Dissertation (MSU)

“Multiscale Computational Modeling of Surgical and Immune-Driven Outcomes: A Digital Twin Framework for Organ Systems and Predictive Medical Robotics”

▼ Formal Dissertation Plan

- A professional-grade research portfolio
- A grant/fellowship supplement
- A technical resume enhancer for AI, digital health, or computational science roles

Step 1: Structure of the PDF

Here’s the layout we’ll use (6–8 pages):

Page 1: Cover

Title:

Digital Twin Intelligence

Multiscale AI Systems for Immune-Guided Surgery, Organ Therapy, and Planetary Health

Author:

Ihunna Amugo, DDS Candidate, MHA, MS, REHS, PhD(c) in Computational Engineering

Affiliation:

Mississippi State University

PhD in Computational Engineering

Page 2: Abstract

A concise version of the abstract I provided above, framed for research and career relevance.

Page 3: Origin Story + Narrative Vision

A brief 2-paragraph story:

- Starting without engineering training
- Vision to unify dentistry, medicine, simulation, and AI
- Self-taught trajectory across simulation, ML, RL, and control
- Clinical motivation and scientific ambition

Page 4: 6-Year Timeline (Visual Summary)

Condensed view of your 72-month roadmap in a single visual (we'll format this as a table or infographic layout).

Page 5: Key Modules + Systems

Project	Domain	Summary
Immune-guided Surgery	FEM + RL + Dash	Inflammation field + agent + real-time control
Organ Therapy Simulator	Drug diffusion + AI	Cross-organ therapy with toxicity alerts
Digital Twin Dashboard	Python Dash	Multi-organ, real-time UI with sliders + visual overlays
PPO Curriculum Agent	RL (PyTorch)	Trains across complexity levels to optimize surgical behavior
Brain–Machine Interface	EEG + Kalman + RNN	Control surgical tool via synthetic or real brain data
Zoonotic Disease Model	ABM + PDE	Spread simulation across species with outbreak analytics

Page 6: Skill Stack + Tools

A chart or grouped list showing tools mastered:

- Math/Modeling: PDEs, FEM, stochastic processes, ODEs
- Programming: Python, C++, Fortran, Julia, MATLAB
- AI/ML: PyTorch, TensorFlow, Scikit-learn, PPO, PINNs
- Simulation Tools: OpenFOAM, Abaqus, COMSOL, PyBullet
- Data + Viz: Dash, Paraview, Tableau, SQL, SHAP
- Hardware: ROS, LSL (EEG), RL agents, robot arms

Page 7: Applications + Vision

Where this can go:

- NASA/space health
- Hospitals for immune-aware surgical planning
- Organ transplant optimization
- Zoonotic policy modeling
- Personalized adaptive cancer treatment

Page 8 (Optional): Roadmap to Completion + Contact

- Final modules in progress
- Timeline to dissertation defense
- Invitation for collaboration
- Email, GitHub, LinkedIn

Page 1: Cover Page

Digital Twin Intelligence

Multiscale AI Systems for Immune-Guided Surgery, Organ Therapy, and Planetary Health

Author:

Ihunna Amugo

DDS Candidate | MHA | MS Health Informatics | REHS

PhD Candidate in Computational Engineering

Mississippi State University

Contact:

Email: ihunna.amugo@email.com (placeholder – update if needed)

GitHub: github.com/ihunna-amugo (placeholder – update if applicable)

LinkedIn: linkedin.com/in/ihunna-amugo (optional)

Page 2: Abstract

Abstract

This dissertation presents a multiscale, AI-augmented digital twin platform designed to simulate human physiology, guide robotic surgery, predict immune rejection, and extend to planetary-scale modeling of health systems. The system integrates finite element modeling, reinforcement learning agents, drug diffusion simulators, and biologically inspired AI control to construct an end-to-end framework that bridges clinical medicine, computational engineering, and public health.

Unique to this work is its origin: I began with no formal engineering training—just a clinical foundation in dentistry and health informatics. Over six years, I systematically acquired skills in simulation, programming, and AI while building practical tools for surgical robotics, immune therapy, brain–machine interfacing, and cross-species modeling. Each project demonstrates not just technical capacity, but a deeper vision for how engineering can elevate medicine and public health through explainable, interactive, and intelligent digital systems.

