

# Haptic Lathe Simulator

MAE 6800: Design and Control of Haptic Systems, Fall 2025

Caleb Farrelly, Daria Kot, and Richard Zheng  
*Sibley School of Mechanical and Aerospace Engineering*  
*Cornell University*  
Ithaca, NY, USA

**Abstract**—We present the design and implementation of a haptic lathe training device that simulates the tactile experience of operating a manual lathe. The system integrates a directly-driven handwheel with a virtual lathe environment, providing force feedback based on tool-material interaction, spindle speed, and cutting depth. Using a geared DC motor with a quadrature encoder as the haptic interface and a Processing-based graphical user interface, the device enables trainees to develop muscle memory for common machining operations in a low-risk setting. The haptic rendering algorithm employs a damping-based virtual wall model derived from the Hapkit framework, with material-specific force profiles for Delrin, Aluminum 6061, Stainless Steel 316, and Inconel. This paper details the system architecture, hardware implementation, control algorithms, and force rendering equations.

**Index Terms**—haptics, lathe simulator, force feedback, virtual machining, training systems

## I. INTRODUCTION

### A. Motivation

Traditional lathe training relies on access to full-scale machines and close expert supervision, and it carries inherent risks including injury, tool damage, and material waste. Novice machinists must develop the tactile “feel” of cutting—recognizing overly aggressive feeds, dull tools, chatter onset, and material-dependent behavior at the tool–workpiece interface. Developing this intuition typically requires extensive hands-on practice.

Haptic technology offers a safer and more accessible alternative: virtual training environments that provide force feedback without the cost or hazard of physical machining. We present a two-DOF haptic lathe simulator that reproduces the hand-feel of manual lathe operation using a directly driven handwheel that can be toggled between the X cross-slide and Z carriage axes. A virtual lathe environment renders stock geometry, tool position, target drawings, and tolerances, and live X/Z readouts. The system computes and reflects cutting forces and torques as functions of spindle speed, feed, and depth of cut, enabling trainees to practice core operations such as facing and turning to diameter while building correct muscle memory in a low-risk setting.

### B. Educational Objectives

The Haptic Lathe Simulator is designed to achieve the following educational objectives:

- 1) **Muscle Memory Development:** Users develop proper hand coordination for feed control by experiencing realistic resistance during cutting operations.
- 2) **Material Differentiation:** Different materials (Delrin, Aluminum, Stainless Steel, Inconel) present distinct force profiles, teaching users to recognize and adapt to material properties.
- 3) **Safety Awareness:** The system simulates crash conditions when the tool contacts the workpiece with the spindle stopped and provides increased haptic resistance for overly aggressive cutting attempts, reinforcing safe operating procedures and proper feed control.
- 4) **Cutting Parameter Understanding:** Users experience how spindle speed, feed rate, and depth of cut affect cutting forces, developing intuition for optimal parameters.

### C. Prior Work in Haptic Machining Simulation

A complementary user study by [7] examined the perceived importance of haptic cues at the lathe handwheel across beginner, intermediate, and expert operators. Using a questionnaire-based methodology and chi-square analysis, the study found a statistically significant relationship between operator experience and perceived force intensity: beginners reported strong reliance on haptic feedback, whereas expert users perceived weaker force cues. These findings suggest that physically meaningful handwheel feedback plays a critical role during early motor-skill acquisition and motivate the inclusion of haptics in lathe training simulators.

Significant prior work has explored force-feedback interfaces for virtual machining, with an emphasis on modeling tool–workpiece interaction and material removal in real time. Early systems such as the Haptic Virtual Turning Operation System (HVTOS) employed efficient workpiece deformation algorithms to generate continuous force feedback as a function of cutting depth and material properties [1]. However, force feedback in this work was rendered using a commercial PHANTOM haptic device, which, while capable of high-fidelity kinesthetic output, is expensive and does not reflect the mechanical interaction of a manual lathe, where feedback is transmitted through a rotating handwheel.

