AI for Automation of Modeling and Simulation (M&S) Workflows

Sponsored by General Dynamics Electric Boat

Senior Design CSE Team #46 & ECE Team #37

Levin Mathew, John Morgan, Fajr Maqsood, Nikole Riera, Halmar Laing, Elizabeth Berry

Professor Jinbo Bi and Professor Park Sung-Yeul

Agenda

ENGINEERING CHALLENGE

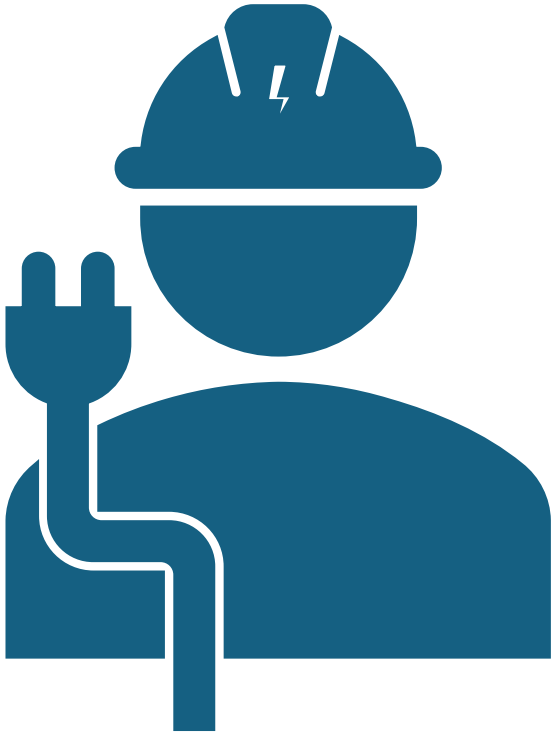
OBJECTIVE

CSE & EE TEAM RESPONSIBILITIES

SYSTEM PIPELINE

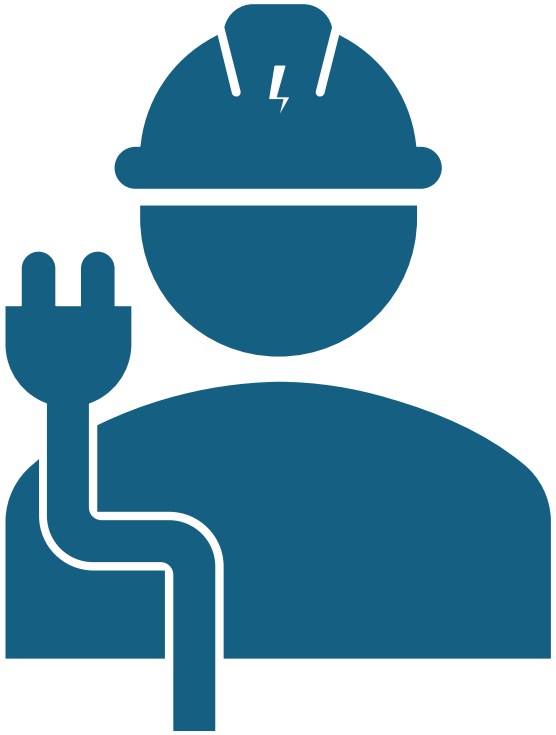
WORKING DEMO

FUTURE DEVELOPMENTS



Engineering Challenge

- Engineers at General Dynamics Electric Boat (GDEB) heavily rely on computer modeling and simulation (M&S) tools, like MATLAB/Simulink for design, testing, and troubleshooting.
- Tasks involving motor drives, inverters, DC-DC converters, and KPI analysis require running many repetitive simulations.
- Creating and modifying models is a manual process that requires engineers to write scripts, configure parameters, and interpret simulation results; tasks that consume valuable time and expertise.



Limitations

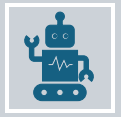
- GDEB is unable to provide the team with their data and models for training purposes
- This means we have to use open source Simulink models when training and testing
- Computational resources are limited as we need a lot of RAM to train and run models locally

Objectives

The project aims to leverage MATLAB/Simulink to improve the efficiency of M&S workflows.

The project will develop an AI-based tool to automate simulation tasks such as script generation, parameter configuration, and optimization.

The system will enable users to generate and run simulations using natural language prompts.



Automating repetitive tasks will improve engineering productivity.