Applications of this work include immune-aware surgical planning, organ transplant forecasting, adaptive cancer therapy, zoonotic outbreak simulation, and space medicine modeling. The ultimate goal is to demonstrate that digital twins are not abstract ideals—they are buildable, flexible platforms that can support real-world decision-making across individual, institutional, and global scales.

Page 3: Personal Narrative + Dissertation Vision. This page will articulate where you're coming from, what you're building, and why it matters — in your voice, with scholarly and personal authority.

Page 3: Personal Narrative & Vision

From Clinic to Code: A Story of Reengineering Purpose

I did not begin this journey as an engineer. I entered the world of computational modeling from the clinical trenches of dentistry and the systems-oriented world of public health and informatics. While studying the ethical, immunologic, and diagnostic complexities of oral pathology, I began to sense a void: too many medical decisions were reactive, fragmented, and poorly informed by system-level modeling. What if we could simulate a patient's physiology, anticipate immune complications, or visualize a therapy's risk in real-time? What if we could build systems as dynamic and adaptive as the body itself?

With that question, I began building from scratch—teaching myself Python, mathematical modeling, FEM, AI/ML, and control systems. I made mistakes. I iterated. But I never stopped. Over six years, I transformed that question into a platform: a set of intelligent digital twins that model the behavior of drugs, tissues, immune cells, brain signals, and robotic instruments across scales.

Dissertation Vision: Intelligence Across Biological and Robotic Systems

This dissertation is not just about AI or simulation. It is about the fusion of engineering, medicine, and ethics to build systems that respond to the complexity of life. The projects span:

- Tissue-aware surgery, where RL agents adapt to immune gradients
- Organ-level therapy planning, with dashboards that forecast toxicity
- Cross-species modeling, predicting zoonotic risks in planetary health
- Brain–machine interfacing, where synthetic EEG guides robotic precision
- Space physiology, adapting therapy models to altered gravity and motion

Each system is modular, explainable, and built with tools I learned from the ground up. This is not a dissertation in isolation. It is a functional blueprint for AI-powered medicine, rooted in the belief that health can be modeled, learned from, and improved—if we're bold enough to build it.

Page 4: The 6-Year Timeline overview. This will serve as a condensed visual summary of the 72-month roadmap, showing the structure, major milestones, and evolution of my dissertation.

Page 4: 6-Year Dissertation Timeline

Phase 1: Foundations (Months 1–12)

Objective: Learn programming, math, physics, simulation basics

Area	**Focus**
Programming	Python, C++, MATLAB, Julia, Fortran basics
Math	ODEs, PDEs, linear algebra, numerical methods

Simulation	1D/2D FEM, OpenFOAM/COMSOL starters
AI/ML	PyTorch, supervised learning, PINNs
Systems	CLI-based digital twin prototype, Dash UI, CLI modeling tools

Phase 2: Systems Integration (Months 13–24)

Objective: Build live simulators, integrate AI with therapy models

Area	**Focus**
AI Controllers	PPO, supervised dosing, threshold logic
Simulation	Multi-organ drug diffusion, tissue inflammation
UI	Interactive Dash digital twin planner
Fusion	Connect RL agents with toxicity alerts, forecasting, and tissue dynamics

Phase 3: Robotics, Neuroscience, and Hardware (Months 25–36)

Objective: Develop real-time control systems tied to surgical robotics and neural inputs

Area	**Focus**
Robotics	RL-controlled surgical tools, robot arm + FEM contact model
Neuroscience	RNN + Kalman brain–machine decoders
Integration	ROS, LSL, PyBullet, EEG simulator bridge
Strategy	Immune-based reward design + agent learning visualization

Phase 4: Adaptive Systems + Planetary Modeling (Months 37–54)

Objective: Generalize models to other organisms, health systems, and global disease threats

Area	**Focus**
Species	Multi-species FEM, zoonotic ABMs, vet AI tools
Brain Health	Stroke/seizure/disease spread simulators, connectome predictors
Art & Music	Behavioral AI from abstraction, therapy interactions
Space	ODE + RL models of zero-G adaptation, exosuit-guided treatment tools

Phase 5: Curriculum Learning + Optimization (Months 55–66)