Subsequent research expanded virtual machining toward immersive visual and mixed-reality training platforms, emphasizing three-dimensional visualization, task sequencing, and safety instruction rather than physically rendered cutting forces

[2]–[4]. In many of these systems, user interaction is limited to kinematic input—such as rotating a handwheel or manipulating a virtual tool—with explicit force or torque feedback. Educational implementations such as the *Turnin’ and Burnin’* lathe similarly focus on visual understanding of machining operations and geometry while omitting continuous, physically meaningful haptic feedback through the handwheel [9].

In contrast, the haptic lathe simulator presented in this work emphasizes the continuous, parameter-dependent hand feel of manual lathe operation. By using a directly driven handwheel and a damping-based force rendering model explicitly tied to feed rate, spindle speed, depth of cut, and material properties, our system prioritizes kinesthetic realism and muscle-memory development.

## II. SYSTEM OVERVIEW

The haptic lathe simulator consists of three primary subsystems:

- 1) **Haptic Interface:** A CQR37D geared DC motor with integral quadrature encoder, directly coupled to a handwheel that serves as both position input and force feedback output.
- 2) **Virtual Environment:** A Processing-based graphical user interface that renders the lathe workpiece, tool position, and material removal in real-time.
- 3) **Control Bridge:** A Python-based communication layer that manages bidirectional data flow between the GUI and microcontroller while computing haptic forces.

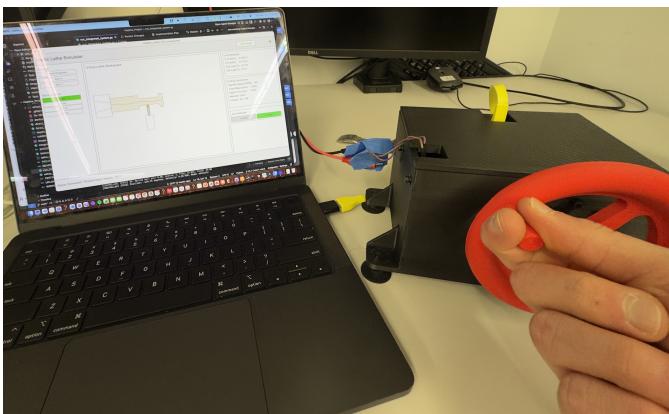


Fig. 1. Haptic Lathe Simulator Setup

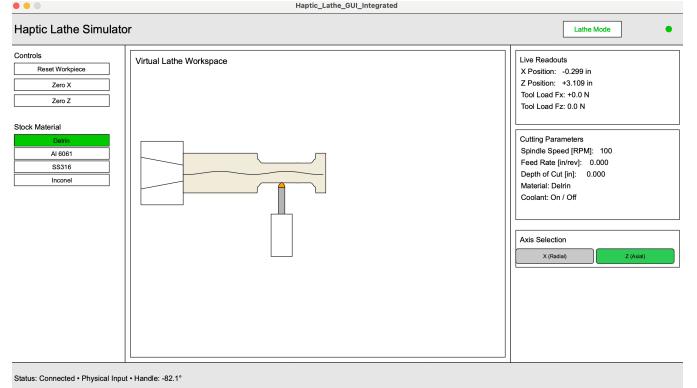


Fig. 2. Virtual Environment showing Delrin being cut.

### A. Project Scope

The presented work demonstrates the principles of machining with real-time force feedback to replicate the two primary control inputs of a conventional manual lathe: the X cross-slide and the Z carriage. The resultant device is a two-DOF haptic trainer featuring one directly-driven handwheel to ensure high-fidelity transmission of simulated forces, with the operator being able to rapidly switch between the two axes. By concentrating the design on a low-inertia, directly-coupled system, the device minimizes parasitic friction and backlash, which are critical design considerations in high-precision mechanical assemblies. The system computes and reflects realistic cutting forces and torques based on operational variables, allowing trainees to practice basic machining operations (e.g., facing, turning to a specified diameter) in a safe virtual environment.

