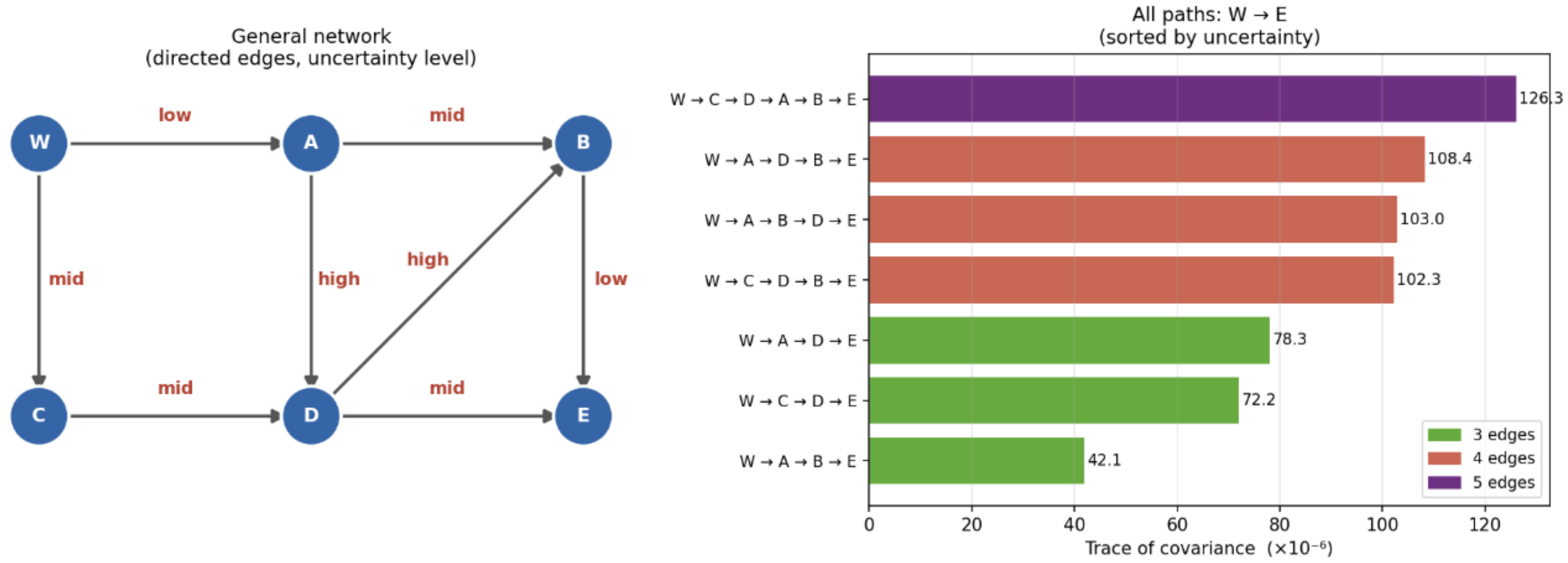
TODO:
Currently, the framework applies loop constraints one pair at a time. The next step is to support **multiple constraints simultaneously**

In the diamond network, there are **4 distinct paths** from node 1 to node 2. The framework automatically discovers all of them and propagates uncertainty along each one. Short 2-hop paths (1→L→2, 1→R→2) accumulate less uncertainty than longer 3-hop paths (1→L→R→2, 1→R→L→2). When a path traverses an edge backwards, the **inverse transform** is used automatically.

Preliminary Result

- Automatic Path Discovery in a General Network(Automatic Path Discovery in a General Network)

Fig 4 — Uncertainty propagation across all discovered paths



TODO:
make loop discovery **start/end aware** — given a specific start node and end node, automatically find all independent loop constraints between them

In a general 6-frame network with 8 edges, the framework discovers all 7 simple paths from W to E without any manual specification. Each path gets its own propagated covariance. The path W→A→B→E (3 edges, well-calibrated) has the lowest uncertainty at 42.1 $\times 10^{-6}$, while the longest path W→C→D→A→B→E (5 edges) reaches 126.3 $\times 10^{-6}$. This shows that the choice of path through the network directly affects the quality of the result.
Note: edge covariances here are synthetic — in a real system they would come from calibration data.

