

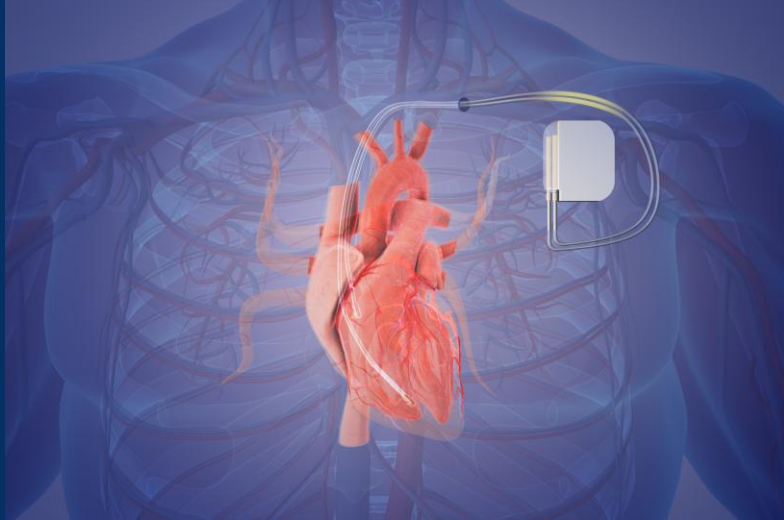
MATLAB EXPO

Embedded Intelligence: The Future of AI in Engineering Design

Mary Ann Freeman, MathWorks

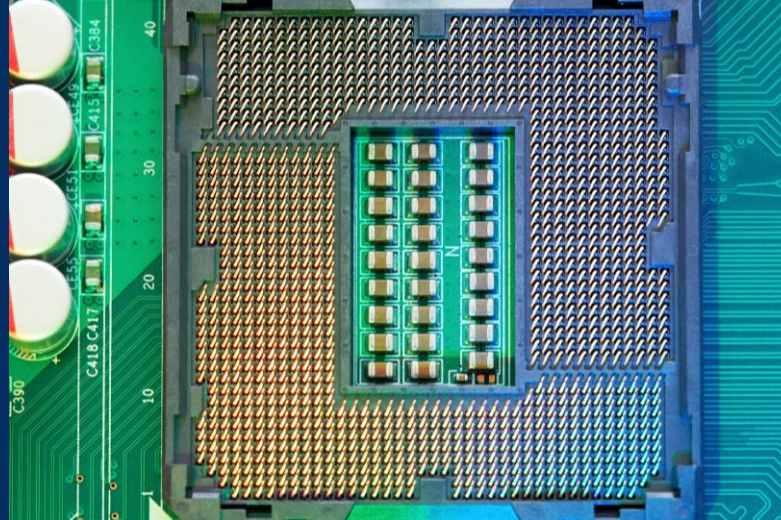


Combining AI and engineering



Medtronic

Detecting heart disorders in implantable cardiac monitors



Princeton University

Creating wireless chip designs



Eaton

Advancing solar power systems performance

AI can even write code

The screenshot displays the MATLAB R2023a software interface. The top ribbon includes tabs for HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a code editor window for a file named 'PlotCurrentTorque.m'. The code editor contains the following MATLAB code:

```
1 % Create a new figure
2 figure;
3
4 % Create axes for the current and torque
5 yyaxis left
6 plot(time, current, 'LineWidth', 1);
7 ylabel('Current (A)');
8 grid on
9 title('Motor Current and Load Torque');
10
11 yyaxis right
12 plot(time, -torque, 'LineWidth', 1);
13 ylabel('Torque (N*m)');
14 ylim([-3e-4 5e-5]);
15
16 % Add x-axis label
17 xlabel('Time (s)');
18
```

Overlaid on the code editor is a Copilot suggestion box. It contains the instruction: "Put both lines on the same chart, with torque on the left and current on the right". Below the instruction, it states "Generated code might be inaccurate. Validate before use." and provides buttons for "Accept", "Discard", and "View Diff". There are also thumbs up and thumbs down icons for feedback. A green checkmark is visible in the top right corner of the suggestion box.

At the bottom of the interface, the Command Window shows the prompt: ">> Press to generate code with Copilot".

Key themes

- The long-term relationship of AI and engineering
- What's possible today
- Prepare for an increasingly AI-enabled future

AI is mostly linear algebra

MATLAB loves linear algebra!