### B. Virtual Wall Rendering

Virtual walls are fundamental haptic primitives that create the sensation of solid surfaces. When a user penetrates a virtual wall, the haptic device generates a restoring force proportional to penetration depth:

$$F = k \cdot x \quad (1)$$

where  $k$  is the wall stiffness (N/m) and  $x$  is the penetration depth (m). Pure spring walls can feel “buzzy” at high stiffness values, motivating the addition of damping:

$$F = k \cdot x + b \cdot \dot{x} \quad (2)$$

where  $b$  is the damping coefficient (Ns/m) and  $\dot{x}$  is the penetration velocity (m/s).

### C. Cutting Force Modeling

In actual machining, the feed force experienced by the lathe’s feed mechanism is given by:

$$F_f = K_c \cdot f \cdot a_p \quad (3)$$

where  $K_c$  is the specific cutting force of the workpiece material ( $N/mm^2$ ),  $f$  is the feed per revolution (mm/rev), and  $a_p$  is the depth of cut (mm). This relationship guides our haptic rendering strategy: cutting forces should increase with feed rate (velocity) and depth of cut (penetration).

### III. SYSTEM ARCHITECTURE

#### A. Architecture Overview

The system employs a three-tier architecture with clear separation of concerns. Fig. 3 shows the data flow between components.

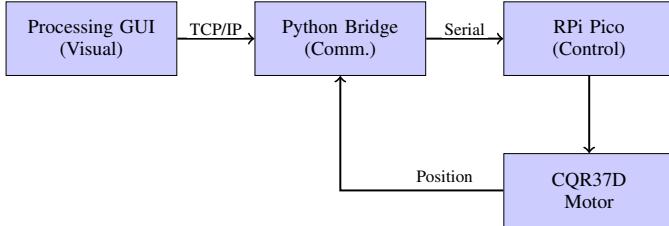


Fig. 3. System architecture showing data flow between components.

#### B. Communication Protocol

The Processing GUI sends JSON-formatted commands to the Python bridge via TCP/IP on port 5005. High-speed force commands are sent at 100Hz using the format: FORCE:<Fx>,<Fz>,<Freq>,<Yield>.

The bridge sends text commands to the Pico microcontroller via serial at 921600 baud, including commands such as spring\_wall, status, stop, and zero.

#### C. Update Rates

Table I shows the system update rates to ensure responsive haptic feedback.

TABLE I  
SYSTEM TIMING PARAMETERS.

Component	Rate	Purpose
Pico control loop	100 Hz	Force rendering, motor control
Bridge main loop	500 Hz	Communication management
GUI rendering	60 Hz	Visualization
Position polling	100 Hz	Encoder/state feedback
Safety checks	20 Hz	Timeouts, limits

### IV. HARDWARE IMPLEMENTATION

#### A. Component Selection

The haptic interface requires a motor capable of sufficient torque for realistic cutting force simulation, low friction for transparent backdrivability, and integrated encoder for position sensing.

We selected the **CQRobot CQR37D** geared DC motor. Table II shows its specifications. The 30:1 gear ratio provides mechanical advantage, amplifying motor torque while reducing output speed—appropriate for the slow, deliberate movements of lathe handwheel operation.

The **Raspberry Pi Pico** provides dual-core ARM Cortex-M0+ at 133 MHz, hardware PWM for motor control, GPIO interrupts for encoder counting, and MicroPython support for rapid development.

TABLE II  
CQR37D MOTOR SPECIFICATIONS

Parameter	Value
Gear Ratio	30:1
Encoder CPR	64 counts/rev (motor shaft)
Effective Resolution	1920 counts/output rev
Rated Voltage	12V DC
No-Load Speed	150 RPM
Stall Torque	3.1 Nm

TABLE III  
HARDWARE PIN CONNECTIONS

Component	Pico Pin	GPIO	Color
Encoder A	Pin 9	GP6	White
Encoder B	Pin 10	GP7	Yellow
Motor ENA	Pin 1	GP0	—
Motor IN1	Pin 4	GP2	—
Motor IN2	Pin 2	GP1	—

#### B. Wiring Configuration

Table III shows the hardware pin connections.