Preliminary Result

- Combining Path Discovery with Loop Constraints

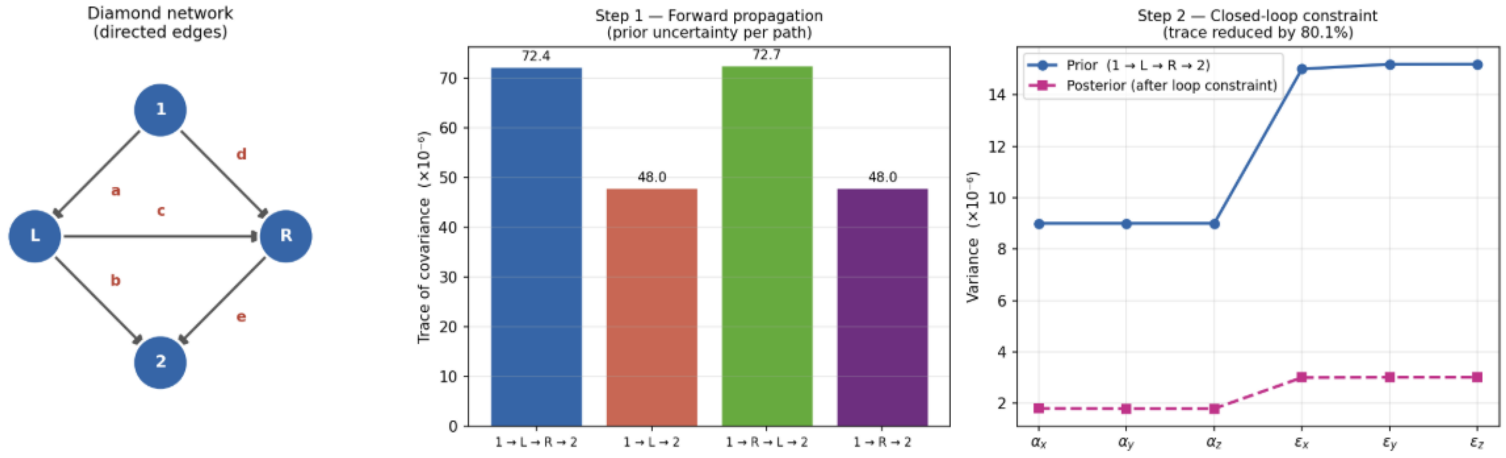


Fig 5 — Full pipeline: forward propagation → closed-loop constraint

TODO:
introduce a general **Observation interface** that can represent any type of measurement — loop closures, point observations, distance measurements, and more.

Fig2: The framework finds all 4 paths from node 1 to node 2 and computes the propagated uncertainty for each one independently. The prior covariance trace ranges from **48.0** to **72.7 $\times 10^{-6}$** depending on the path taken.
Fig3:In a consistent network, all 4 paths must give the same transform: a+b = d+e = a+c+e = d+inv(c)+b. The framework uses this equality as a constraint to condition the uncertainty of the reference path. After applying the loop constraint, the posterior covariance drops significantly across all 6 components, reducing the trace from **72.4** down to **14.4 $\times 10^{-6}$** — an **80.1% reduction**. This shows that the more alternative paths exist in the network, the more loop constraints can be applied, and the tighter the final uncertainty estimate becomes.

Next Steps & Milestones

- **Multiple Constraints (Batch Closed-Loop)**
  - Currently, the framework applies loop constraints one pair at a time. The next step is to support **multiple constraints simultaneously**
  - **Milestone:** Implement batch conditioning + validate with synthetic multi-loop graphs (04/03/2026)
- **Automatic Loop Discovery with Specified Start and End**
  - The framework can already discover loops across the entire network. The next step is to make loop discovery **start/end aware** — given a specific start node and end node, automatically find all independent loop constraints between them. This ensures the direction of every loop constraint is meaningful and intentional, which is critical for correct closed-loop conditioning.
  - **Milestone:** Develop constrained loop search (start/end aware) + independence filtering (04/10/2026)
- **Observation and Factor Abstraction**
  - Currently, loop constraints are the only type of constraint supported. The next step is to introduce a general **Observation interface** that can represent any type of measurement — loop closures, point observations, distance measurements, and more. This makes the framework extensible to real sensor data without changing the core network structure.
  - **Milestone:** Design Observation class + implement 2–3 example factors (loop, point, distance) (04/17/2026)
- **Visualization**
  - Plots the graph structure, highlights discovered paths, and displays the uncertainty on each edge and path.
  - **Milestone:** Build basic visualization (graph + covariance ellipses) + path highlighting (04/24/2026)

Next Steps & Milestones (Sim-to-Real, Summer 2026)

- **AMBF Integration**
  - Integration with AMBF for simulated surgical environment and geometry interaction.
  - **Milestone:** Connect framework to AMBF scene + validate geometry-driven uncertainty inputs (Before July)
- **GUI**
  - User interface for interacting with the network, selecting nodes, visualizing paths, and inspecting uncertainty.
  - **Milestone:** Prototype lightweight GUI (e.g., PyQt/web dashboard) for graph interaction (Before August)
- **Full System Integration & Validation**
  - Combine all modules (multi-constraint, observation model, AMBF, visualization) and validate on a surgical use-case scenario.
  - **Milestone:** End-to-end demo + evaluation (Before September)

MANAGEMENT

Meeting Schedule

- Regular standing meetings will be used to review progress, discuss modeling assumptions, and adjust scope as needed
  - Every Friday, 1:30-2:30 p.m. with Dr. Taylor
- Extra meetings may be scheduled if clarification or design decisions are required

Data Management

- All simulated data, configuration files, and experiment results will be stored in a structured project repository
- Intermediate results (e.g., Monte Carlo samples, uncertainty outputs) will be saved in reproducible formats for analysis and validation
- Version control will be used to track code and data evolution

Dependencies

- Access to computing – resolved
- Consultation from Dr. Munawar for AMBF – resolved

Background Reading

Uncertainty Propagation

https://en.wikipedia.org/wiki/Propagation_of_uncertainty

Uncertainty Modeling

https://en.wikipedia.org/wiki/Uncertainty_quantification

Kalman Filter

https://en.wikipedia.org/wiki/Kalman_filter

Measurement Theory

https://en.wikipedia.org/wiki/Measure_(mathematics)

Thank You!

X.M. Christine Zhu

xzhu83@jhu.edu