Resource-Efficient FPGA-based Machine Learning Control for Active Structural Damping in Shock Environments

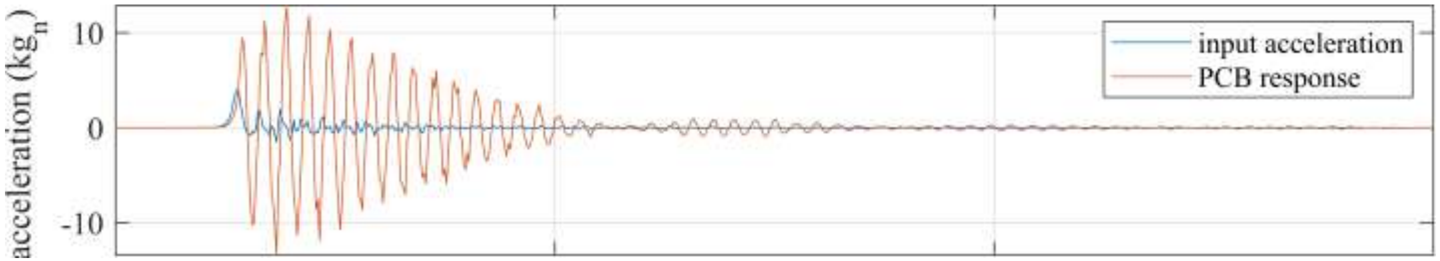
Trotter Roberts[1], Joud Satme[1], Puja Chowdhury[1], Austin R.J. Downey[1,2], and Jason Bakos[3]

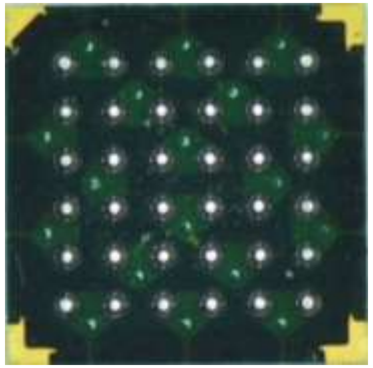
1. Department of Mechanical Engineering, University of South Carolina, Columbia, SC 29208
2. Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC 29208
3. Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208

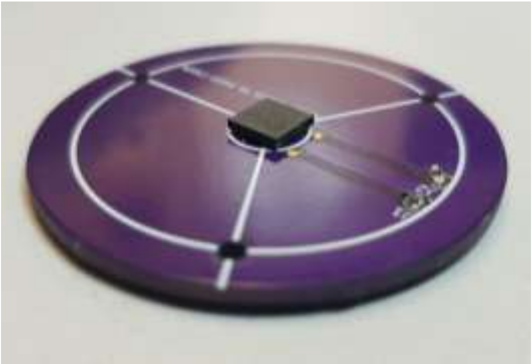
UNIVERSITY OF SOUTH CAROLINA

Outline

- Introduction
    - High-rate Challenge?
    - What is an FPGA?
    - Project Motivation and Support
- Ongoing Work
    - Focused Study
    - Simulation
    - Governing Equations
    - Set Parameters
- Control Strategies
- FPGA Implementation
- Results and Analysis
- Conclusion
- Future Work



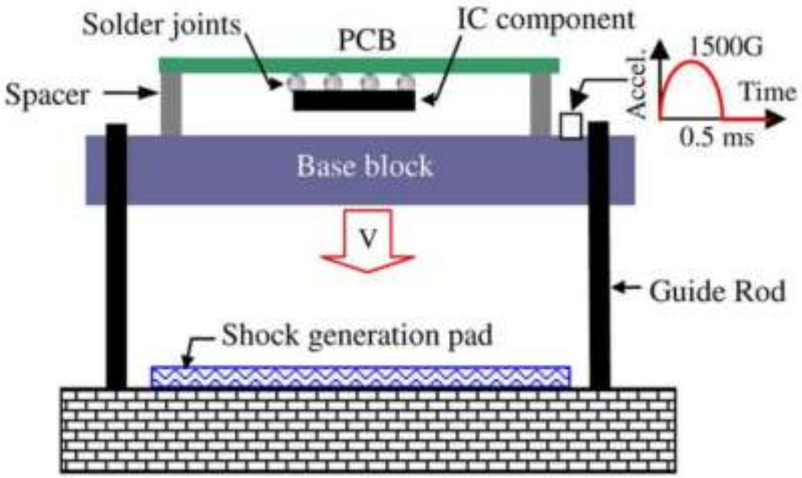




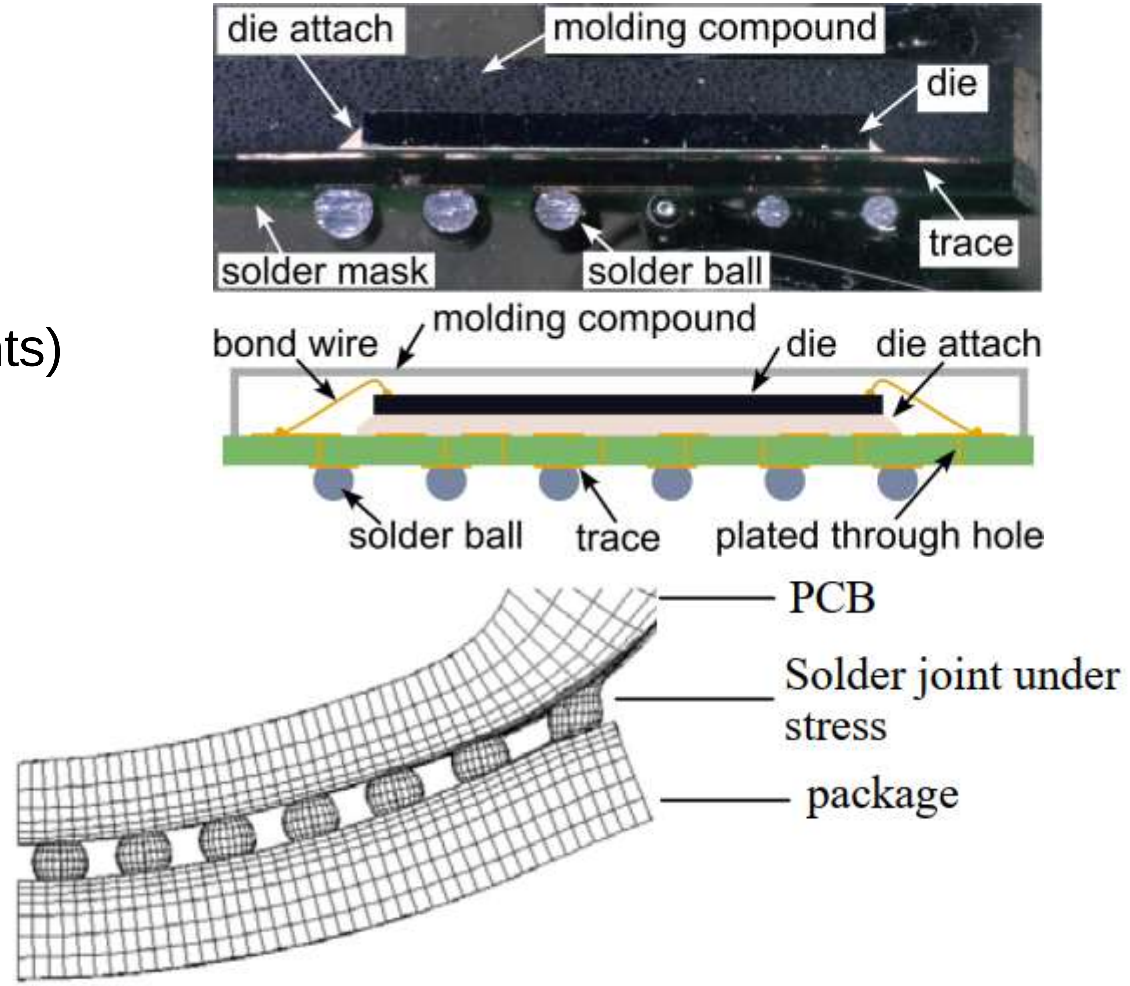
PCB failure mechanisms under shock

PCB failures under shock are caused by:

- Bending of the base PCB board, causing stresses to build up at the solder balls.
- Adhesion challenges of masses (components) accelerating away from the PCB.