Objective: Refine AI behavior, curriculum progression, policy-based learning

Area	Focus
PPO Curriculum	Train across scenario difficulty tiers
Strategy Visualizer	Compare RL, threshold, and fusion logic live
Planning Interface	Personalized risk input + replay + immune overlays

Phase 6: Deployment & Legacy (Months 67–72)

Objective: Finalize platform, defense preparation, global collaboration

Area	Focus
Dissertation	Write, defend, and submit core thesis chapters
Platform	Docker + GitHub version of system with UI, examples
Outreach	Present to NASA, WHO, hospitals, policy boards
Legacy	Launch roadmap, post-PhD plan, open-source toolkit release

Page 5: Key Projects and Modules, designed to showcase the most distinctive, interdisciplinary, and deployable systems you've built during your PhD.

Page 5: Key Projects and Modules

1. **Immune-Guided Surgical Simulator**

Tools: FEM, RL (PPO), Dash

Description:

A reinforcement learning agent navigates surgical fields based on real-time inflammation gradients and tissue properties. The agent learns to avoid high-risk immune zones while completing surgical objectives, guided by FEM-derived force–displacement feedback.

Use Case:

Training, planning, and intraoperative support for immune-aware surgery in oncology and transplant cases.

2. **Organ-Level Digital Twin + Drug Diffusion Engine**

Tools: PDE solvers, PyTorch AI controller, Dash

Description:

Simulates drug distribution across multi-organ systems using spatial PDEs. AI agents adjust dosing in real-time based on organ-specific risk, toxicity thresholds, and physiological dynamics.

Use Case:

Personalized therapy planning in liver disease, neuropharmacology, or chemotherapy delivery.

3. **Digital Twin Dashboard (Interactive UI)**

Tools: Python Dash, SHAP, risk forecast overlays

Description:

Live interface allows clinicians or researchers to adjust tissue stiffness, immune sensitivity, and drug parameters, with real-time updates to visuals, graphs, and risk alerts.

Use Case:

Hospital bedside interface for precision therapy or training simulator for surgical residents.

4. **PPO Curriculum Trainer + Strategy Visualizer**

Tools: PPO (PyTorch), Matplotlib/Dash visualizer

Description:

An RL agent trains across increasing levels of difficulty, developing surgical skillsets under different immune profiles. Visualizer compares performance against simpler baselines (threshold, static AI, random).

Use Case:

Explainable AI education, policy tuning for surgical autonomy levels.

5. **Brain–Machine Control Interface**

Tools: RNN, Kalman Filter, LSL (simulated EEG)

Description:

Synthetic neural signals decoded via RNNs or Kalman filters to control robotic instruments in real time. Built-in toggles allow switching between signal types, noise levels, or fusion models.

Use Case:

Robotic control for patients with paralysis or surgeons operating in constrained environments (e.g., space, disaster zones).

6. **Zoonotic Disease and Veterinary AI Simulator**

Tools: Mesa ABM, PDE modeling, supervised learning

Description:

Simulates infection dynamics across species, integrating mobility, biology, and environmental exposure. Veterinary AI tools provide dose recommendations based on cross-species pharmacokinetics.

Use Case:

Predictive tools for One Health initiatives, veterinary planning, or planetary disease surveillance.

Page 6: Skill Stack Overview, organized to reflect on my growth from foundational programming to advanced simulation, AI control, and platform integration. Serving as a strong technical backbone when presenting to advisors, employers, or grant reviewers.

Page 6: Skill Stack Overview

Mathematics & Modeling

- Ordinary Differential Equations (ODEs)
- Partial Differential Equations (PDEs)
- Finite Element Method (FEM)
- Numerical Optimization
- Stochastic Modeling
- Control Theory (Model Predictive Control, Reinforcement Learning)

Programming Languages

Category	**Languages**
General Purpose	Python, C++, Fortran, Julia

Scientific	MATLAB
Parallel	CUDA, MPI, OpenMP
Scripting & Data	Bash, R, SQL

AI/ML Frameworks

- PyTorch (including PPO, RNNs, curriculum learning)
- TensorFlow (used for PINNs, CNNs)
- Scikit-learn (classical models, interpretability)
- XGBoost / LightGBM (structured data)
- SHAP / Grad-CAM (model explainability)

