Surgical Robot Planning Through Governed Speculative Branches

by Nick Clark | Published March 27, 2026 | PDF

Surgical robots operate in environments where planning must be exhaustive and execution must be safe. Current surgical AI plans through optimization over pre-operative imaging, with limited ability to explore alternative approaches or recover from unexpected findings. The forecasting engine enables speculative planning through governed branches: multiple surgical paths explored in contained simulation, risk-evaluated against patient-specific constraints, with only validated plans promoted to the execution layer. The containment boundary ensures that speculative exploration never affects the patient.

The planning limitation in surgical robotics

Surgical robots like da Vinci and Mako operate from pre-operative plans derived from imaging data. The surgeon creates a plan, the robot executes it with precision. But intraoperative discoveries, unexpected anatomy, or tissue conditions that differ from imaging require real-time replanning. Current systems handle this by returning control to the human surgeon, because the robot has no structural mechanism for safe speculative replanning.

The gap is in planning under uncertainty. A surgical robot that encounters an unexpected vessel cannot safely explore alternative approaches without a mechanism that guarantees speculation does not affect execution. The robot needs to think through possibilities without moving, evaluate alternatives without committing, and only execute the path that passes all safety constraints.

Why optimization-based planning is insufficient

Current surgical planning optimizes a single path: the best trajectory to the target given pre-operative imaging. This produces an optimal plan for the expected scenario. It does not produce contingency plans for unexpected scenarios, because optimization finds one solution rather than exploring the solution space.

Monte Carlo planning generates multiple possible paths but provides no structural mechanism for evaluating which paths are safe to execute. The paths are statistical samples, not governed plans with safety evaluations attached. The surgeon must evaluate each alternative manually, which defeats the purpose of autonomous surgical planning.

How the forecasting engine addresses this

The forecasting engine creates speculative branches as first-class planning structures. Each branch represents a possible surgical path with its own risk assessment, tissue interaction predictions, and safety evaluations. Critically, all branches exist within a containment boundary that prevents any speculative computation from reaching the robot's actuators.

When the robot encounters an unexpected finding, the forecasting engine spawns multiple branches exploring alternative approaches. Each branch simulates the surgical path forward: instrument trajectories, tissue interactions, proximity to critical structures. Branches that violate safety constraints are pruned. Branches that pass all constraints are evaluated by the executive aggregation layer, which selects the optimal validated plan for promotion to execution.

The containment boundary is not a software guard. It is a structural separation between the planning layer and the execution layer. No speculative branch can reach the actuators without passing through the promotion gate, which requires all safety constraints to be satisfied. The robot can think aggressively while acting conservatively.

Personality-modulated speculation controls how broadly the engine explores. A conservative configuration explores near-modifications of the original plan. An aggressive configuration explores fundamentally different approaches. The surgeon can tune the exploration breadth based on the clinical situation.

What implementation looks like

A surgical robot deploying the forecasting engine maintains a real-time planning graph alongside its execution state. Pre-operative plans are the initial branches. Intraoperative discoveries trigger new branch generation. Each branch carries its own risk assessment, and the containment boundary ensures that planning activity is invisible to the robot's physical behavior until a plan is explicitly promoted.

For surgical device manufacturers, the forecasting engine provides a path to greater surgical autonomy without sacrificing safety. The robot can plan autonomously within the containment boundary while the surgeon maintains oversight through the promotion gate. The surgeon does not need to evaluate every speculative branch. They approve or reject the promoted plan.

For hospitals, governed speculative planning reduces procedure time by enabling faster recovery from unexpected findings. Instead of stopping and replanning manually, the robot presents a pre-validated alternative within seconds of encountering the unexpected condition.

Forecasting Engine All 21 steps →

Plan before you act. Contain speculation. Promote only what passes.

Primary Technical Disclosure
○ Forecasting and Executive Graphs in Autonomous Cognitive Systems
Secondary Technical
○ Planning Graphs as First-Class Cognitive Structures○ Containment Layer and Delusion Boundary○ Branch Classification System○ Personality Field as Structural Modifier○ Executive Engine Multi-Agent Graph Aggregation○ Branch Dormancy and Deferred Promotion○ Proactive Speculative Maintenance (Dream State)○ Planning Graph Archival for Cognitive Forensics○ Cross-Agent Planning Graph Visibility○ Slope-Constrained Speculative Simulation○ Structural Separation From Verified Memory○ Forecasting Engine Architecture○ Forecasting Execution Cycle○ Emotional Modulation of Planning○ Executive Graph Conflict Resolution○ Planning Graph Delegation and Forking○ Temporal Anchoring and Lifecycle Management○ Forecasting as Coordination Primitive○ Forecasting-Modulated Discovery Traversal○ Forecasting as Confidence Input○ Integrity-Constrained Forecasting○ Forecasting for Training Curriculum○ Biological Signal to Forecasting Coupling○ Substrate-Agnostic Forecasting Deployment
Applications (General)
● Surgical Robot Planning Through Governed Speculative Branches○ Defense Tactical Planning With Contained Speculation○ Forecasting Engine for Logistics Planning○ Forecasting Engine for Disaster Response Planning○ Forecasting Engine for Financial Portfolio Planning○ Forecasting Engine for Construction Project Planning○ Forecasting Engine for Epidemic Response Planning○ Forecasting Engine for Space Mission Planning
Applications (Specific)
○ da Vinci Plans Trajectories, Not Consequences○ Anduril's Lattice Plans Missions Without Speculative Containment○ Boston Dynamics Plans Motion, Not Missions○ Shield AI's Hivemind Cannot Contain Its Own Speculation○ MuJoCo Simulates Physics Without Planning Governance○ NVIDIA Isaac Sim Renders Worlds Without Governing Plans○ Unity ML-Agents Trains Without Governing Speculation○ Gazebo Simulates Robots Without Governing Their Plans○ Drake Optimizes Trajectories Without Governing Planning Structures○ robosuite Benchmarks Manipulation Without Governing Plans
Forecasting Engine overview →