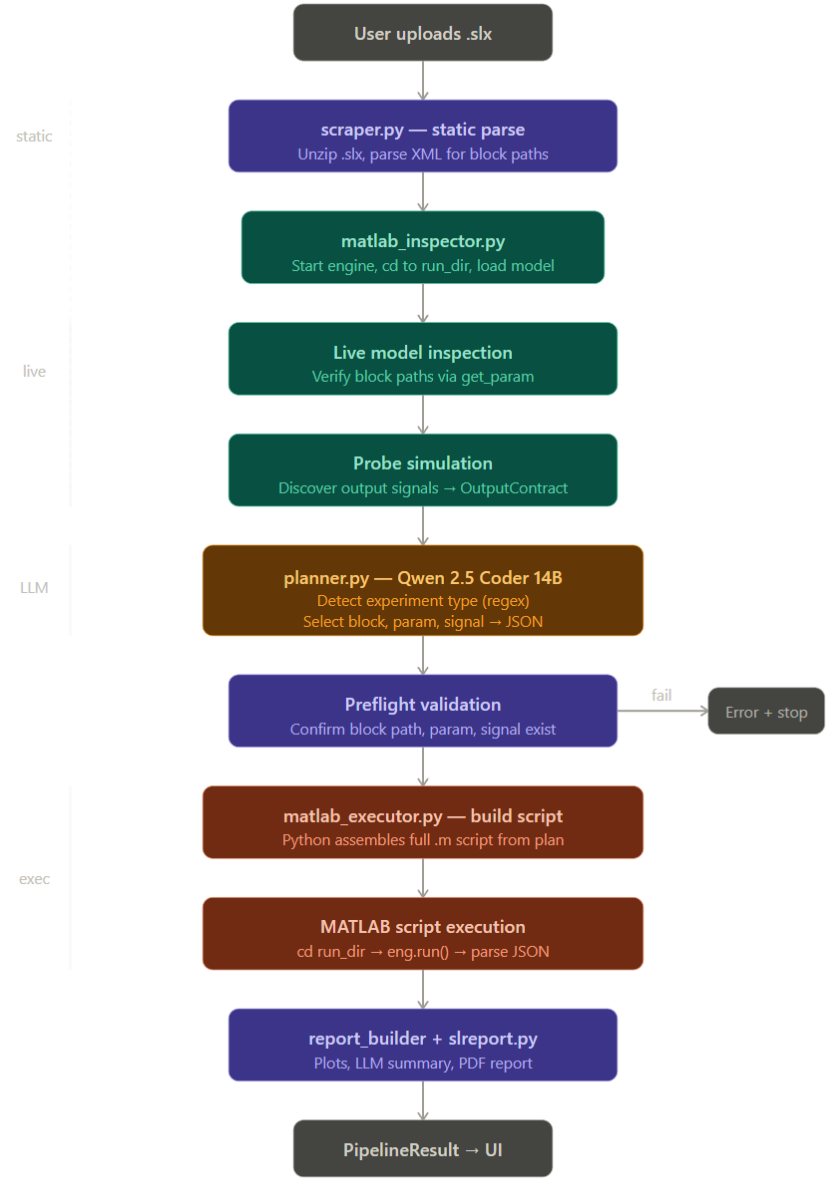
CSE Team Responsibilities

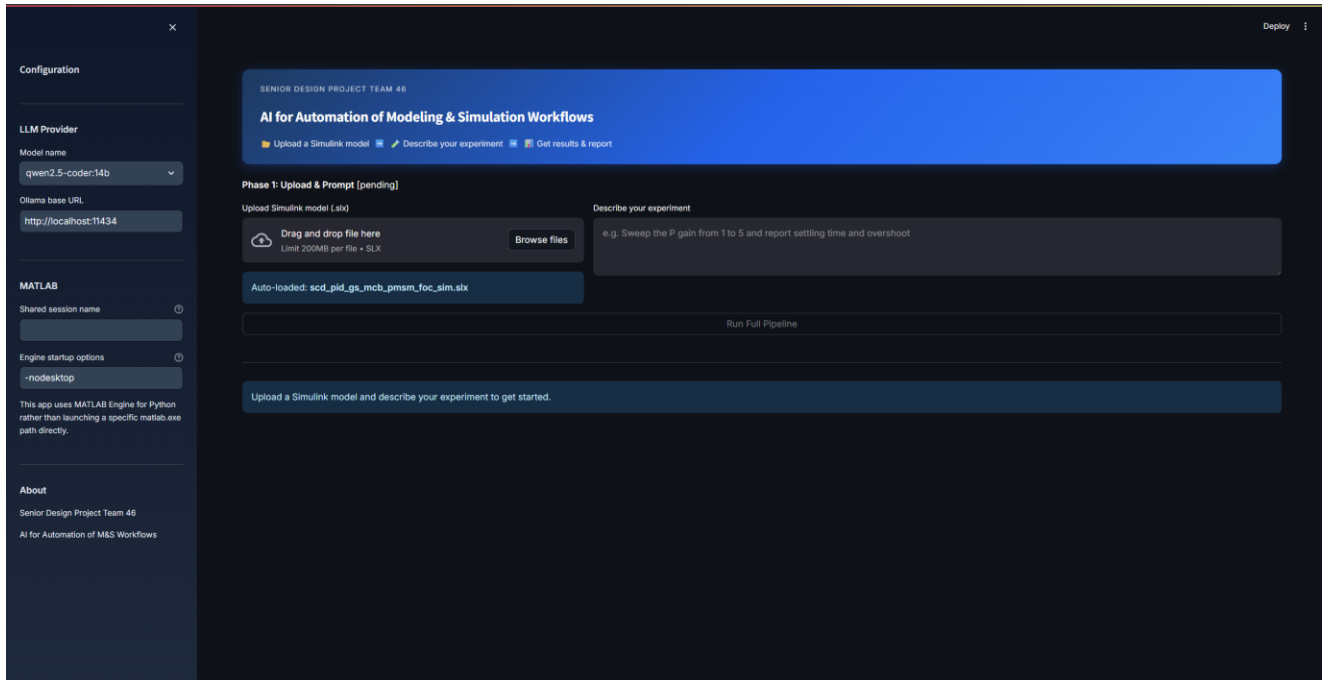
- Focused on developing the LLM and identifying the resources needed to support its implementation
- Model Selection, training, and fine-tuning
- Python GUI development
- Simulation Report & Analysis

ECE Team Responsibilities

- Assisted with training and developing an understanding of the various models involved
- Collected, cleaned, and provided the CSE team with data and parameters for use in LLM training
- Created prompts and scenarios for testing and training purposes
- Validated simulation report output
- Manual verification of LLM output and data analysis