Training via Backpropagation:

$$W \leftarrow W - \eta(\Delta A^T)$$

```
% Backpropagation (training)
```

```
delta = Z - Y;
```

```
gradW = delta * A.;
```

```
W = W - eta * gradW;
```

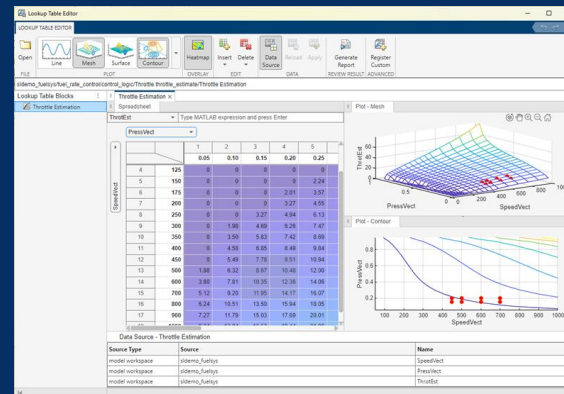
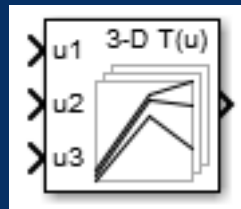
Inference:

$$Z = WA + b$$

```
% Forward pass (inference)
```

```
Z = W * A + b;
```

Simulink Loves Lookup Tables!



1990s

2016

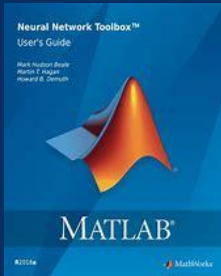
2017-2019

2020-2022

2023-2025

Toolboxes

- Neural Network Toolbox
- Statistics Toolbox



GPU



Toolboxes

- Deep Learning Toolbox
- Text Analytics Toolbox
- Reinforcement Learning
- Predictive Maintenance Toolbox

Code Generation

- GPU Coder
- MATLAB Coder

Apps

- Image Labeler
- Deep Network Designer
- Classification Learner
- Regression Learner
- Video Labeler
- Signal Labeler

Interoperability

- TensorFlow-Keras Importer
- ONNX Support

Compression

- Quantization

Code Generation

- Deep Learning HDL Coder

Apps

- Experiment Manager
- Lidar Labeler
- Reinforcement Learning Designer

Model-Based Design

- Image Classification & Model Prediction
- Recurrent Neural Networks
- Object Detectors