#### C. Bill of Materials

Table IV shows the complete bill of materials.

TABLE IV  
BILL OF MATERIALS (EXCLUDING 3D-PRINTED PARTS).

Item	Qty	Cost
Raspberry Pi Pico	1	\$4.00
CQR37D geared DC motor (30:1)	1	\$34.00
Motor driver (H-bridge)	1	\$15.70
Shaft coupler	1	\$4.50
<b>Total</b>		<b>\$58.20</b>

#### D. Mechanical Design

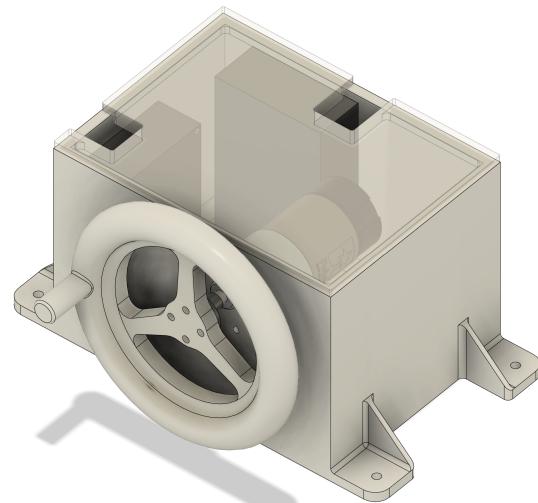


Fig. 4. CAD of the Lathe Simulator with transparent lid.

The mechanical assembly (Fig. 4) includes: 3D-printed housing with integrated mounting for motor and boards, 3D-printed handwheel (90mm OD) with ergonomic grip, shaft coupler connecting motor output to handwheel, and flanged base for suction cup attachment that can be clamped to the table.

## V. SOFTWARE IMPLEMENTATION

### A. Motor Control (*MicroPython*)

The motor controller runs on the Raspberry Pi Pico, implementing real-time position tracking and force rendering.

1) *Quadrature Encoder Decoding*: The encoder uses hardware interrupts to track motor position. Position in degrees is computed as:

$$\theta = \frac{\text{encoder\_count}}{\text{CPR} \times \text{gear\_ratio}} \times 360^\circ \quad (4)$$

For our system:  $\theta = \frac{\text{count}}{64 \times 30} \times 360^\circ = \frac{\text{count}}{1920} \times 360^\circ$ .

2) *Velocity Calculation*: Angular velocity is computed by differentiating encoder counts:

$$\omega = \frac{\Delta \text{count}}{\text{CPR} \times \text{gear\_ratio}} \times \frac{60}{\Delta t} \quad [\text{RPM}] \quad (5)$$

An IIR low-pass filter smooths the velocity estimate:

$$\dot{x}_{\text{filt}} = 0.9 \cdot \dot{x} + 0.1 \cdot \dot{x}_{\text{prev}} \quad (6)$$

### B. Virtual Wall Force Rendering

The haptic feedback algorithm implements a damping-based cutting force model inspired by the Hapkit framework.

### C. Control Loop and Force Pipeline

Fig. 5 summarizes the haptic rendering pipeline executed each control cycle. The Pico runs a fixed-rate loop at 100 Hz. Encoder counts are converted to angle  $\theta$ , differentiated to estimate angular velocity  $\omega$ , filtered, and then mapped to a handle-space linear velocity  $\dot{x}_{\text{filt}} = \omega_{\text{filt}} r_h$ . Penetration depth  $x_{\text{pen}}$  is computed relative to the current virtual wall location. The damping-based cutting force in (8) is then converted to motor torque (9) and mapped to PWM duty (10). Finally, direction and safety logic clamp output based on penetration limits and heartbeat timeout.