System Pipeline



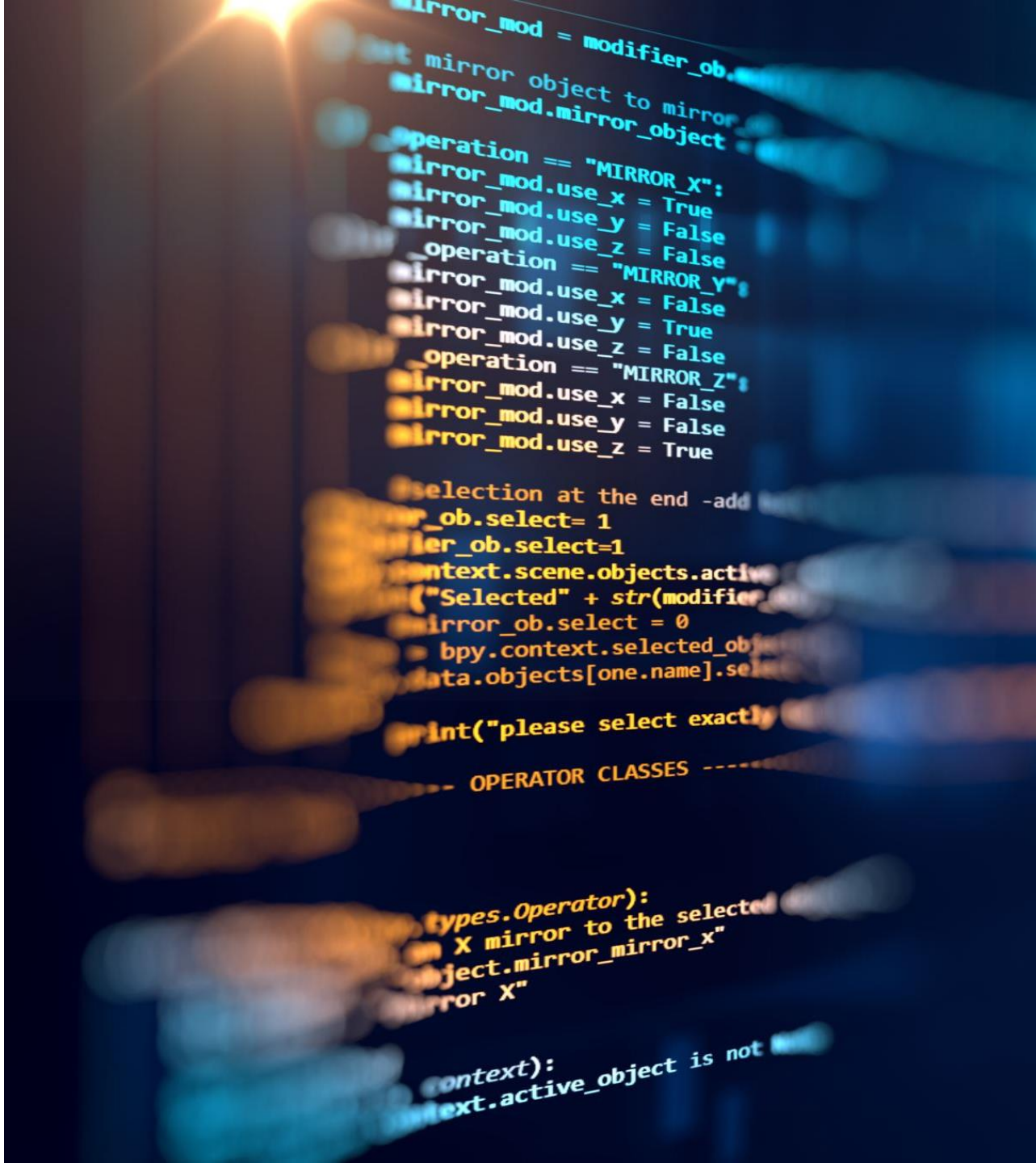


- The UI also prompts the user to upload a natural language prompt which is meant to be an experiment/analysis question about the Simulink model
- Upon launching our UI with Streamlit the user is prompted to upload a Simulink model in the form of a .slx file

User Interface

Scraper

- Opens .slx as ZIP → reads XML files
- Reconstructs full block hierarchy
- Extracts blocks, parameters, and connections
- Builds model summary for LLM context in natural language

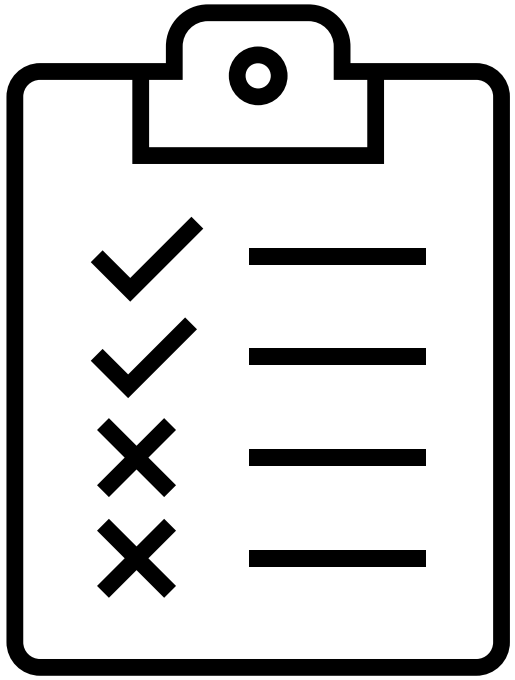


MATLAB Inspector

- Launches or connects to MATLAB engine
- Copies model to run directory
- Sets working directory
- Loads model into MATLAB

MATLAB Inspector – Verify Model

- Confirms block paths using MATLAB (find_system)
- Identifies tunable blocks (e.g., PID, Gain)
- Detects masked parameters
- Extracts current PID values