Interoperability

- TensorFlow Model Importer & Exporter
- PyTorch Importer

Compression

- Taylor, Projection Pruning

Code Generation

- TensorFlow Lite

Interoperability

- Co-execution blocks

Verification

- Out of distribution detection
- Robustness

Domain Specific AI

- Medical Imaging
- Automated Visual Inspection

Accessibility

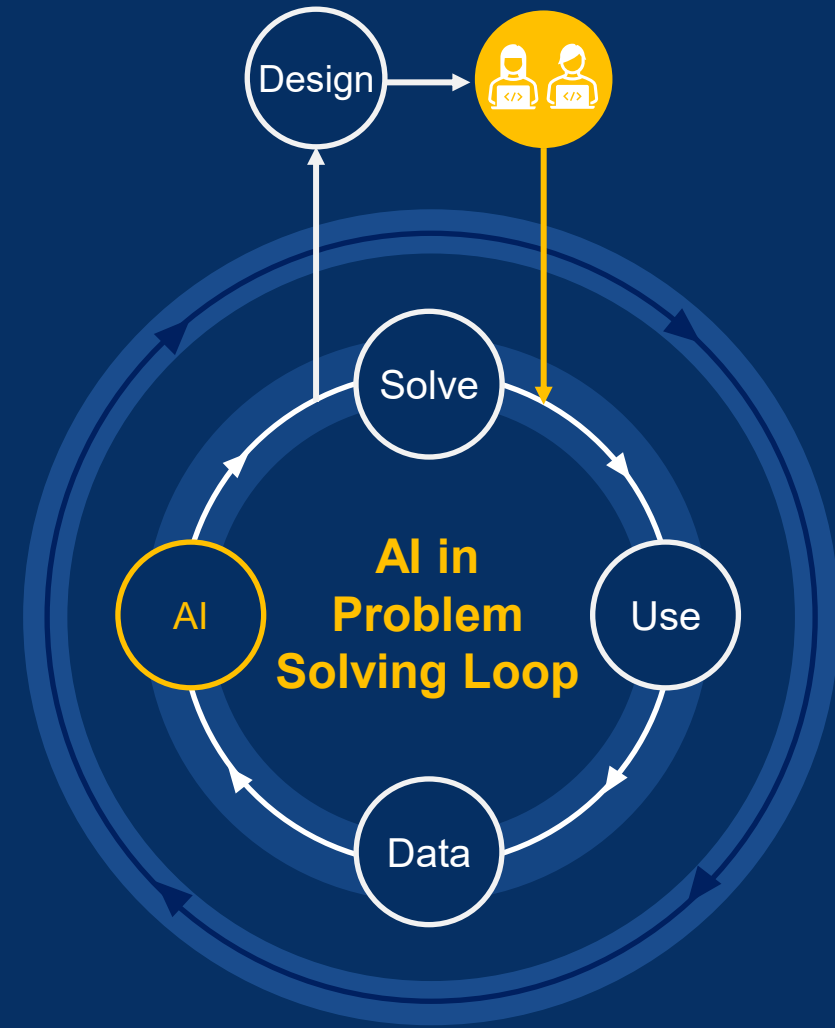
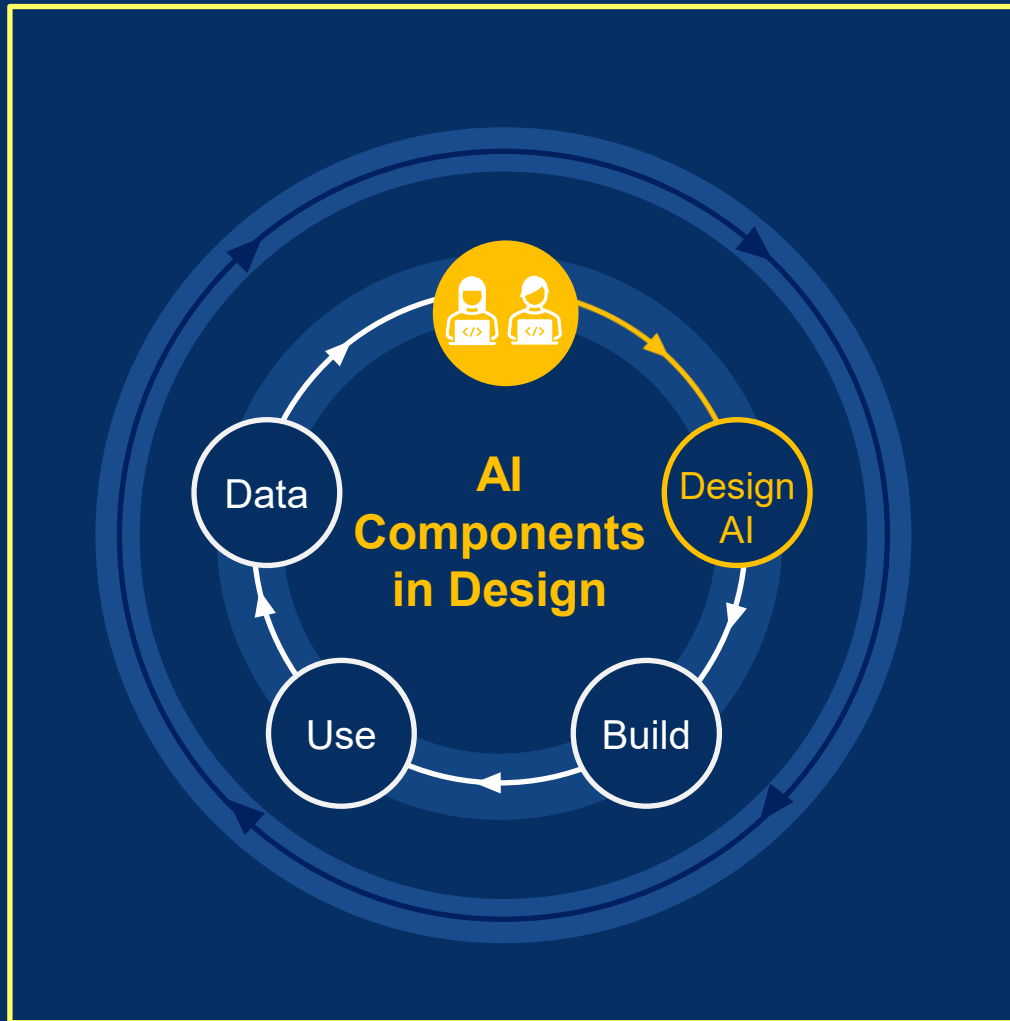
- Deep Learning Model Hub

Over 500+ examples



What is the **future of AI** in engineering design?