Simulation Platforms

- OpenFOAM (CFD + PDE-based simulation)
- COMSOL / Abaqus (FEM and multi-physics modeling)
- PyBullet (robotics and physics scenes)
- ROS / ROS MoveIt (robot operating system for real tools)

Digital Twin & Visualization Tools

- Dash (main UI/UX platform for real-time apps)
- Plotly / Matplotlib (2D/3D plotting)
- Paraview (field simulation post-processing)
- Tableau (data visualization for health analytics)

High Performance Computing (HPC)

- MPI (domain decomposition and distributed simulation)
- OpenMP (shared-memory CPU parallelism)
- CUDA (GPU acceleration for physics or ML models)
- SLURM (job scheduling on clusters/supercomputers)

Medical / Biomedical Extensions

- LabStreamingLayer (EEG signal bridge)

- FEM-tissue integration for surgical contact modeling
- Agent-based modeling (Mesa, custom Gym environments)
- Connectomics (graph-based brain structure modeling)

Page 7: Applications and Vision, your capstone page. This section communicates why this body of work matters across sectors and how it can evolve into real-world impact.

Page 7: Applications and Vision

Clinical Impact

Your platform empowers precision medicine in ways few systems can. By modeling the interplay of drug dynamics, immune response, tissue mechanics, and surgical behavior, you offer clinicians:

- Immune-Aware Surgery: RL-guided agents trained to avoid inflamed regions during real-time procedures
- Organ-Specific Therapy Planning: Digital twins that forecast drug accumulation and toxicity across organ systems
- Brain–Machine Interfaces for Robotic Control: Potential use in stroke rehab, paralyzed patients, and zero-gravity microsurgery
- Explainable AI in Oncology and Transplant Planning: Dashboard-integrated agents with real-time overlays and interpretability

Public and Planetary Health

Your research extends beyond the human body. Your zoonotic modeling tools and multi-species FEM engines simulate:

- Cross-Species Disease Spread (e.g., avian influenza, COVID spillover)
- Veterinary Dose Predictors based on comparative pharmacokinetics
- Global Health Surveillance Tools for One Health frameworks

These can inform outbreak forecasting, inter-agency health interventions, and wildlife public health.

Space, Neuroscience, and Beyond

Your system supports extreme environments and frontier challenges:

- Space Physiology Modeling: Drug delivery under zero-G, bone density loss, and vestibular adaptation
- Connectomic AI: Modeling brain disease spread via structural connectomes
- Cognitive–Robotic Feedback Loops: From EEG to actuation in surgery, building closed-loop neural feedback systems

Vision Statement

This dissertation proves that computational engineers can build medical intelligence — not just algorithms. By fusing physics, AI, biology, and robotics, we are no longer asking “what if we could predict?” We are designing systems that do.

Page 8: Timeline to Completion + Contact. This closes out the PDF with a forward-looking structure and optional contact details for collaborators, advisors, or employers.

Page 8: Timeline to Completion + Contact

Final Milestones: Dissertation Timeline (Year 6)

Quarter	Objective	Deliverables
Q1	Final Integration Testing	Real-time dashboard + controller simulations across platforms
Q2	Writing + Defense Prep	Chapters: Background, Methods, Results, Use Cases, Ethical Design
Q3	Final Defense + Revisions	Dissertation defense, examiner feedback integration
Q4	Dissemination + Deployment	Public GitHub repo, Docker containers, open-source dashboard app

Post-PhD Trajectory

You are prepared for a rare breadth of impact roles:

- Clinical AI R&D (medtech, dental AI, surgical robotics)
- NASA or space–health simulation teams
- AI-enabled global health analytics and digital twin strategy

- Academic research in computational medicine, AI for neuroscience, or planetary health

You also have the basis for:

- A patentable therapy planning interface
- A translational grant proposal
- A signature open-source toolkit for digital twins in medicine

Contact & Links

Ihunna Amugo

PhD Candidate, Computational Engineering

Mississippi State University

DDS Candidate, Meharry Medical College

- Email: []
- 🧠 GitHub: []
- 📘 LinkedIn: []
- Project Portfolio: [insert if hosted on Notion, GitHub Pages, or elsewhere]

✅ **Next Step**

▼ Computer Foundations

ALL Concepts

Strategic Framework for Mastery

▼ **Phase 1: Systems Foundations (Weeks 1–6)**

Goal: Understand how computers actually work.