AQ
deterministic
autonomy

Legal

Subject to one or more pending U.S. and international patent applications, see Patents for the current list and status. No license, express or implied, is granted. Any use requires a separate written agreement—see Licensing. Patent applications referenced on this site are pending. Claim scope, if any, is subject to examination and may issue in altered form or not at all. See Legal for terms and conditions.

Adaptive Query™ is a trademark of Nicholas Clark. U.S. federal registration is pending. federal registration. AQ™ , AQ Inside™ , Adaptive Index™ , Adaptive Network™ , Semantic Agent™ , @AQ™ , AQID™ , and Adaptive Coin™ are used as trademarks in connection with the Adaptive Query platform and brand. Other names may be trademarks of their respective owners.

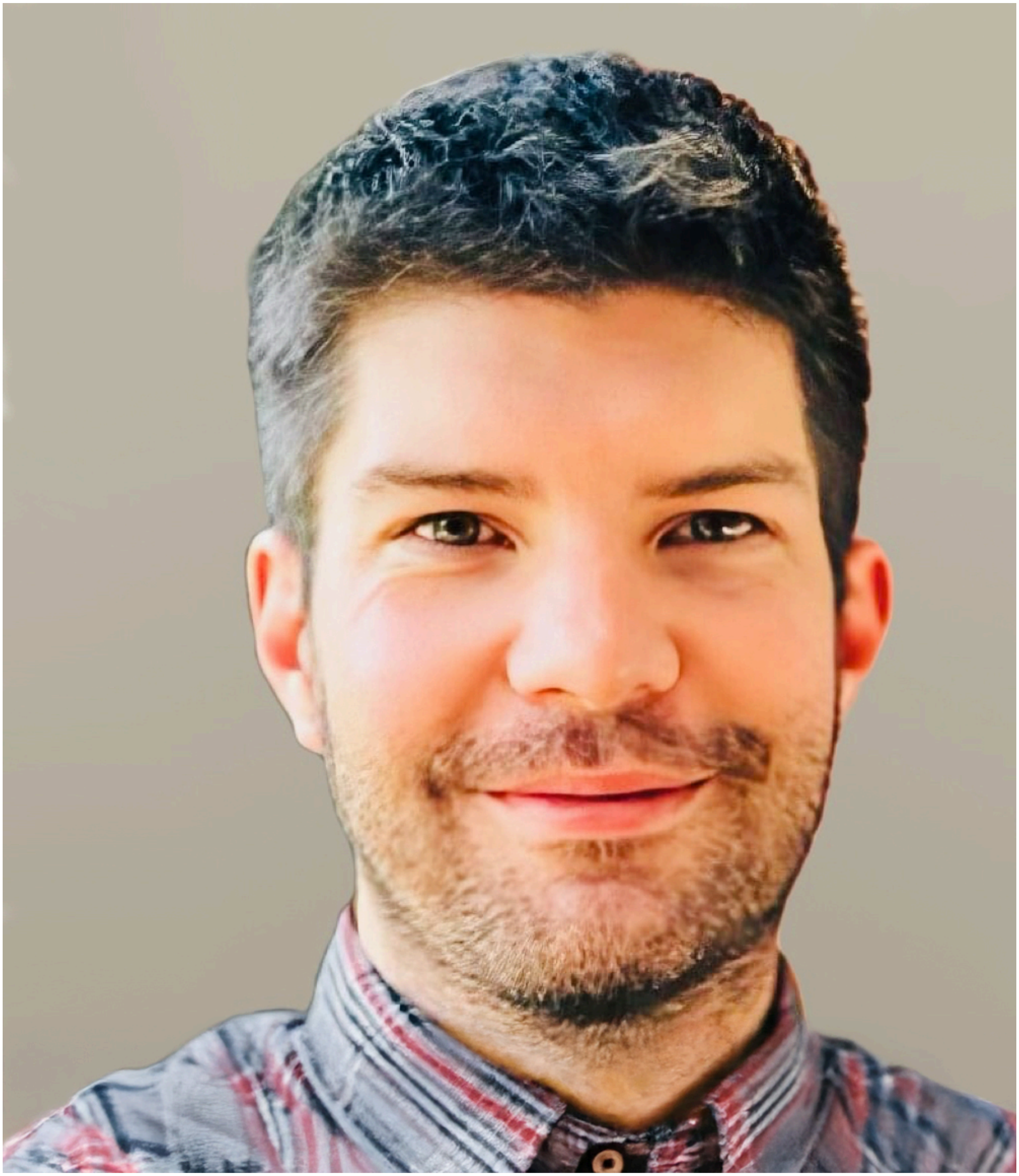
Platform operated by Adaptive Query LLC, which provides patent and trademark licensing services.

Last updated: 2026-03-03

-
- Inventive Steps
- Licensing
- Patents
- Articles

- Legal
- Opportunities
- Sitemap

- nick@qu3ry.net
- 72 28 14 36 01


Invented by Nick Clark | Founding Investors: Devin Wilkie