AI in engineering design loops

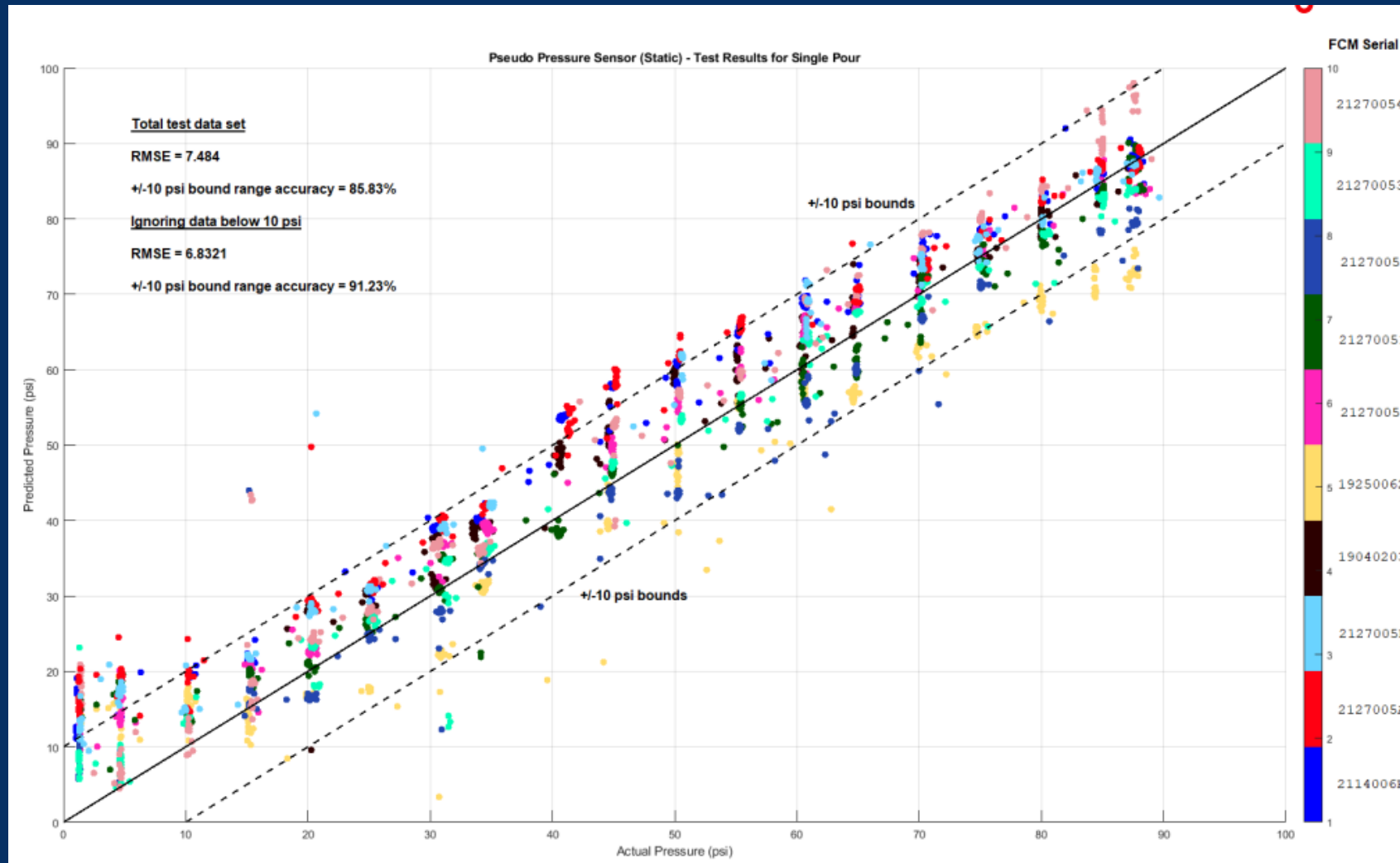


AI becomes part of the design



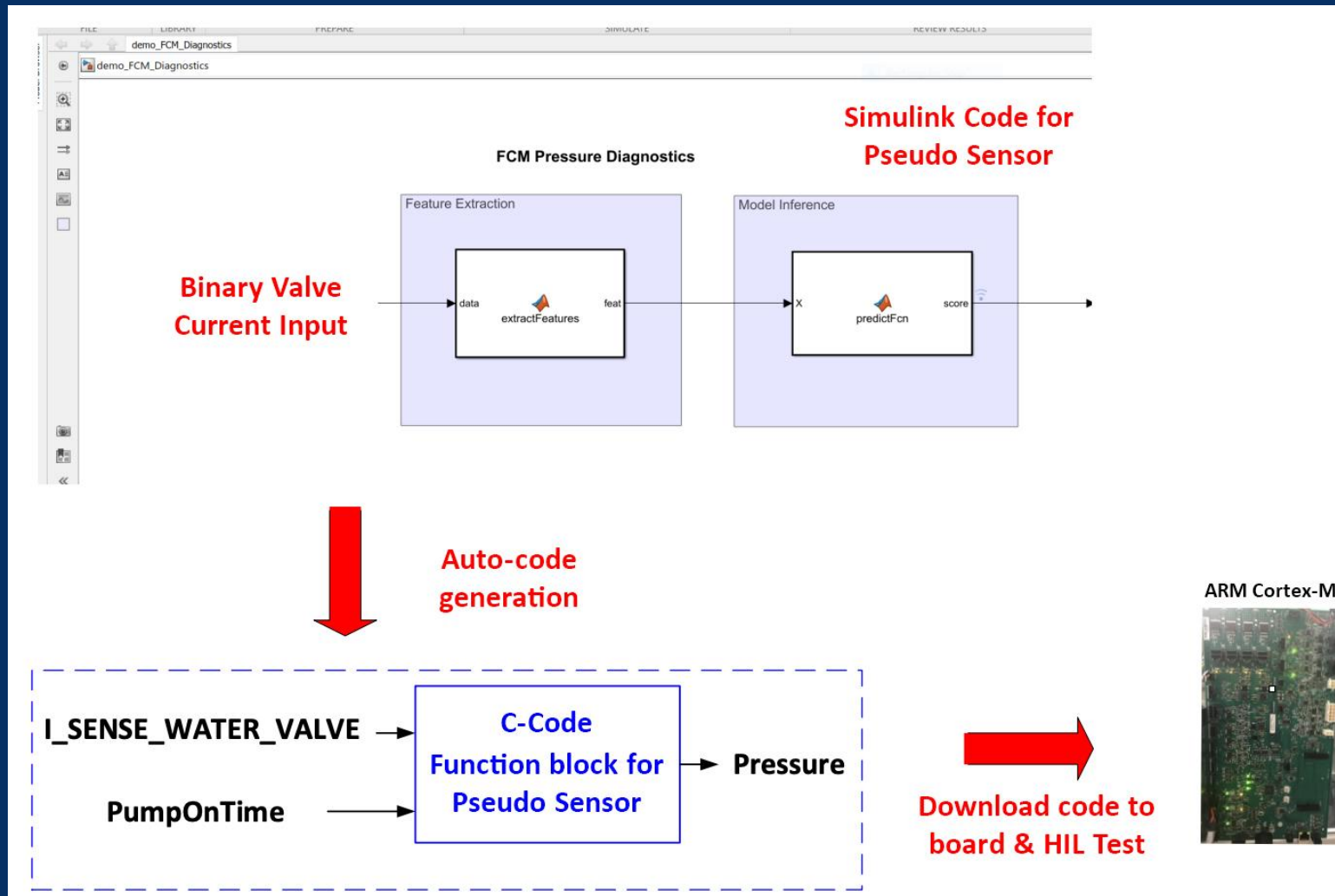
Source: <https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/company/events/conferences/matlab-expo-2023/ww-2023-expo-pseudo-pressure-sensor-to-improve-diagnostics-in-a-solenoid-actuated-valve.pdf>