Topics:

- Turing Machine, CPU, Transistors
- Bits, Bytes, Binary, Hexadecimal, ASCII

- Memory: RAM, Memory Addressing, Stack vs Heap
- Machine Code, Assembly (optional), Executables
- Operating Systems: Kernel, Shell, CLI, SSH
- Networking: IP, DNS, TCP, SSL, URLs

Resources:

- CS50 by Harvard (free)
- Nand2Tetris (build a CPU from logic gates)
- Computer Networking: A Top-Down Approach

▼ **Phase 2: Programming Language Theory + Practice (Weeks 6–12)**

Goal: Get fluent in writing code and understanding how languages work.

Topics:

- Data types, variables, pointers, memory management
- Interpreted vs Compiled, Dynamically vs Statically Typed
- OOP (Classes, Inheritance, Methods)
- Procedural, Imperative, Declarative paradigms
- Learn Python, C++, and one FP-heavy language (e.g. Haskell or Scala)

  Resources:

  - Functional Programming Concepts (20 core ones first)
  - Structure and Interpretation of Computer Programs (SICP)
    - CS3110: OCaml-based FP course from Cornell
    - Crafting Interpreters by Bob Nystrom

▼ **Phase 3: Algorithms and Data Structures (Weeks 13–22)**

Goal: Build problem-solving fluency.

Topics:

- 20 core algorithms (searching, graph, sorting, dynamic, greedy)
- 10 core data structures

- Time/space complexity, Big-O
- Recursion, call stack, memoization
- Real-world problem solving (LeetCode, HackerRank)

  Resources:

  - Grokking Algorithms (visual)
  - MIT 6.006 on YouTube
  - Skiena’s Algorithm Design Manual

▼ **Phase 4: Software Engineering + Systems (Weeks 23–30)**

Goal: Connect coding to real-world software and runtime environments.

Topics:

- Compilation pipelines
- Virtual machines
- Multithreading, concurrency, parallelism
- APIs and interfaces
- Software design patterns
- Garbage collection, memory leaks

Resources:

- Operating Systems: Three Easy Pieces
- Design Patterns by the Gang of Four
- CS162 (Berkeley OS course)

▼ **Phase 5: Functional Programming Mastery (Weeks 31–40)**

Goal: Develop formal, composable reasoning.

Topics:

- All 40 advanced functional concepts (monads, lenses, etc.)
- Lambda calculus, algebraic data types
- Parser combinators, type inference

- Category theory lite (functors, applicatives)
- Building your own mini-FP language (optional)

  Resources:

  - Learn You a Haskell for Great Good!
  - Category Theory for Programmers by Bartosz Milewski
  - Scala with Cats

How to Learn It All Deeply

1. Code everything yourself - don't just read.
2. Explain aloud what you're doing - teaching reinforces memory.
3. Use spaced repetition (e.g. Anki) for dense material.
4. Rotate between building and reading - theory → project → debug → reflect.
5. Join one coding community or repo discussion weekly (e.g., GitHub, r/learnprogramming, FP Discords).
6. Track your mastery with a checklist and real output: functions written, projects deployed, bugs fixed.

Would you like this as a printable or editable roadmap tracker? Or would you prefer this converted into an Anki deck-style spaced repetition plan?

Proposed timeline (6 year model)

Year 1: Foundational Math, Programming, and Simulation

Focus: Build strong theoretical and programming foundations. Begin light modeling work.

Month	Focus Area	Topics & Tools
Months 1–3	Math Fundamentals	Linear Algebra (Strang MIT), Multivariable Calculus, Vector Calculus, Matrix Decomposition

	Programming I	Python (NumPy, SciPy, Matplotlib), Git, Jupyter, CLI basics
	Theory	Intro to PDEs (heat, wave, Laplace)
Months 4–6	Programming II	C++ (pointers, classes, STL), Julia (basic syntax, broadcasting, Plots.jl)
	Math Models	Intro ODEs & PDEs (first-order systems, boundary conditions)
	Simulation	Start OpenFOAM tutorials (mesh, solvers, turbulence)
Months 7–9	Numerical Methods	Finite Difference, Finite Volume, Newton-Raphson, Stability analysis
	HPC I	MPI Basics (MPI4Py, OpenMPI), Slurm job scheduling
	Visualization	Paraview basics, intro to Tableau dashboards
Months 10–12	Tools	MATLAB (Simulink, PDE toolbox), Julia DifferentialEquations.jl
	AI Intro	Python: Scikit-learn (regression, classification), TensorFlow (linear NN)
	Mini Project	Solve simple 2D heat/diffusion equation using OpenFOAM + Python wrapper

Year 2: Simulation, FEM, HPC, and First AI Models

Focus: Deep dive into mathematical modeling + full-stack simulation tools.