1) *Parameterization Used in This Prototype*: The prototype used the following fixed parameters: handle radius  $r_h = 0.05$  m; base damping  $c_{\text{base}} = 2000$  Ns/m; depth scaling saturation  $d_{\text{scale}} = \min(x_{\text{pen}}/0.005, 2.0)$ ; maximum penetration of 200°; and a 5 s heartbeat timeout. Material differences were implemented by scaling the effective damping via the yield force  $F_{\text{wall}}$  (Table V) as  $c_{\text{damping}} = c_{\text{base}} \cdot (F_{\text{wall}}/50)$ , producing a 4:1 spread between Delrin and Inconel.

2) *Penetration Detection*: When the tool enters the workpiece, penetration depth is computed:

$$x_{\text{pen}} = (\theta_{\text{current}} - \theta_{\text{wall}}) \cdot \text{dir} \cdot \frac{\pi}{180} \cdot r_h \quad (7)$$

where  $r_h = 0.05$  m is the effective handle radius.

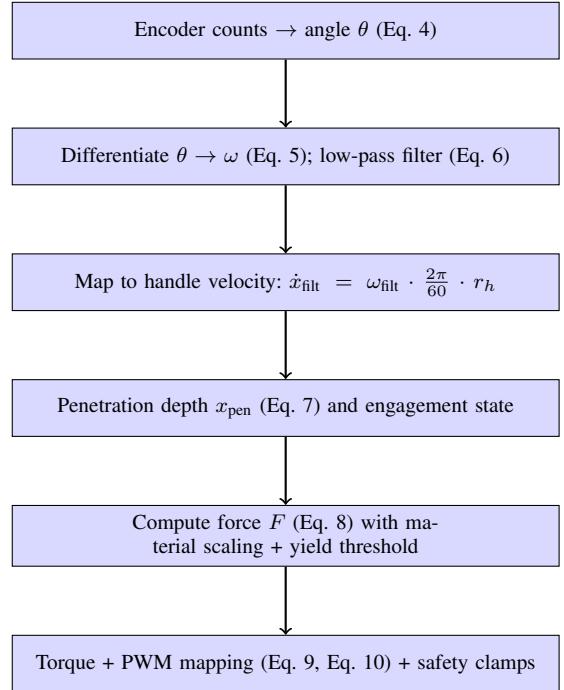


Fig. 5. Force rendering pipeline executed each Pico control cycle.

3) *Damping-Based Force Calculation*: The cutting force is computed using velocity-proportional damping, scaled by penetration depth:

$$F = -\dot{x}_{\text{filt}} \cdot c_{\text{damping}} \cdot (1 + d_{\text{scale}}) \quad (8)$$

where  $\dot{x}_{\text{filt}}$  is the filtered velocity at handle (m/s),  $c_{\text{damping}} = c_{\text{base}} \times \frac{F_{\text{wall}}}{50}$  is the material-scaled damping with  $c_{\text{base}} = 2000$  Ns/m, and  $d_{\text{scale}} = \min(\frac{x_{\text{pen}}}{0.005}, 2.0)$  is the depth scaling factor.

4) *Torque-to-PWM Conversion*: The computed force is converted to motor torque and then to PWM duty cycle using the Hapkit non-linear mapping:

$$T_p = \frac{F \cdot r_h}{\text{gear\_ratio}} \quad (9)$$

$$\text{duty} = \sqrt{\frac{|T_p|}{0.03}} \quad (10)$$

This square-root relationship linearizes the perceived force output.

### D. Material Properties

Different materials are simulated by varying the wall stiffness  $k_{\text{wall}}$  and yield force. Table V shows the material haptic properties.

The 4:1 ratio between softest (Delrin) and hardest (Inconel) materials provides clear tactile differentiation.

### E. Graphical User Interface

The Processing-based GUI provides: (1) Virtual Lathe Workspace with real-time rendering of stock geometry, tool position, and material removal; (2) Live Readouts of X/Z position, tool load, and cutting parameters; (3) Control Panel

TABLE V  
MATERIAL HAPTIC PROPERTIES

Material	$k_{\text{wall}}$ (N/m)	Yield (N)	Color
Delrin	50,000	25	Cream
Al 6061	100,000	50	Silver
SS316	150,000	75	Steel
Inconel	200,000	100	Bronze

for material selection, axis selection, and reset/zero functions; and (4) Connection Status with visual indicator for bridge connectivity.