Predicting water pressure with an AI virtual sensor



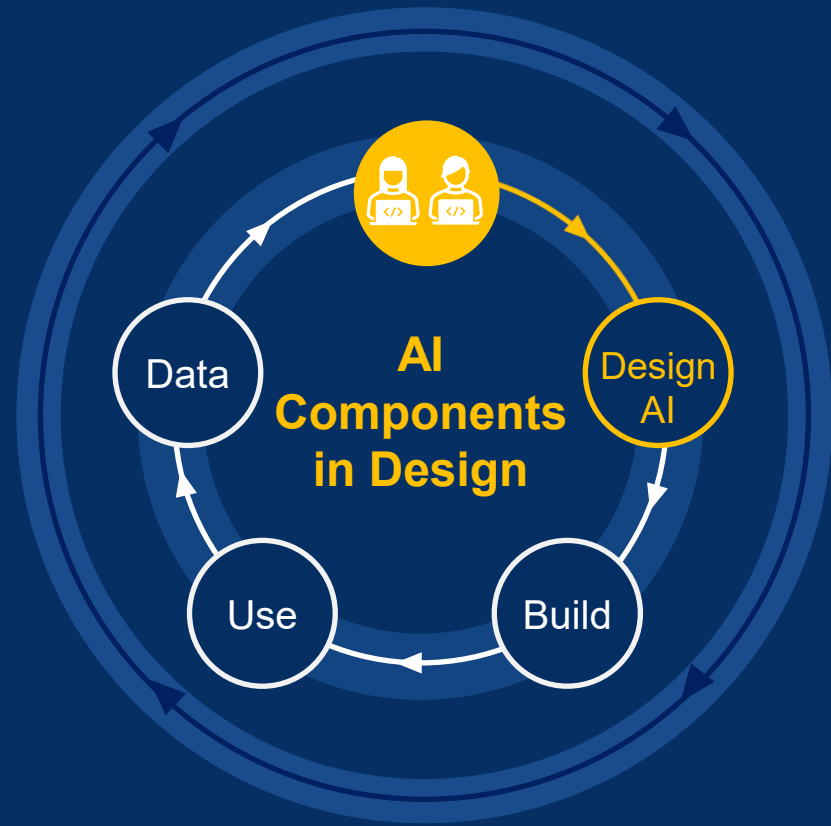
Source: <https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/company/events/conferences/matlab-expo-2023/ww-2023-expo-pseudo-pressure-sensor-to-improve-diagnostics-in-a-solenoid-actuated-valve.pdf>