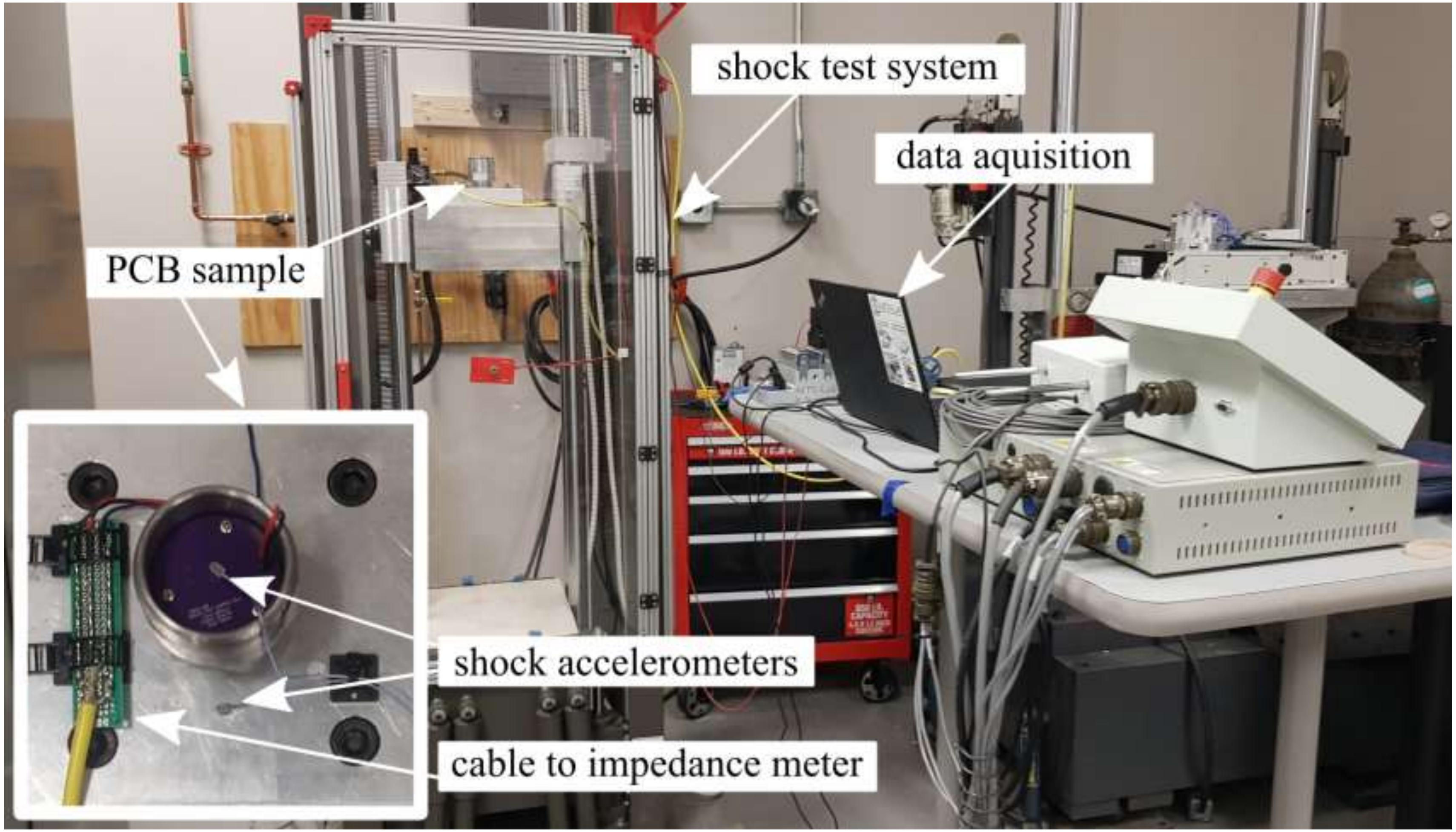


Wong, E. H., Yiu-Wing Mai, and Matthew Woo. "Analytical solution for the damped-dynamics of printed circuit board and applied to study the effects of distorted half-sine support excitation." IEEE Transactions on advanced packaging 32.2 (2009): 536-545.

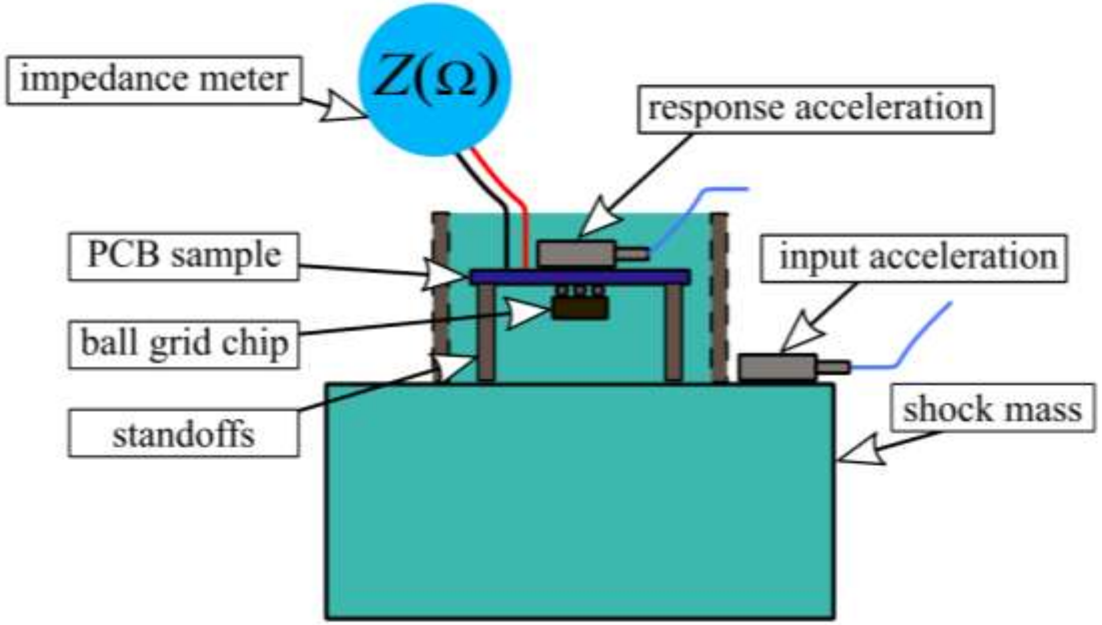


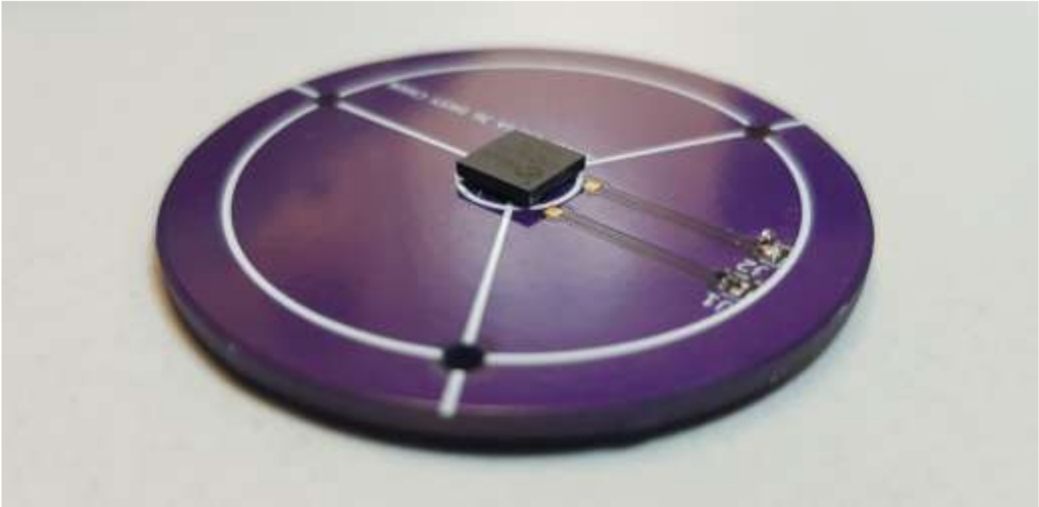
Seah, S. K. W., Wong, E. H., Ranjan, R., Lim, C. T., and Mai, Y. W., 2005, "Understanding and testing for drop impact failure," ASME Pacific Rim Technical Conference and Exhibition on Integration and Packaging of MEMS, NEMS, and Electronic Systems, pp. 1089-1094.