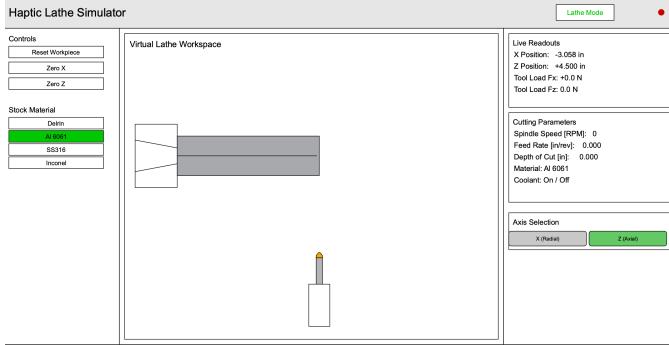


Fig. 6. Virtual Environment showing the lathe workspace.

1) *Collision Detection*: The tool uses a hyperbolic profile for realistic V-tool geometry:

$$y = \sqrt{(s \cdot x)^2 + R^2} - R \quad (11)$$

where  $s$  is the slope factor and  $R$  is the tool tip radius. Collision is detected when the tool profile intersects the stock profile array at any Z-position.

2) *Vibration Frequency Calculation*: Cutting vibration frequency is mapped from surface feet per minute (SFM):

$$\text{SFM} = \frac{\text{RPM} \times D \times \pi}{12} \quad (12)$$

$$f_{\text{vib}} = \text{SFM} \quad [\text{Hz}] \quad (13)$$

This 1:1 mapping (10 SFM = 10 Hz) provides intuitive correspondence between spindle speed and haptic vibration.

## VI. RESULTS

### A. Design Decisions

Several key design decisions shaped the final system:

#### 1) Damping-based vs. Spring-based Force Rendering:

We chose velocity-proportional damping over pure spring walls because cutting forces in actual machining are primarily determined by feed rate (velocity) rather than static position. This produces a more realistic “cutting feel.”

#### 2) Single-Axis Interface:

While a complete lathe has two axes (X and Z), we implemented a single handwheel with software axis selection. This reduces hardware

complexity while maintaining training value through the axis toggle feature.

- 3) **Motor as Input and Output**: Using the motor encoder as the position input (rather than a separate sensor) enables the same device to serve as both input transducer and force actuator.

### B. Limitations

- 1) **Single Degree of Freedom**: Real lathe operation involves simultaneous X and Z control. The single-axis interface requires users to mentally switch between axes.
- 2) **No Force Sensing**: The system renders forces based on virtual geometry only; actual cutting force measurement would require additional sensors.
- 3) **Motor Backdrive Friction**: The geared motor introduces friction that cannot be fully compensated, affecting transparency in free motion.

### C. User Study

A formative user study was conducted during a public demonstration of the haptic lathe simulator to evaluate perceived realism, usefulness of the haptic cues, and overall learning impact. The study was exploratory (not a controlled experiment) and focused on subjective perception and qualitative feedback.

1) *Procedure*: Participants interacted freely with the simulator and were encouraged to (i) toggle between axes, (ii) try multiple materials, and (iii) vary depth of cut and feed behavior to observe changes in resistance and vibration. After the interaction, participants completed a short questionnaire with 10-point Likert items (1 = strongly disagree, 10 = strongly agree) and optional open-ended comments. Participants also answered whether they felt more confident operating a lathe afterward (Yes/No).

2) *Participants*: Ten participants completed the questionnaire ( $N = 10$ ). Seven reported prior lathe experience, two reported no prior lathe experience, and one left the question blank. This mix allowed feedback from both experienced users (realism expectations) and novices (interpretability of cues).