Deploy the AI model to the edge



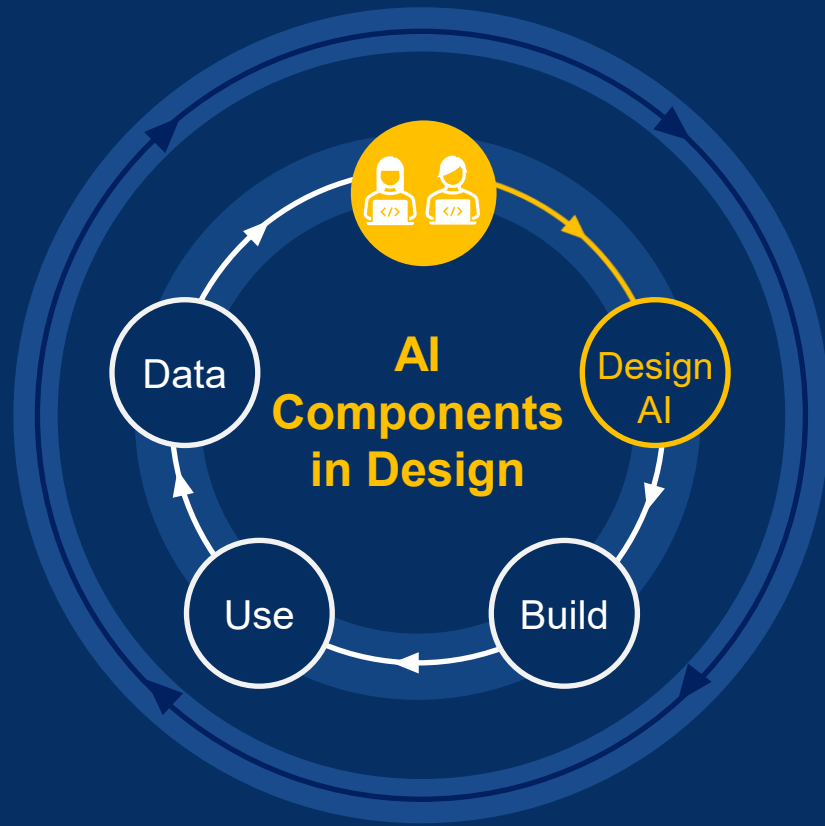
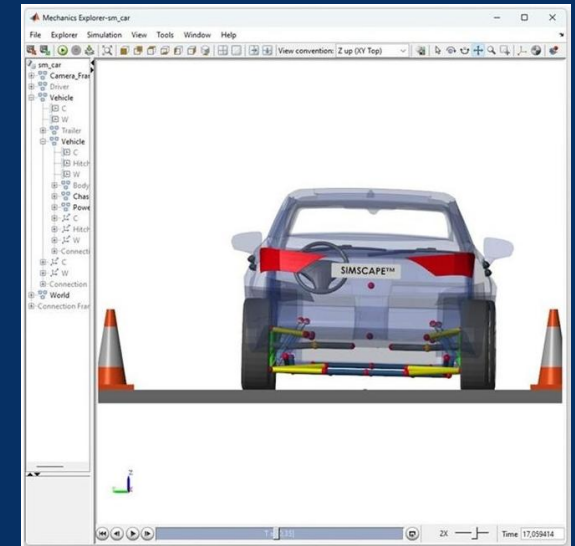
Source: <https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/company/events/conferences/matlab-expo-2023/ww-2023-expo-pseudo-pressure-sensor-to-improve-diagnostics-in-a-solenoid-actuated-valve.pdf>