MATLAB Inspector – Probe Simulation

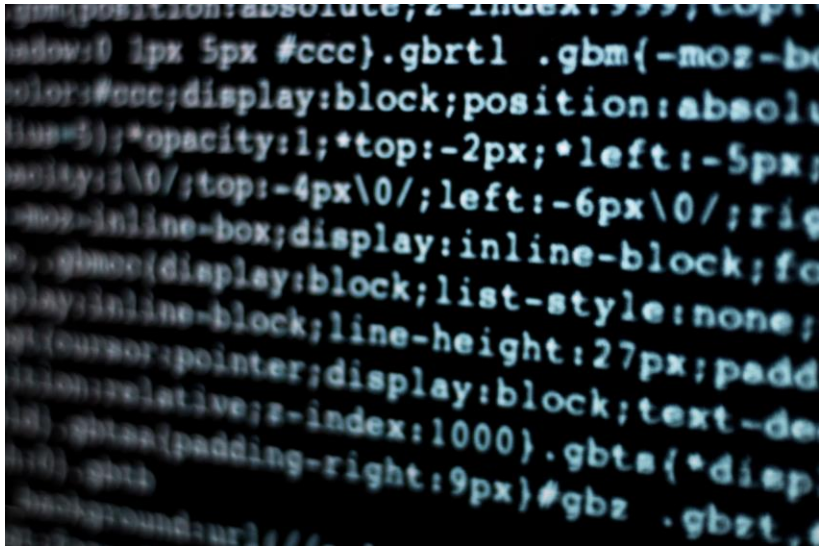
- Runs model once to detect outputs
- Checks outputs in priority order:
- logsout → yout → simlog → workspace
- Builds OutputContract (signal names + indices)

Planner

- Detects experiment type from prompt
- Filters valid tunable parameters
- Sends structured prompt to LLM
- LLM returns JSON experiment plan
- Python validates and finalizes plan

Planner - Preflight Validation

- Confirms block exists in model
- Verifies parameter is valid
- Ensures signal exists in outputs
- Stops early if errors found



MATLAB Executor

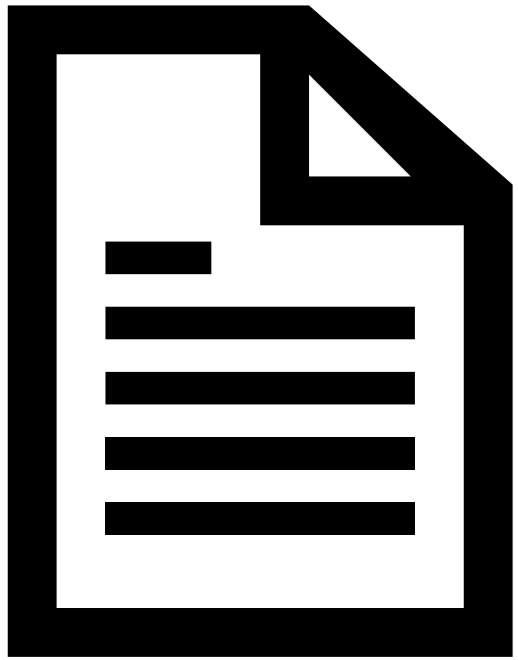
- Generates .m script entirely in Python
- Selects correct signal extraction method
- Adds loop (for sweeps) or single-run logic
- Appends metric calculations + helper functions
- Wraps script in try/catch