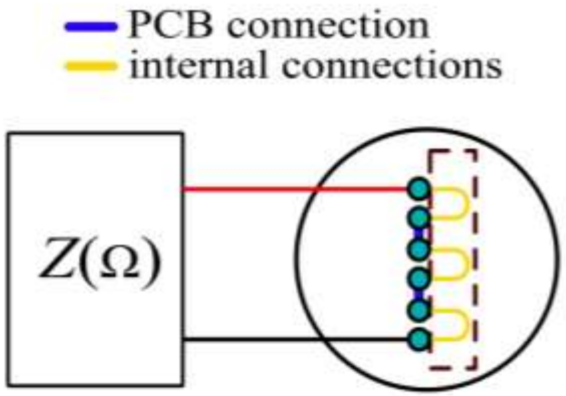
Ongoing work

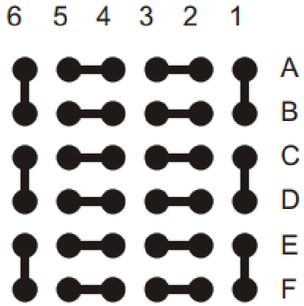


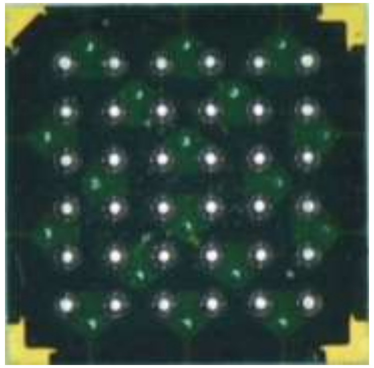
Relevant Experimental system

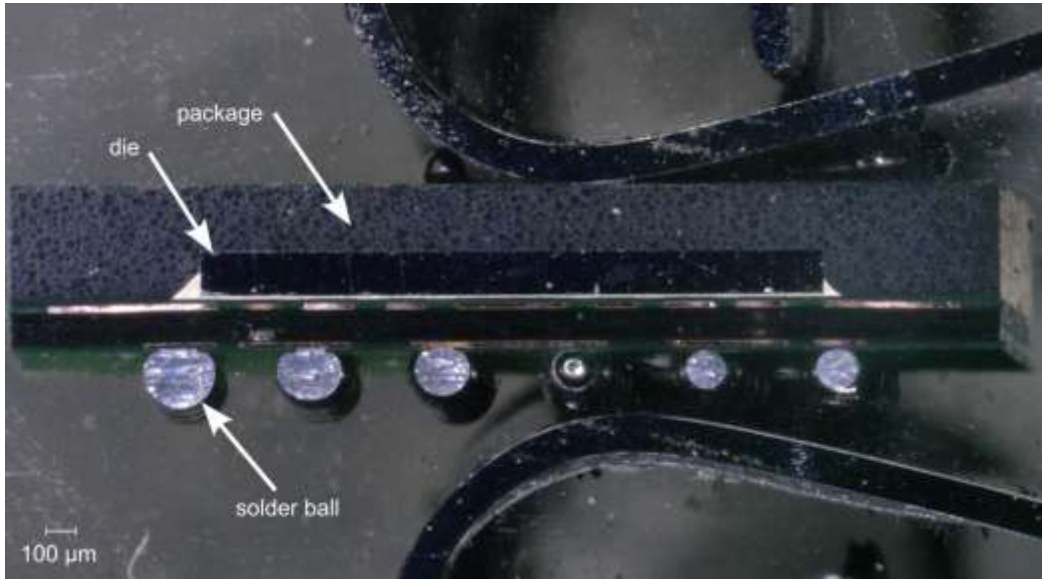




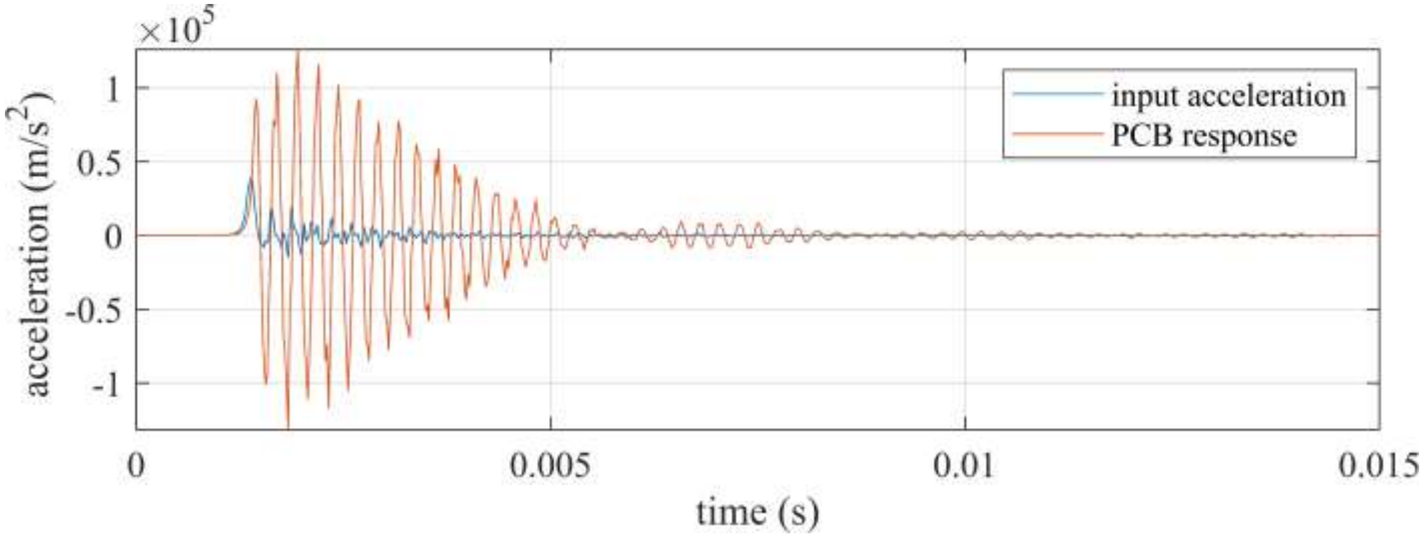


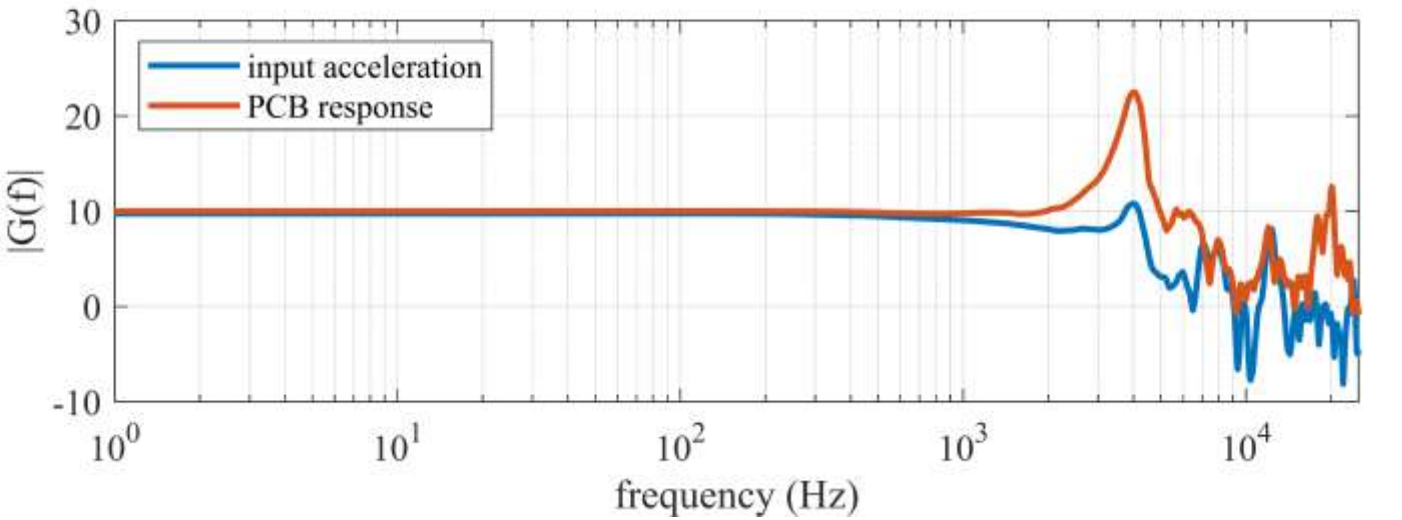






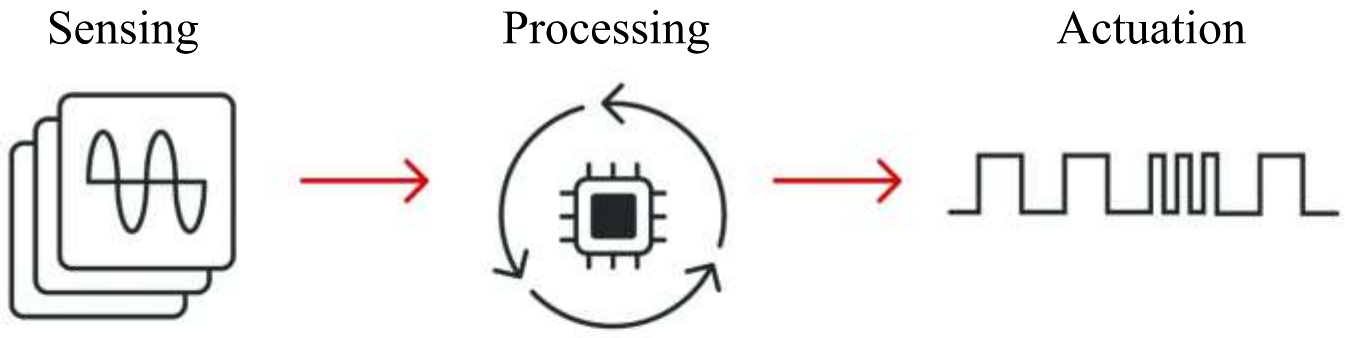
Experimental procedure

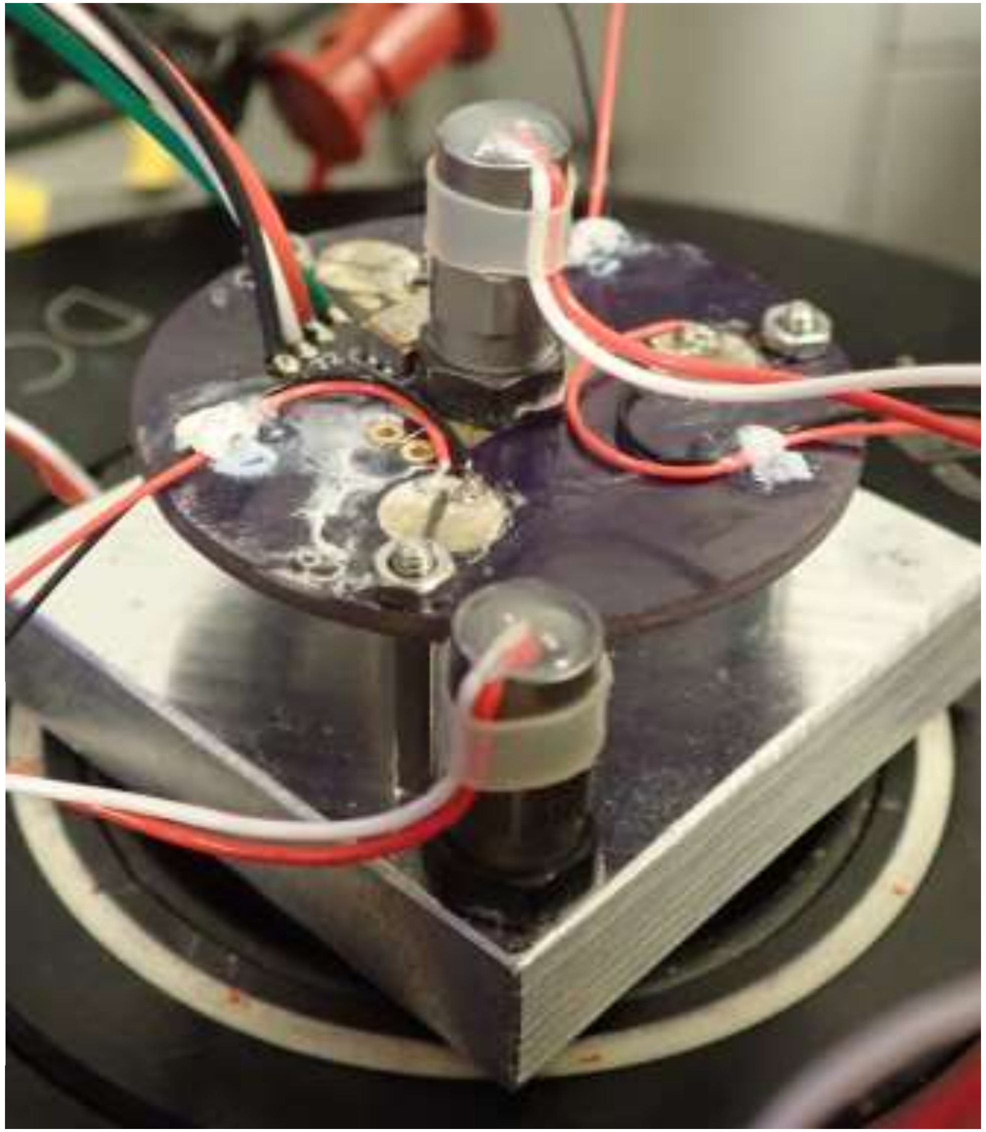




Project Goal

- Develop data-driven real-time control solutions for systems under shock.
- Extend electronic component lifetime in harsh environments.

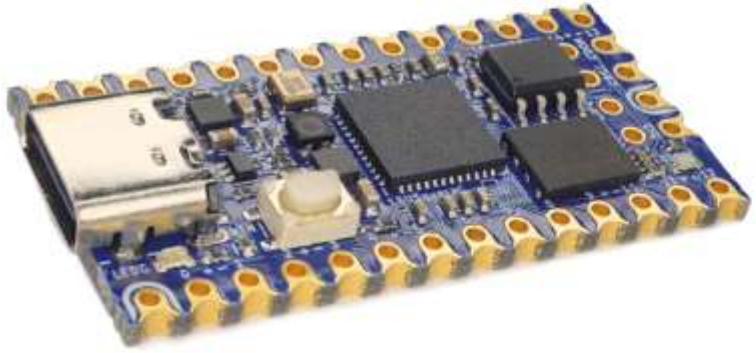




What is an FPGA?

Field Programable Gate Array

- A Field-Programmable Gate Array (FPGA) is a reconfigurable hardware device that allows for parallel processing, making it ideal for real-time control applications.
- Low-latency processing for rapid control adjustments.
- Parallel computation enables efficient execution of control algorithms.
- Custom hardware acceleration for ML and structured controllers.