Month	**Focus Area**	**Topics & Tools**
Months 13–15	FEM I	Concepts: shape functions, Galerkin methods, 1D/2D FEM
	Software	COMSOL Multiphysics, Abaqus GUI simulations
	Data	SQL basics, R tidyverse, statistical testing
Months 16–18	Biological Models	Immune modeling, diffusion-reaction equations, tumor growth PDEs
	AI II	PyTorch (CNNs, LSTM), TensorFlow (functional API)
Months 19–21	Optimization	Gradient descent, constrained optimization, adjoint methods

	HPC II	CUDA fundamentals (Thrust, memory mgmt), OpenMP parallelism
Months 22–24	Mini Project II	3D tumor growth model in OpenFOAM + PyTorch PINN surrogate
	Paper 1	Publish review on “AI + FEM for medical simulation” or replicate a core study

Year 3: Coupling AI + Simulation, Robotics Models, Real-World Data

Focus: Begin dissertation prototyping with AI/FEM fusion and surgical relevance.

Month	**Focus Area**	**Topics & Tools**
Months 25–27	Robotics I	Kinematics, force modeling, surgical robot dynamics
	Reinforcement Learning	TensorFlow Agents, StableBaselines3
Months 28–30	Digital Twin I	Real-time FEM solver + TensorRT AI inference on CUDA
Months 31–33	Data II	Integrate patient data from real case studies (brain trauma, transplant)
Months 34–36	Project III	Brain perfusion under ischemia (COMSOL) + PINN model for outcome prediction
	Paper 2	Submit to Journal of Computational Surgery or IEEE TBME

Year 4: Full Dissertation Systems + Collaboration

Focus: Complete core models, HPC validation, publish impactful results.

Month	**Focus Area**	**Topics & Tools**
Months 37–39	Coupled Modeling	FEM + Agent-based hybrid (tumor + immune), stochastic + deterministic methods
Months 40–42	HPC III	Scale simulations on cluster (MPI + CUDA + Slurm), use real-time dashboards

Months 43–45	Project IV	Graft rejection predictor (immune cell dynamics) + Tableau dashboard
Months 46–48	Paper 3	Submit to Computers in Biology and Medicine or Nature Digital Medicine

<u>Year 5: Robotics + Surgical Sim, Validation + Dissemination</u>

Focus: Apply your framework in surgical planning, optimize tools, prep defense.

Month	Focus Area	Topics & Tools
Months 49–51	Robotics II	Real-time path planning + incision feedback using Abaqus + PyTorch
Months 52–54	Validation	Compare model vs. clinical outcomes (transplant, oncology, trauma data)
Months 55–57	Wrap Models	Create user-friendly pipeline (Jupyter dashboard, CLI tools, APIs)
Months 58–60	Paper 4 & 5	Final high-impact papers: Physics-informed Digital Twins for Surgery

<u>Year 6: Defense, Dissemination, Career Development</u>

Focus: Defend, present, submit dissertation, transition to postdoc/industry/startup.

Month	Focus Area	Topics & Tools
Months 61–63	Dissertation Writing	Compile all results, structure chapters
Months 64–66	Defense Prep	Mock defenses, presentations, stakeholder review
Months 67–69	Conference Tour	Present at ASME, IEEE EMBS, NIH or NSF-funded symposia
Months 70–72	Career	Apply for postdocs (NASA, NIH), faculty, or spin-off/startup track

By end of year 6:

All Required Programming Languages

Major Simulation Frameworks

AI + ML Pipelines for Medicine & Robotics

Multiscale Modeling (PDE + FEM + Agent Models)

Real-Time Systems & HPC (MPI, CUDA, Slurm)

Scientific Publication & Real-World Validation