3) *Measures*: Participants rated the following statements: (1) The simulation was realistic; (2) The materials were differentiable; (3) Force changes communicate tool engagement; (4) I could tell when I was doing something “wrong” based on feel alone; (5) The visual and haptic cues were synchronized; (6) The simulator helped me understand lathe cutting dynamics. Participants also reported whether they felt more confident operating a lathe after use.

4) *Results*: Table VI summarizes the Likert responses. Overall scores were high across all dimensions, with particularly strong ratings for visual–haptic synchronization and understanding of cutting dynamics. All participants reported increased confidence operating a lathe after using the simulator (10/10 “Yes”).

Novice users tended to rate tool engagement and learning impact slightly higher, while experienced users provided more critical realism scores, consistent with expert calibration to

TABLE VI  
USER STUDY LIKERT RESPONSES (1–10).

Statement	Mean	SD	Range
Simulation was realistic	8.9	0.9	8–10
Materials differentiable	9.0	1.1	7–10
Force indicates engagement	9.0	1.6	6–10
Could detect “wrong” by feel	8.4	1.6	5–10
Visual–haptic synchronized	9.9	0.3	9–10
Helped understand cutting dynamics	9.8	0.6	8–10

TABLE VII  
USER STUDY MEANS BY EXPERIENCE GROUP (LIKERT 1–10).

Statement	Experienced (n=7)	Novice (n=2)
Simulation was realistic	8.6	9.5
Materials differentiable	9.1	9.0
Force indicates engagement	9.1	10.0
Could detect “wrong” by feel	8.1	8.5
Visual–haptic synchronized	9.9	9.5
Helped understand cutting dynamics	9.7	10.0

real machine feel. Although the novice sample is small, this split suggests the simulator is interpretable for first-time learners while remaining recognizable to users with prior lathe exposure.

5) *Qualitative Feedback:* Open-ended comments highlighted (i) perceived realism of resistance changes with depth of cut (including experienced-user agreement that it “matched the feel”), and (ii) the usefulness of haptic cues for identifying aggressive cutting without relying on visuals. One participant noted that the suction-cup mounting reduced immersion, suggesting improved fixturing (e.g., clamps or a heavier base) for future iterations.

6) *Study Limitations:* This evaluation was conducted in an uncontrolled demo setting with a small sample size, so results should be interpreted as formative. Participants had different exposure times and explored different subsets of operations. A controlled follow-on study could standardize tasks (e.g., facing to a target diameter under fixed spindle speed), compare conditions (haptics on/off), and measure objective performance (time-to-completion, overshoot, surface error, and number of “crash” events).

#### D. Discussion

The results suggest that the simulator successfully conveys key aspects of manual lathe operation through haptic feedback, particularly in communicating tool engagement, material differences, and unsafe cutting conditions. The consistently high ratings for visual–haptic synchronization indicate that the tight coupling between force rendering and graphical feedback contributed to a coherent user experience.

Importantly, both experienced and novice users reported benefits, supporting the hypothesis that physically meaningful handwheel feedback can aid early skill acquisition while remaining recognizable to expert operators. While the small sample size limits statistical generalization, these findings

align with prior work emphasizing the role of haptics in motor-skill learning and motivate further controlled studies with larger participant groups.

#### VII. FUTURE WORK

- 1) **Two-Axis Implementation:** Add a second motor/hand-wheel for simultaneous X and Z control.
- 2) **Force Sensing:** Integrate torque sensors for closed-loop force control.
- 3) **Milling Mode:** The architecture extends naturally to milling simulation as a stretch goal.
- 4) **Extended User Study:** Conduct a formal study comparing training outcomes between haptic and non-haptic simulation.
- 5) **Audio Feedback:** Add cutting sounds synchronized with spindle speed and material for multi-modal feedback.

#### VIII. CONCLUSION

We have presented the design, implementation, and initial evaluation of a haptic lathe training simulator. The system successfully renders material-dependent cutting forces through a directly-driven handwheel interface, enabling users to experience realistic resistance during virtual machining operations.