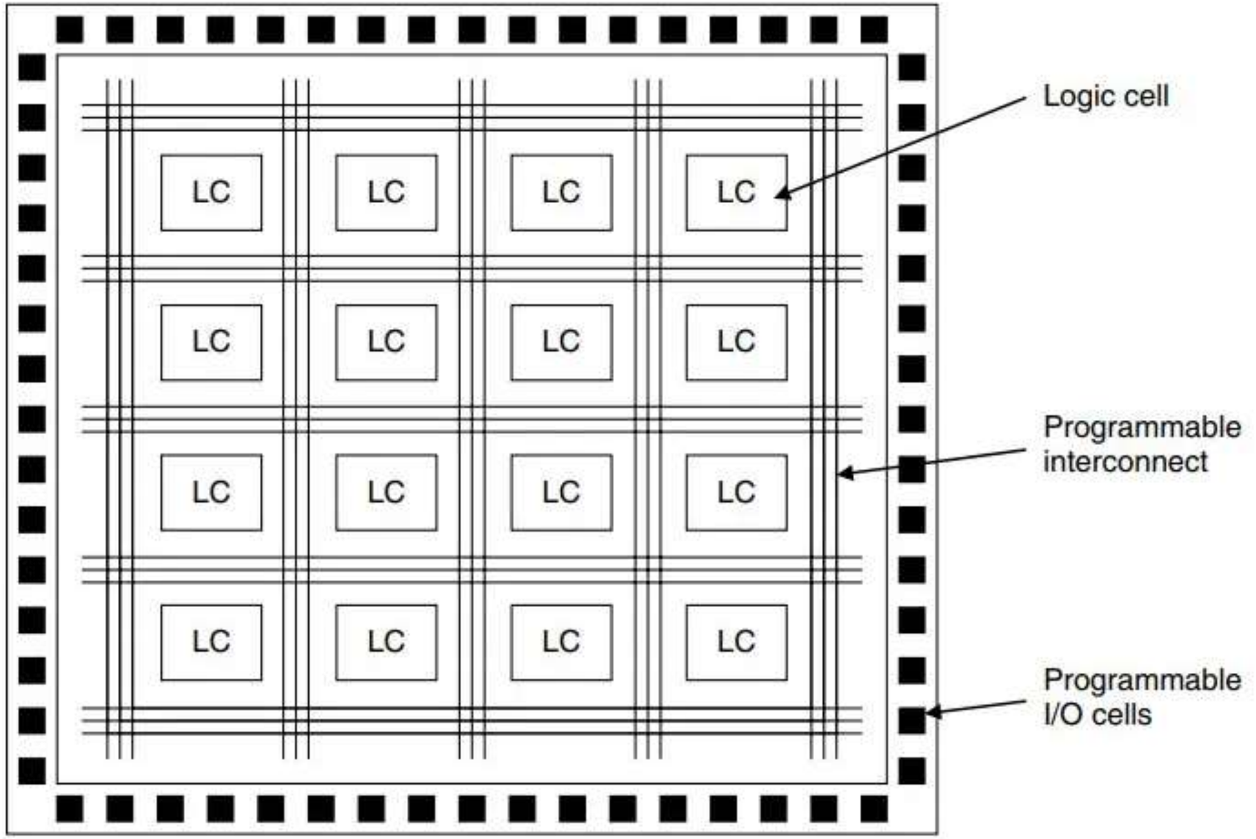
ICE40

Kintex-7 K70T-2C

U55C UltraScale+

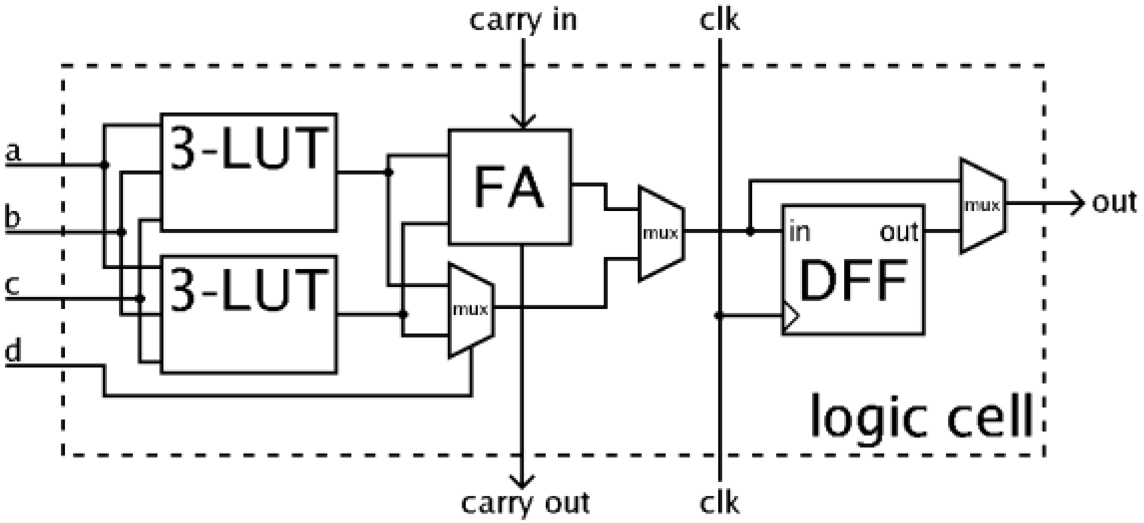
FPGA Components

- Typically, an FPGA consists of three basic components:
  - Programmable Logic Cells/Blocks (also called slices)
  - Programmable Routing
  - IO Blocks
- Logic Blocks are responsible for implementing the core logic functions.
- Routing is responsible for connecting the Logic Blocks.
- IO Blocks are connected to the Logic Blocks through the routing and help to make external connections.



What is Inside an FPGA?

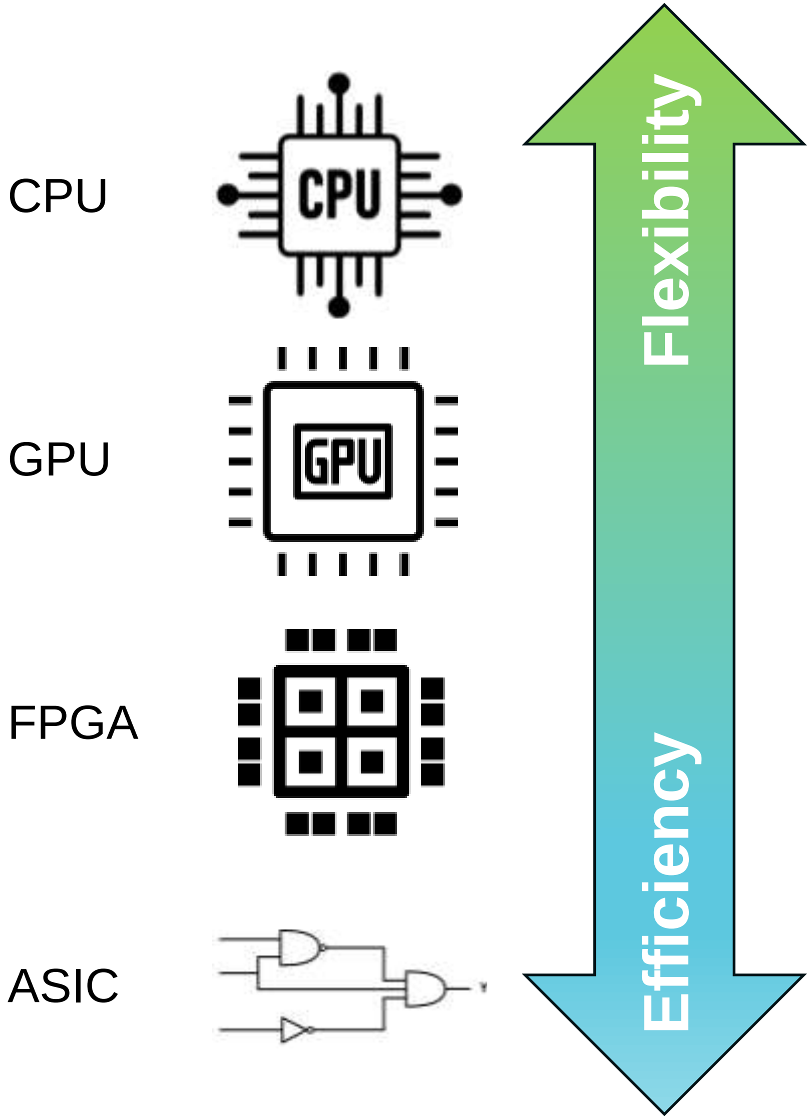
- What is a CLB?
  - A Configurable Logic Block (CLB) is a programmable circuit inside an FPGA.
  - Think of it like LEGO bricks for digital circuits—each CLB can be configured for different tasks.
- What's Inside a CLB?
  - Look-Up Tables (LUTs): Store logic functions.
  - Flip-Flops (FFs): Hold values for sequential operations.
  - Carry Chains: Optimize arithmetic operations.
  - Multiplexers (MUX): Decide which signals to use.
- Why is CLB Hardware Important in Design?
  - Every logic operation, memory storage, and computation needs CLB resources.
  - More features = More CLBs = Higher FPGA cost & power consumption.



