Invention Disclosure: Derivux for MATLAB

1. Administrative Information

Field	Value
Title	Derivux for MATLAB: Workspace-Aware AI Integration with Secure Code Execution
Inventor	Sebastian Hondl
Date of Conception	January 15, 2026
Status	Working Prototype (Private Repository)
Prior Public Disclosure	None

University Affiliation

Field	Value
Institution	University of Minnesota
Department	Mechanical Engineering
Development Context	Personal project, not university research

Independent Development Declaration

This invention was developed **independently** using:

- Personal computer and personal time
- Skills from coursework (not research assistantship)
- University-provided MATLAB license (standard student resource, available to all enrolled students)

Not used: Research funding, lab facilities, research staff, faculty collaboration, specialized equipment.

The MATLAB license is a standard educational resource (like library access or campus WiFi), distinct from research-specific resources that might trigger university IP claims.

---

2. Summary of Invention

Derivux integrates Claude AI into MATLAB/Simulink with:

1. **Workspace Awareness** — AI sees live variables, types, and values
2. **Secure Execution** — Sandbox blocks dangerous operations before running AI-generated code
3. **Simulink Manipulation** — Programmatic model creation, modification, and layout optimization
4. **Visual Capture** — Plots and diagrams displayed inline in chat
5. **Multi-Session Isolation** — Independent contexts per chat tab
6. **Agent Routing** — Pattern-based routing to specialized agents

---

3. Architecture

```
MATLAB Layer          Python Bridge          AI Layer
─────────────         ─────────────          ────────

DerivuxApp      ──►   MatlabBridge     ──►   MatlabAgent

CodeExecutor          SessionContext         MCP Tools

SimulinkBridge        AgentRouter            Claude API
```

```
WorkspaceContext
```

---

4. Novel Claims

Claim 1: Bidirectional Async Communication Bridge

Persistent Python event loop maintains AI client state across MATLAB's synchronous calls, enabling conversation memory without process spawning.

- **Ref:** `python/derivux/bridge.py:124–186`

Claim 2: Real-Time Workspace Introspection

Type-aware extraction of MATLAB workspace state, formatted for AI context with size-based truncation and priority ordering.

- **Ref:** `toolbox/+derivux/WorkspaceContextProvider.m:31–85`

Claim 3: Security-Sandboxed Code Execution

Domain-specific blocklist (system calls, eval chains, shell escapes) validates AI-generated code before execution, with configurable modes (prompt/auto/bypass).

- **Ref:** `toolbox/+derivux/CodeExecutor.m:21–27, 111–175`

Claim 4: Simulink Layout Algorithm

Five-phase graph layout (extraction → layer assignment → crossing minimization → coordinate assignment → wire routing) producing publication-quality diagrams.

- **Ref:** `toolbox/+derivux/SimulinkLayoutEngine.m` (807 lines)

Claim 5: Multi-Session Context Isolation

Per-tab conversation contexts within single process, with session switching that preserves state.

- **Ref:** `python/derivux/bridge.py:1414–1516`

Claim 6: Pattern-Based Agent Routing

Confidence scoring routes requests to specialized agents via explicit commands or keyword detection.

- **Ref:** `python/derivux/agents/specialized_agent.py:61-96`

---

5. Prior Art Differentiation

Feature	MATLAB Copilot	General AI (ChatGPT)	**Derivux**
Workspace awareness	Limited	None	Full
Code execution	None	None	Secure sandbox
Simulink manipulation	None	None	Programmatic
Inline figures	None	None	Yes

---

6. Key Implementation Files

Component	File	Lines
Orchestrator	`toolbox/+derivux/DerivuxApp.m`	385
Security	`toolbox/+derivux/CodeExecutor.m`	256
Simulink Bridge	`toolbox/+derivux/SimulinkBridge.m`	537
Layout Engine	`toolbox/+derivux/SimulinkLayoutEngine.m`	807
Python Bridge	`python/derivux/bridge.py`	1516
AI Agent	`python/derivux/agent.py`	590

Appendix A: Company Agreement Language

Background IP Carve-Out

> **Pre-Existing IP.** Inventor has pre-existing intellectual property disclosed to the University of Minnesota under [CASE ID], titled "Derivux for MATLAB" (the "Background IP"). This Background IP is excluded from any IP assignment in this Agreement. Company receives no rights to source code, designs, or specifications.

Limited Output License (if demonstrating)

> **Tool Outputs.** Company receives a non-exclusive, non-transferable license to use outputs generated by Inventor's tools solely within [NARROW FIELD OF USE]. This license excludes rights to underlying tools, methods, or IP, and terminates upon project completion.

Appendix B: Next Steps

- [ ] Submit to UMN Tech Commercialization → obtain case ID
- [ ] Confirm university does not claim ownership
- [ ] Before company engagement: include Background IP carve-out referencing case ID
- [ ] Before going public: file provisional patent application

---

Version 2.0 — January 27, 2026

Concise revision (detailed version in git history: 68516ad)