Key contributions include: (1) A three-tier architecture separating visualization, communication, and real-time control; (2) A damping-based force rendering algorithm that produces realistic cutting feel; (3) Material differentiation through adjustable wall stiffness and yield force parameters; (4) Safety features including crash detection and emergency stop functionality.

The system demonstrates that effective haptic machining training can be achieved with modest hardware (total cost under \$60) while providing meaningful tactile feedback for skill development.

#### ACKNOWLEDGMENTS

AI systems used: Claude Opus 4.5 was used to convert existing codebase documentation and project notes into a well formatted LaTeX final report. The prompt instructed the system to read the repository and the provided background document, produce detailed implementation sections describing the software and hardware (including equations and concept explanations), and insert explicit placeholders wherever required information was missing rather than inventing results. This use primarily improved report structure, formatting quality, and clarity of code and hardware descriptions, while keeping technical content constrained to what was already present in the codebase and project files.

Google Gemini was also selectively used to fix grammatical errors in this report and smooth out sentence structures. The AI was prompted “proof read, dont change tone and content, make list of changes and I will implement them”.

## REFERENCES

- [1] X. He and Y. Chen, "A Haptic Virtual Turning Operation System," in *Proc. IEEE Int. Conf. on Mechatronics and Automation*, Luoyang, China, 2006, pp. 435–440.
- [2] S. B. Choudhury, S. K. Saha, and A. K. Mukherjee, "A Virtual Machining Simulator for Training and Education," *IEEE Trans. Learning Technologies*, vol. 12, no. 3, pp. 386–398, 2019.
- [3] S. Mishra, S. K. Saha, and A. K. Mukherjee, "Development of a virtual machining environment with haptic feedback," *Procedia Manufacturing*, vol. 5, pp. 538–549, 2016.
- [4] A. D. Silva, J. Pereira, and M. A. Otaduy, "A virtual reality lathe simulator with visuo-haptic feedback for machining training," *Virtual Reality*, vol. 27, no. 2, pp. 1121–1136, 2023.
- [5] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. A. Srinivasan, "Haptics in minimally invasive surgical simulation and training," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 56–64, 2004.
- [6] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 24–32, 2004.
- [7] A. Akshay, R. Raghavan, and S. K. Saha, "Significance of Haptic Feedback in Learning Lathe Operating Skills," in *Proc. IEEE Tenth Int. Conf. on Technology for Education (T4E)*, Chennai, India, 2018, pp. 112–119.
- [8] J. K. Salisbury, K. J. Kuchenbecker, and the Stanford Haptics Group, "The Hapkit: A low-cost haptic device for education and research," Stanford University. [Online]. Available: <https://hapkit.stanford.edu>
- [9] Turnin' and Burnin' Haptic Lathe Project, Lafayette College. [Online]. Available: <https://sites.lafayette.edu/turnin-and-burnin/>

## APPENDIX: FILE DESCRIPTIONS

The following files are included for reproduction of this project:

- 1) `run_integrated_system.py`: Main launcher script. **To start the system**, run `python run_integrated_system.py` from the project root. This starts the Python bridge and opens the Processing GUI.
- 2) `integrated_lathe_controller.py`: Python bridge that manages TCP communication between the GUI and the Raspberry Pi Pico microcontroller. Computes haptic forces and relays commands.
- 3) `pico_upload/main.py`: Pico boot script that initializes the motor controller and listens for serial commands. **Upload this file to the Pico** along with `motor_control.py`.
- 4) `pico_upload/motor_control.py`: MicroPython firmware implementing quadrature encoder reading, PID position/velocity control, and damping-based virtual wall haptic rendering.
- 5) `Haptic_Lathe_GUI/Haptic_Lathe_GUI_Integrated.pde`: Processing sketch for the virtual lathe GUI. Renders stock geometry, tool position, and material removal in real-time; sends force commands to the Python bridge via TCP.
- 6) `lathe_config.ini`: Configuration file specifying serial port names and system parameters.
- 7) `INTEGRATED_SYSTEM_README.md`: Comprehensive setup and usage guide for the integrated system.