Petter.kallstrom, Public domain, via Wikimedia Commons

FPGA vs GPU vs CPU

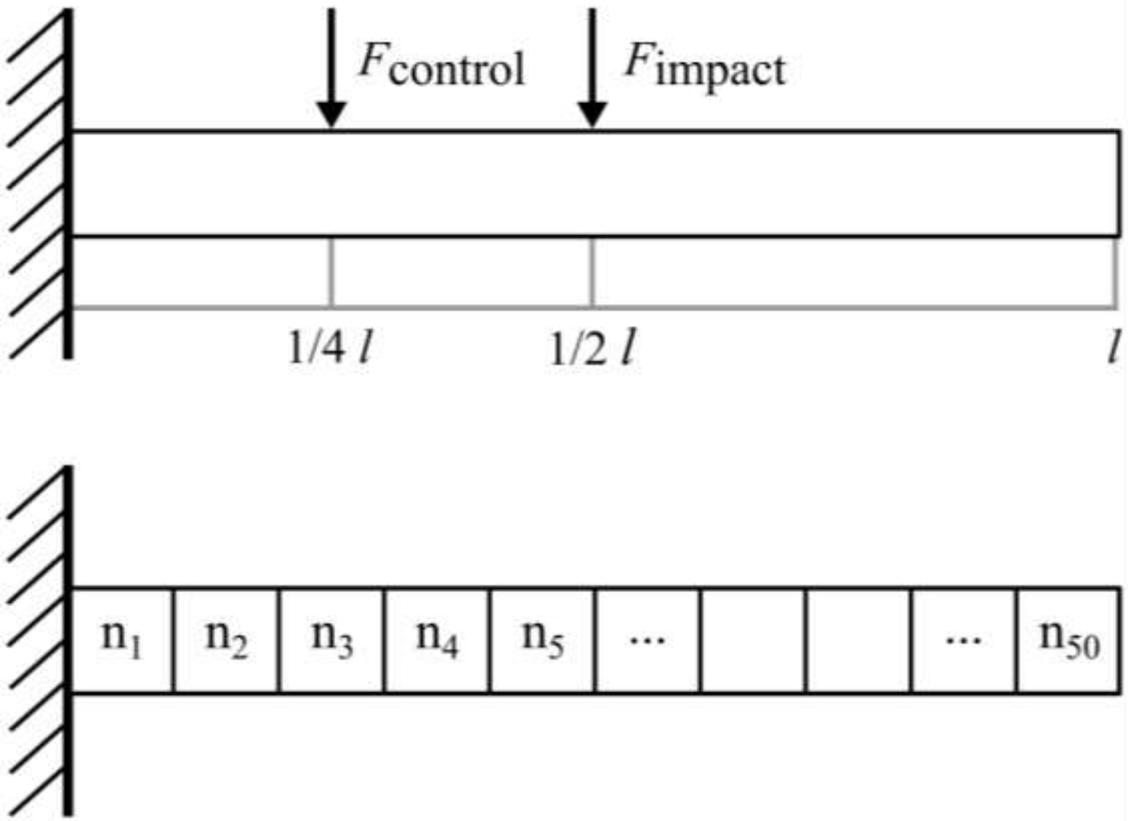
- **CPUs** (Central Processing Units)
  - Highly flexible and easy to program
  - General-purpose, but lower computational efficiency for specialized tasks
- **GPUs** (Graphics Processing Units)
  - Optimized for parallel computing, great for machine learning and graphics
  - More efficient than CPUs for certain workloads, but still power-intensive
- **FPGAs** (Field-Programmable Gate Arrays)
  - Custom hardware acceleration for specific applications
  - More energy-efficient than GPUs and CPUs for dedicated tasks
  - Harder to program, requiring expertise in hardware description languages
- **ASICs** (Application-Specific Integrated Circuits)
  - Highest efficiency but fixed-function and costly to develop
- **Key Insight:** FPGAs balance efficiency and flexibility, making them ideal for edge computing, embedded systems, and real-time applications with power and space constraints.



System Simulation

System Under Test

- Material:
  - Based on slender steel beam.
  - Width = 0.05 m; Thickness = 0.005 m; Length = 1.00 m
  - Young's Modulus = 210e9 Pa; Density = 7850 kg/m^3
  - α = 0.01; β = 0.001
- Simulation:
  - Nodes (n) = 50; DOFs = 100
  - Impact node = ½ n; Control node = ¼ n
  - Impact Force = 1000 N
  - β_n = 0.25; γ = 0.5



Simulation Model

- FEM of a cantilever beam subjected to an impact force.
- Governed by the Euler-Bernoulli Beam Theory:

$$M\ddot{u}(\mathrm{t}) + C\dot{u}(t) + Ku(t) = F(t)$$

- where M is the mass matrix, C is the Rayleigh damping matrix, K is the stiffness matrix, $u(t)$ is the displacement vector, and $F(t)$ is the external force vector.
- Discretized into n nodes; resulting in $2n$ DOFs, due to transverse displacement and rotation at each node.
- External forces are applied at a specific node during the impact; control forces are superimposed.
- Elemental mass and stiffness matrices are assembled into global matrices.

Simulation Model

- The Rayleigh damping matrix C is constructed as a linear combination of the mass and stiffness matrices:

$$C = \alpha M + \beta K$$

where α and β are user-defined damping coefficients.

- Time integration via Newmark-beta method, updating displacements, velocities, and accelerations iteratively.

- Effective Stiffness matrix equation:

$$K_{\text{eff}} = \mathrm{K} + \frac{\gamma}{\beta_n \Delta t} C + \frac{1}{\beta_n \Delta t^2} M$$

where β_n and γ are Newmark-beta parameters.

Control Strategies

Control Strategies

- Control forces are added to the system's force vector to influence the beam dynamics.
- Controllers studied:
  - Proportional-Derivative (PD)
  - Linear Quadratic Gaussian (LQG)
  - Multi-Layer Perceptron (MLP)
- Focus kept to simpler strategies for the time to confirm feasibility.
- Hardware implementation required to guarantee resource (not simulated).

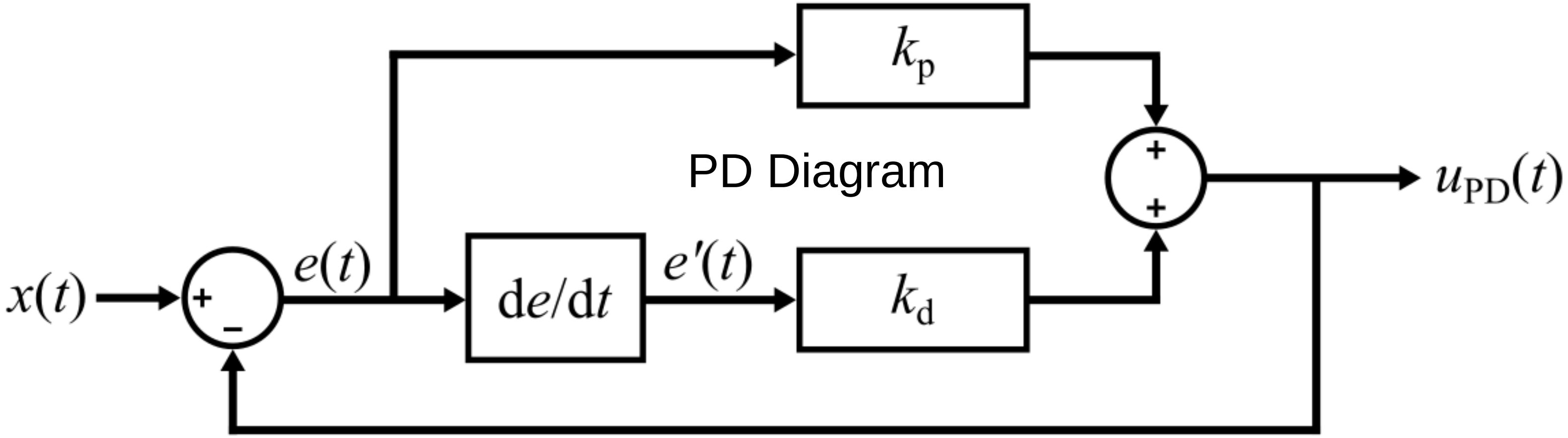
Proportional-Derivative

- The PD Controller applies feedback based on the displacement (e) and velocity ($\dot{e}$) errors:

$$u(t) = k_p e + k_d \dot{e}$$

where k_p is the proportional gain and k_d is the derivative gain.

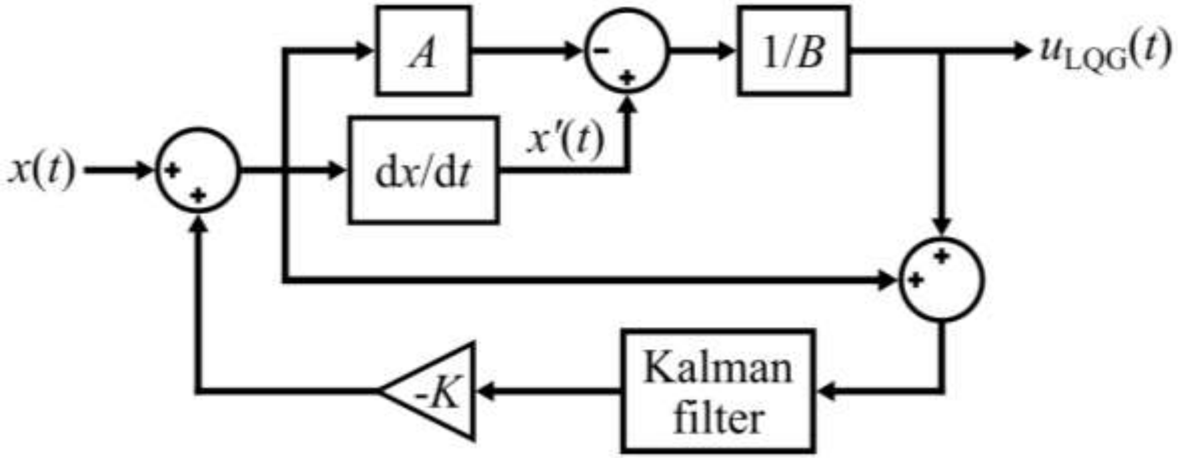
- Effective but linear and limited to its tuning parameters.



PD Diagram

Linear Quadratic Gaussian

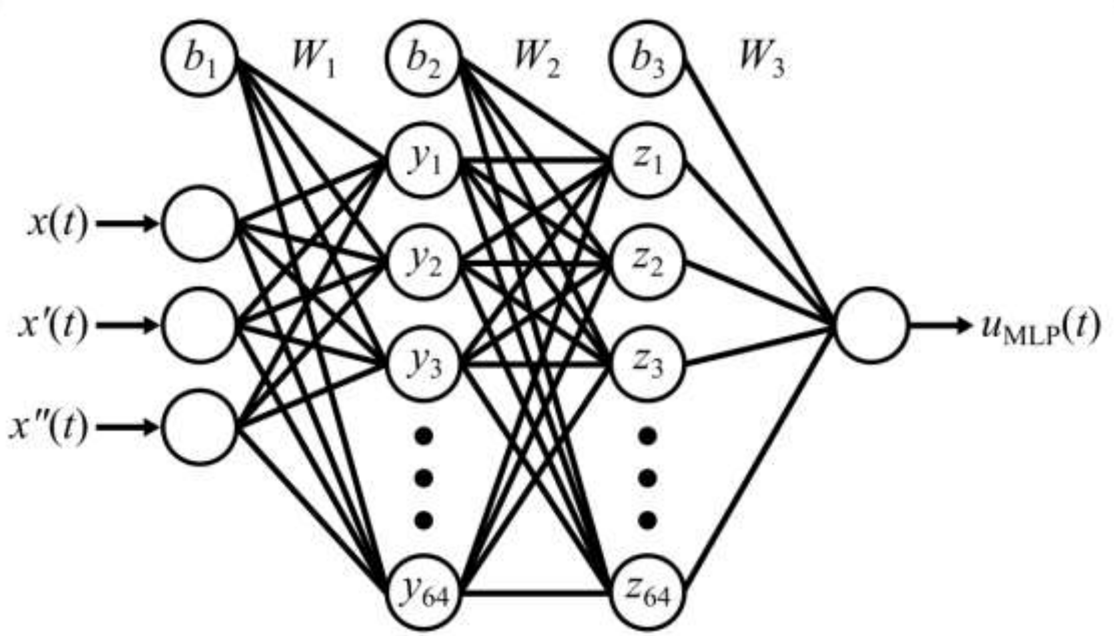
- The LQG Controller combines state feedback with an observer (Kalman filter) for optimal performance in the presence of noise.
- The augmented state-space model includes both displacement and velocity states.
- Feedback is applied to minimize the combined cost of state deviations and control effort.
- State-space representation: $\dot{x}(t) = Ax(t) + Bu(t)$
- Cost function: $J = \int_0^\infty (x^T\, Qx + u^T Ru)dt$
- Control law: $u(t) = -Kx(t)$
- Optimized but sensitive to noise and modeling errors.