AI becomes part of the design



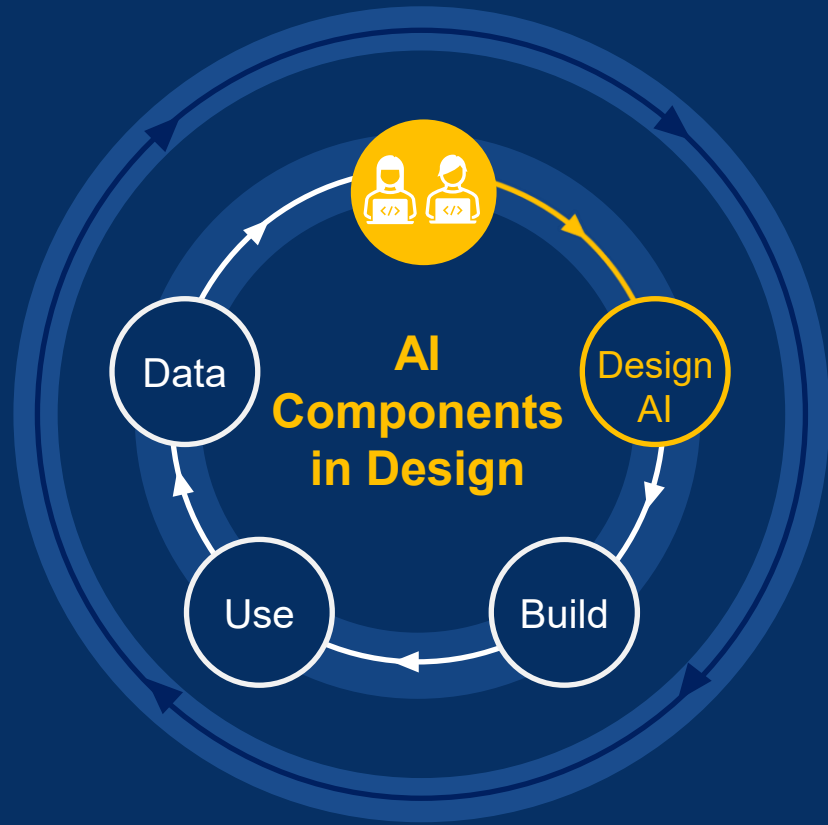
AI reduced order models in the design loops

Physics-based
Model



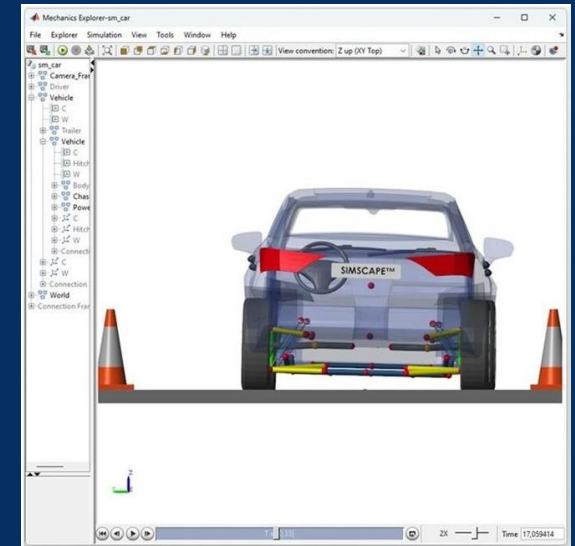
Approximate the physics-based model with AI

Optimization of Design Parameters



A loop within a design loop!