MATLAB Executor - Report Builder

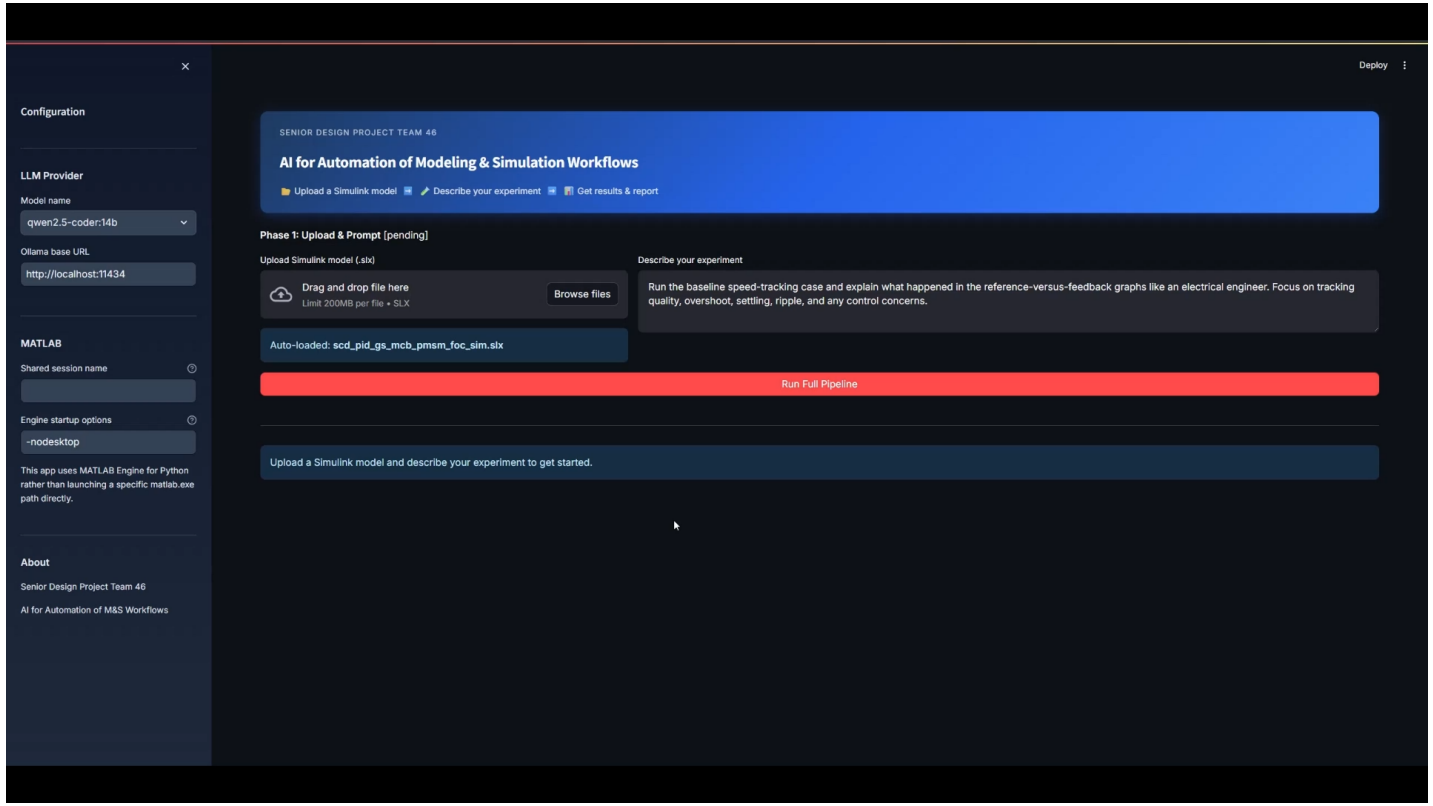
- Saves and runs script in MATLAB
- Extracts sdp_result struct
- Converts results → JSON → Python
- Builds structured RunResult
- Optional metric refinement

Report Output

- Generates plots (time + metrics)
- LLM writes experiment summary
- Creates PDF report (MATLAB tools)
- Returns results to UI for display/download



Demo


Configuration
LLM Provider
Model name
qwen2.5-coder:14b
Ollama base URL
http://localhost:11434
MATLAB
Shared session name
Engine startup options
-nodesktop
This app uses MATLAB Engine for Python rather than launching a specific matlab.exe path directly.
About
Senior Design Project Team 46
AI for Automation of M&S Workflows
Deploy
SENIOR DESIGN PROJECT TEAM 46
AI for Automation of Modeling & Simulation Workflows
Upload a Simulink model
Describe your experiment
Get results & report
Phase 1: Upload & Prompt [pending]
Upload Simulink model (.slx)
Drag and drop file here
Limit 200MB per file • SLX
Browse files
Auto-loaded: scd_pid_gs_mcb_pmsm_foc_sim.slx
Describe your experiment
Run the baseline speed-tracking case and explain what happened in the reference-versus-feedback graphs like an electrical engineer. Focus on tracking quality, overshoot, settling, ripple, and any control concerns.
Run Full Pipeline
Upload a Simulink model and describe your experiment to get started.

Output Verification

- While the reports do show promise, the EE team needed to verify manually the accuracy of the reports generated by the LLM
- The reports are generated with the following categories in order:
  - Experiment Overview, Local LLM Analysis, Model Architecture, Simulation Results, Plots, Generated MATLAB Script
- In order to verify these results, a script was written to check and verify the results of the baseline run.
  - If the baseline is able to run properly, the other experiments can run and compare to the baseline properly

Microsoft Teams

Call with Laing, Halmar

2026-04-06 21:33 UTC

Recorded by

Berry, Elizabeth

Future Developments

01

Build a bigger dataset of specific Electric-Boat models and train off those.

02

Use a bigger parameter model.

03

Add more agents to split up work to improve accuracy.

UCONN

COLLEGE OF ENGINEERING