Multi-Layer Perceptron Layout

The MLP Controller is an artificial neural network made up of:

- an input layer with three neurons representing displacement (x), velocity ($\dot{x}$), and acceleration ($\ddot{x}$);
- two hidden layers with up to neurons each, activated by the ReLU function $\sigma(x) = \max(0, x)$;
- an output layer with one neuron producing the predicted control force (F_{MLP}).

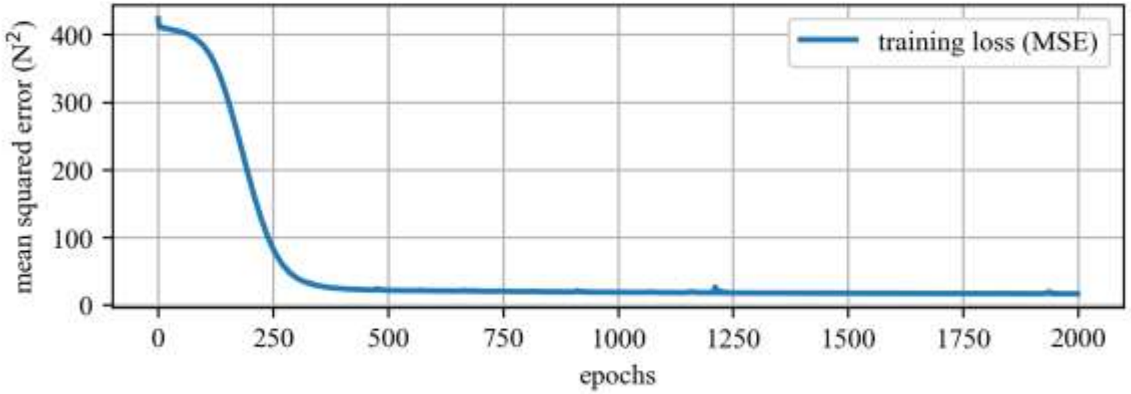


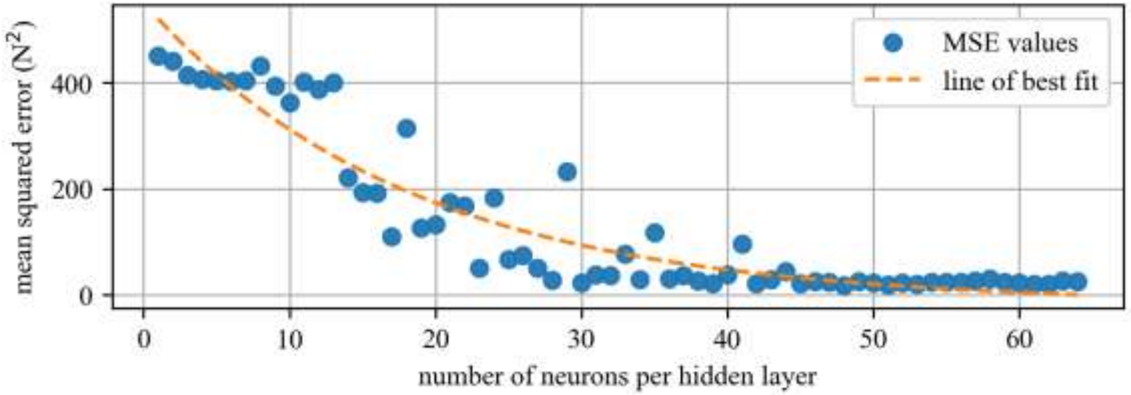
Training Multi-Layer Perceptron

- The MLP is trained using real LQG data, with the control force computed by a LQG controller as the reference.
- The training minimizes the error between F_{MLP} and F_{LQG} using the Mean Squared Error (MSE) loss and optimizes weights and biases through backpropagation using the Adam optimizer.

$$L = \frac{1}{N}\sum_{i=1}^{N}(F_{MLP} - F_{LQG})^2$$

- A network of 30 neurons per hidden layer was chosen.

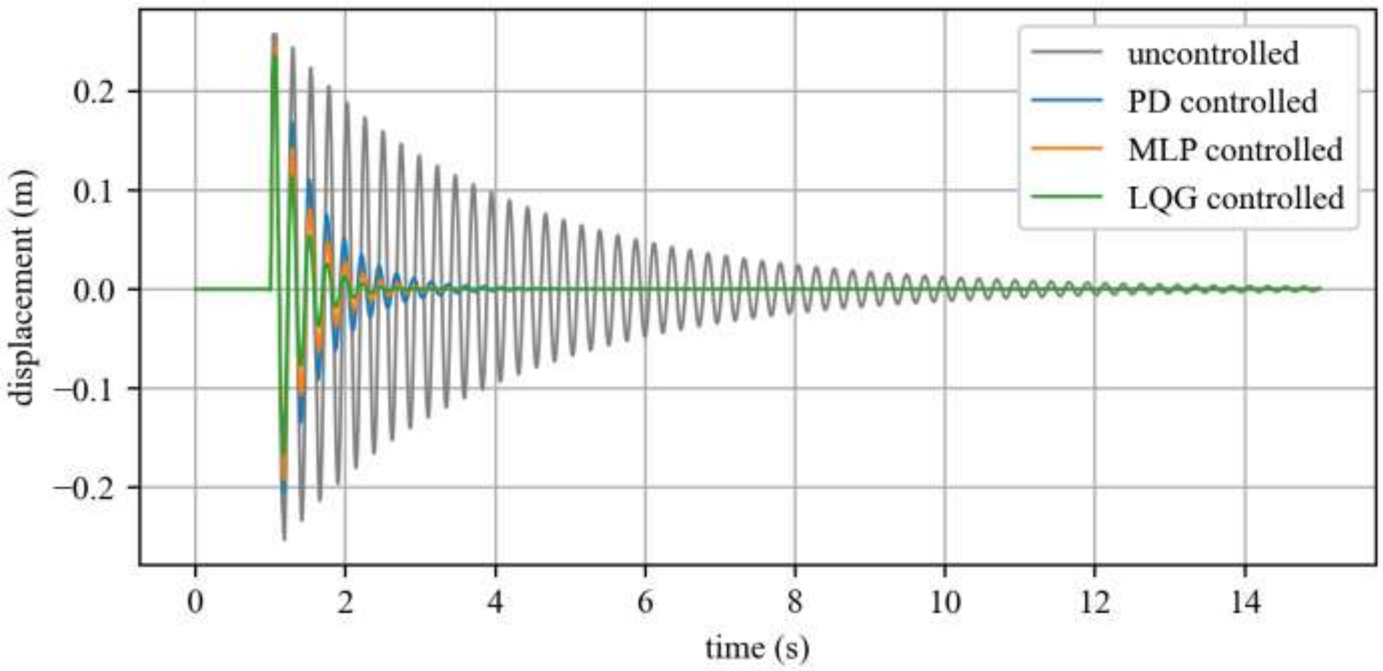




Results

Controller Results

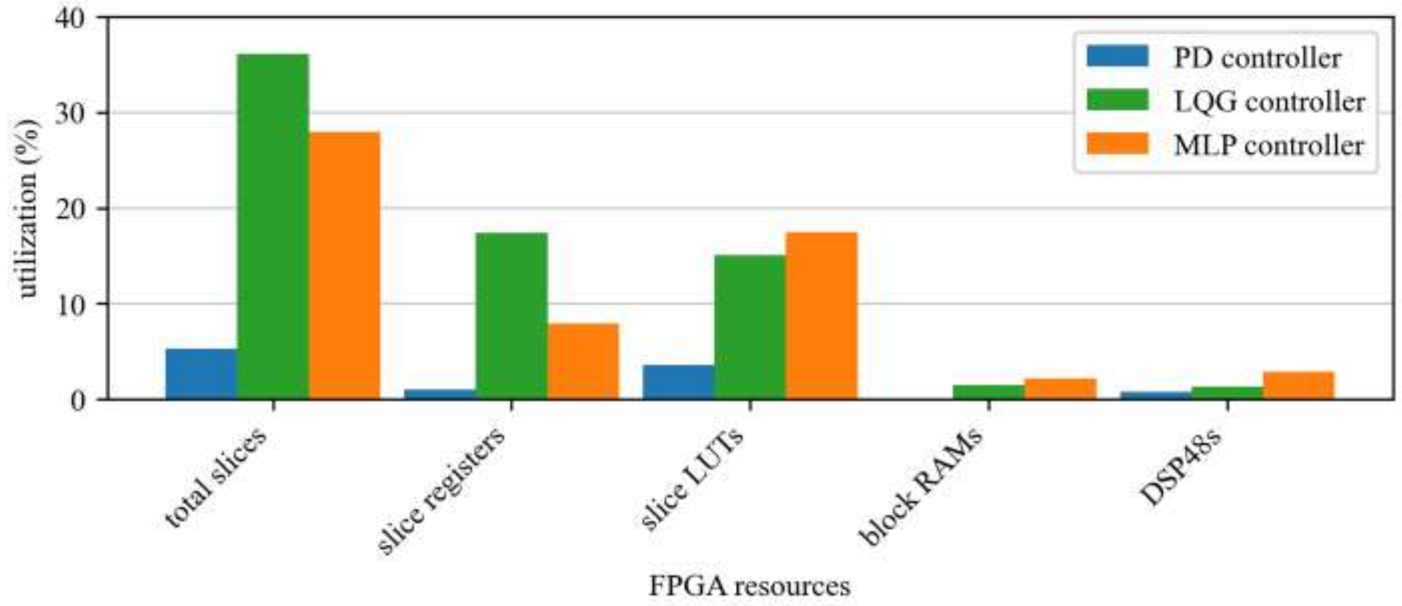
- All control methods do fairly well.
- The MLP was able to learn the majority of the LQR's response.
- In all metrics, the LQR outperforms the MLP which outperforms the PD.



metric	uncontorlled	PD controlled	LQG controlled	MLP controlled
max displacement	0.2575 m	0.2435 m	0.2343 m	0.2451 m
RMS displacement	0.0591 m	0.0266 m	0.0204 m	0.0234 m
settling time	8.74 s	1.82 s	1.01 s	1.36 s
control effort		0.91 N	0.53 N	0.70 N
damping efficiency		3.57%	7.32%	5.09%
time-weighted damping efficiency		1.27%	3.64%	2.25%

FPGA Utilization Results

- MLP has a 22% improvement over the LQG in terms of total slices used.
    - This is the metric that matters the most as its what typically constrains FPGA deployment
- MLP uses more LUTs, this is expected as the MLP has more weights to store



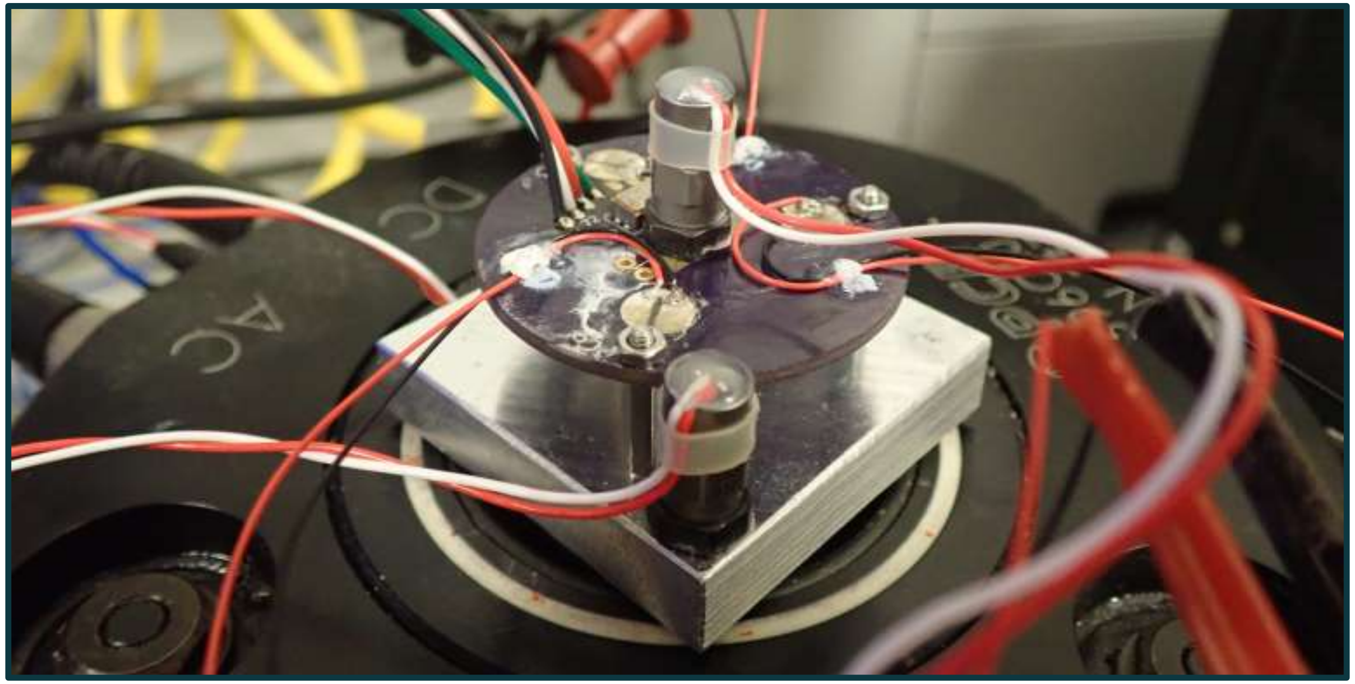
resource	PD controller	LQG controller	MLP controller
total slices	53% (539/10250)	36.1% (3699/10250)	28.0% (2867/10250)
slice registers	1.0% (842/82000)	17.4% (14259/82000)	7.9% (6439/82000)
slice LUTs	3.6% (1482/41000)	15.1% (6208/41000)	17.5% (7165/41000)
block RAMs	0.0% (0/135)	1.5% (2/135)	2.2% (3/135)
DSP48s	0.8% (2/240)	1.3% (3/240)	2.9% (7/240)

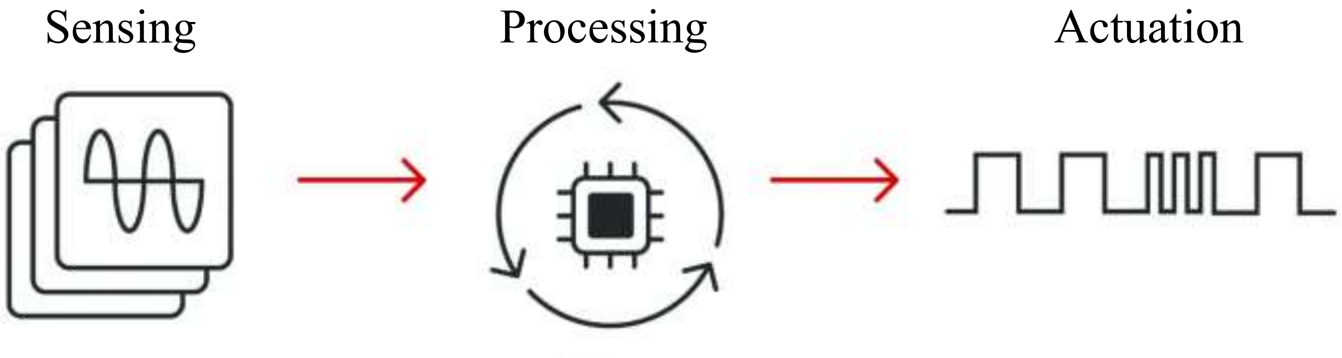
Conclusion

- Control Trade-offs:
  - PD excels in simplicity
  - LQG balances effort and performance
  - MLP offers efficiency with optimization potential.
- MLP Viability:
  - With targeted optimizations, MLP can match or surpass traditional controllers in FPGA deployment.
- Structured vs. Learning-Based Control:
  - Predictable, model-based approaches ensure stability, while learning-based methods enable adaptability.
- Optimization Potential:
  - Techniques like pruning and quantization can enhance MLP’s real-time feasibility.

Future Work

- Improvement of simulation models.
- Optimization of learning-based control strategies.
- System analysis.
- Dataset collection.
- Real-time FPGA experimentation.
- Piezoelectric sensing/actuation experimentation.



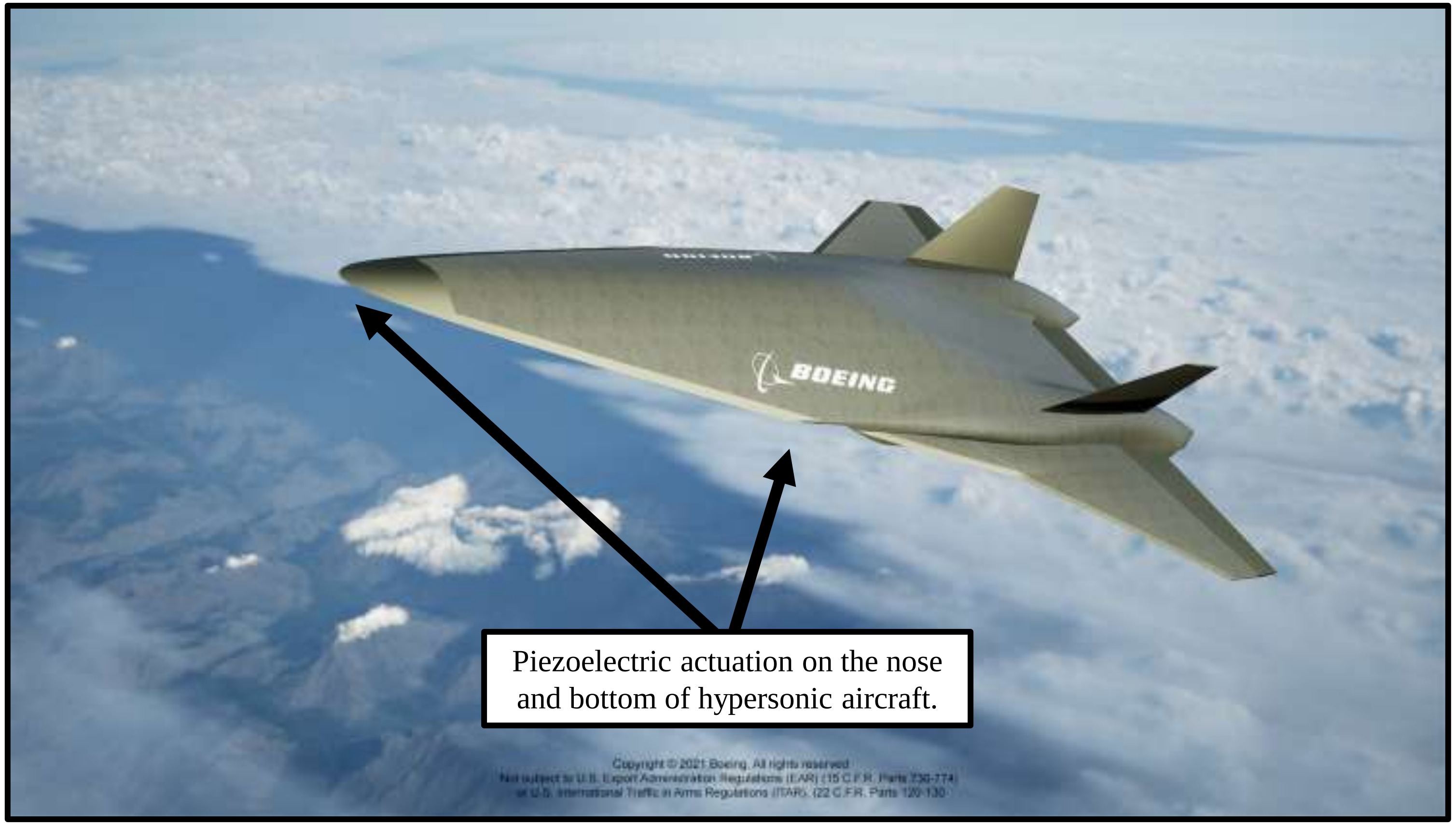


Questions?

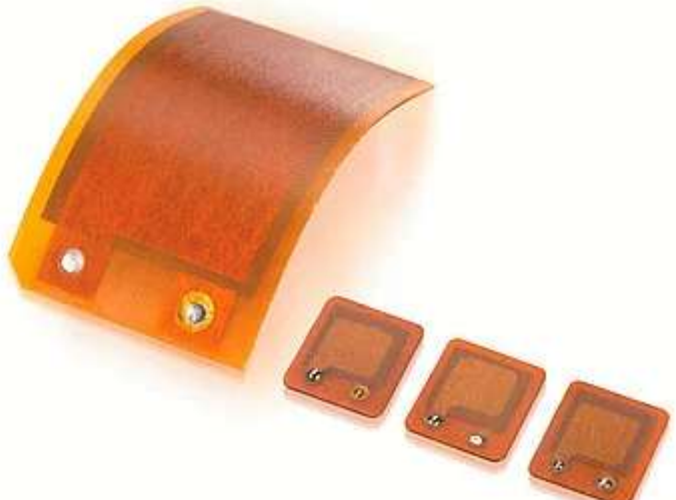
This material is based upon work supported by the National Science Foundation grant numbers 1937535, 1956071, 2234921, and 2237696, with additional support from the Air Force Office of Scientific Research (AFOSR) through award no. FA9550-21-1-0083. Any opinions, findings conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, or the United States Air Force.

Trotter Roberts 03/18/2025

Real-world Applications

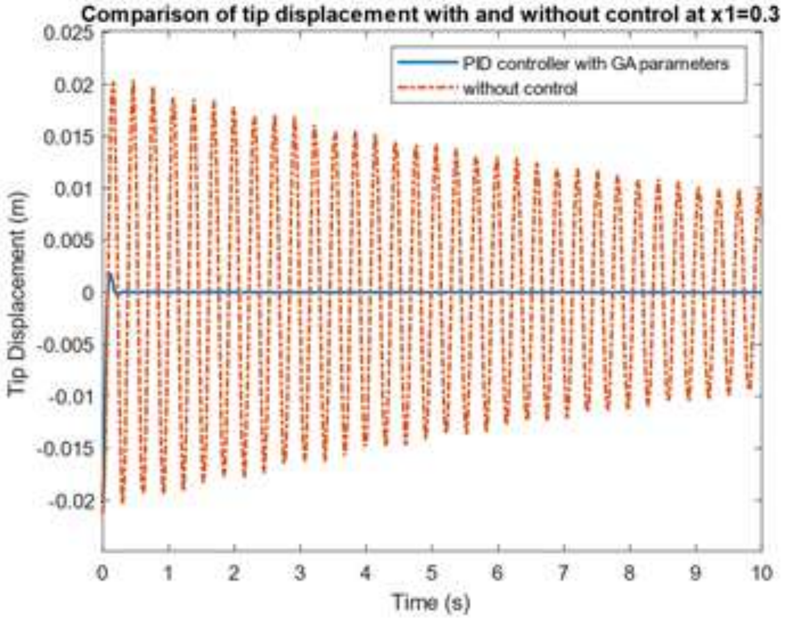


Piezoelectric actuation on the nose and bottom of hypersonic aircraft.

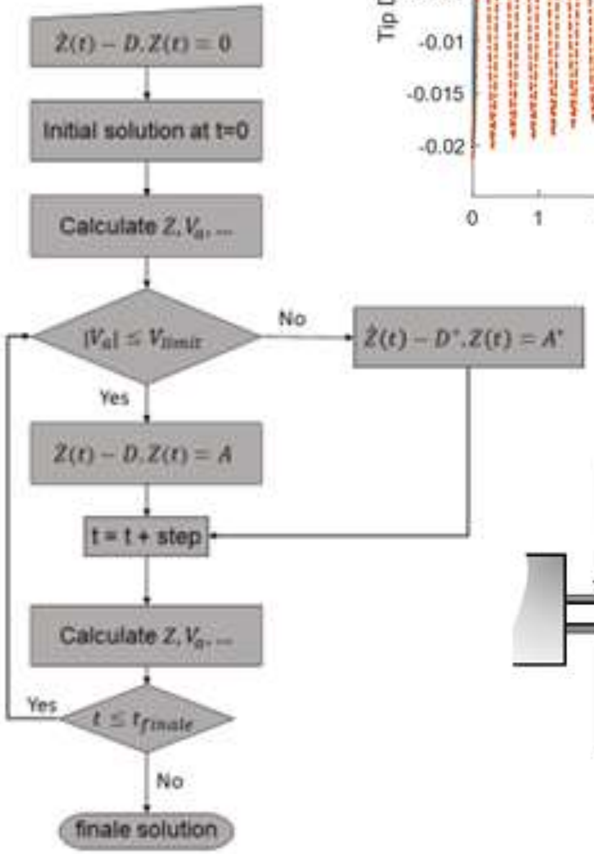


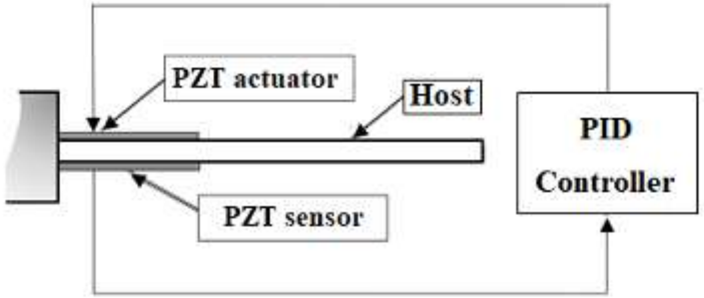
PZT Actuator Patches

Active Control



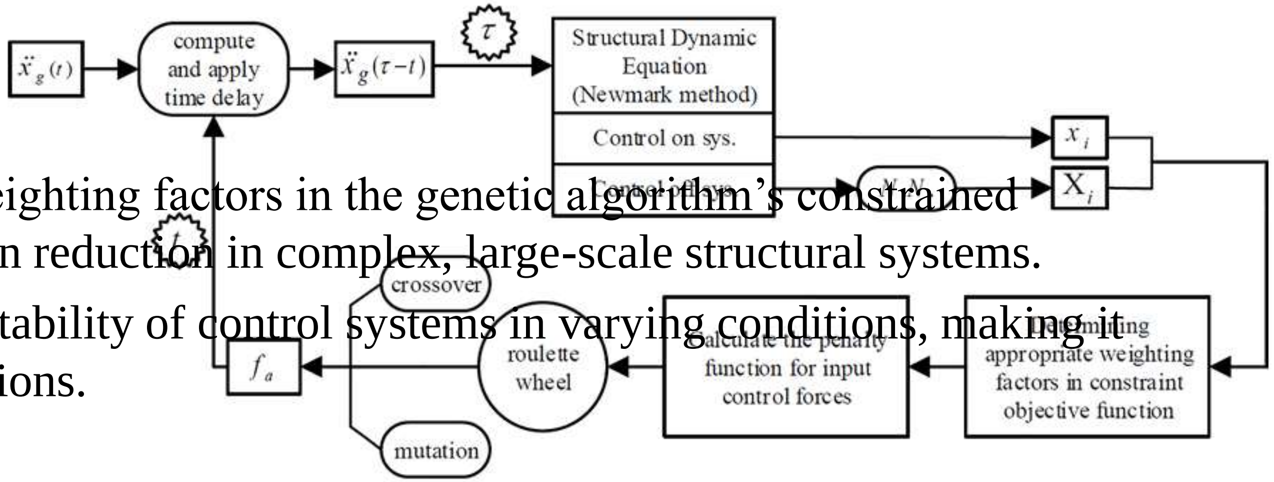
- Key Point: Optimized control of cantilever
- Content:
  - **Study**: Awada et al. (2022)
  - **Conclusion**: The genetic algorithm developed i smart cantilever beam using piezoelectric actua n vibrations.
  - **Takeaway**: A simple PID controller demonstrat ficient vibration control in smart structures. This means that simple, but be used to improve vibration control.





Control Strategies

- <u>Key Point</u>: Improved performance in structural control through adaptive algorithms.
- <u>Content</u>:
  - **Study**: Banaei et al. (2023)
  - **Conclusion**: The introduction of dynamic weighting factors in the genetic algorithm's constrained objective function leads to improved vibration reduction in complex, large-scale structural systems.
  - **Takeaway**: This approach enhances the adaptability of control systems in varying conditions, making it more suitable for complex structural applications.



FPGA Usage

- <u>Key Point</u>: FPGA-based vibration control enab damping.
- <u>Content</u>:
  - **Study**: Leva & Piroddi (2008)
  - **Conclusion**: Implementing an active vibration contr adaptive filtering, reducing vibration in high-precisi
  - **Takeaway**: FPGAs provide a powerful solution for real-time adaptability, which are crucial for systems monitoring.

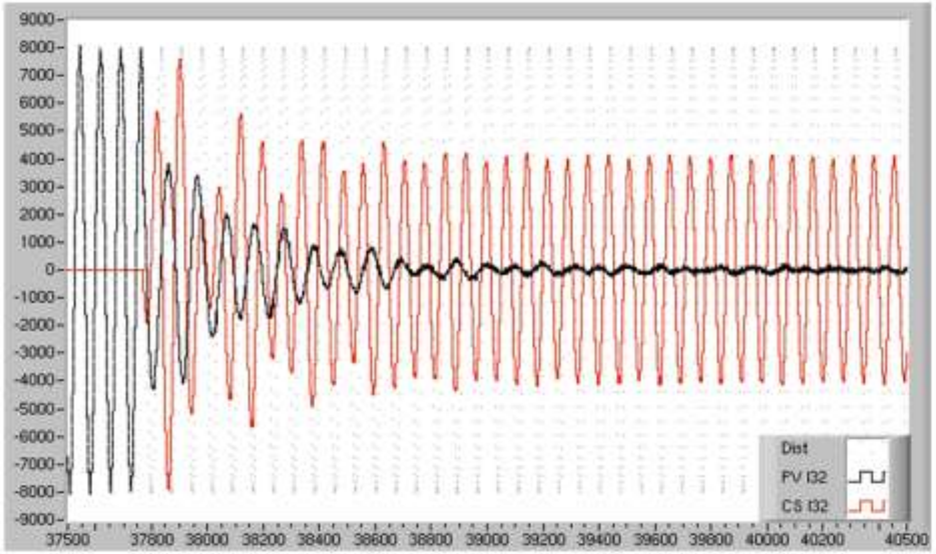


Fig. 6. Cancellation of a single frequency: controlled variable (PV I32, black), control signal (CS I32, red) and disturbance (Dist, gray).