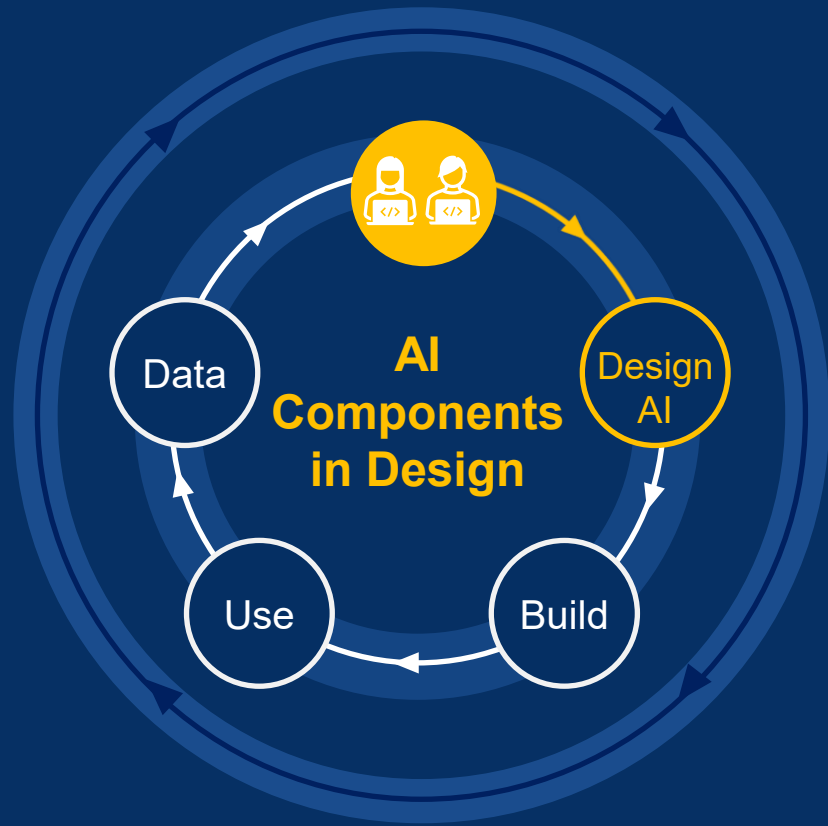
Physics-based
Model



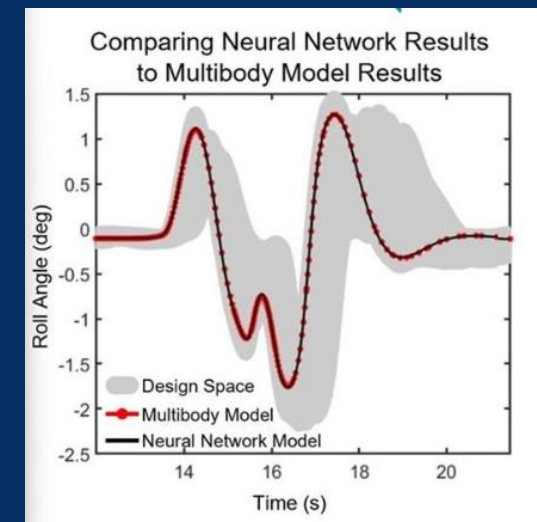
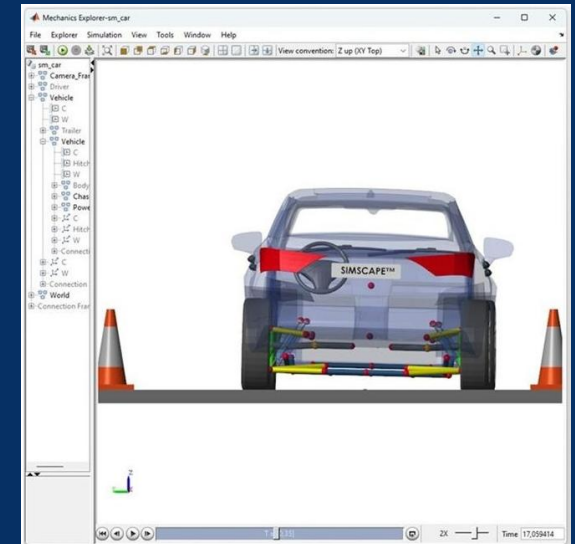
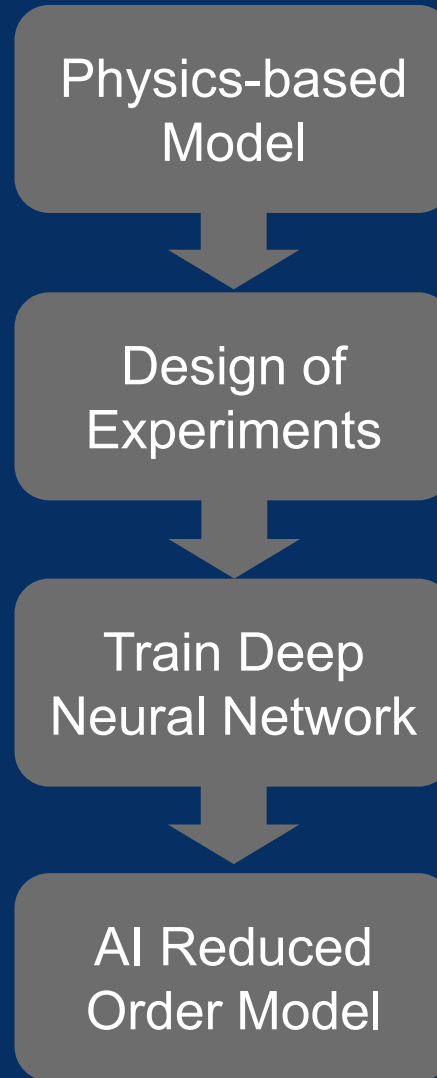
16 Days

Approximate the physics-based model with AI

Optimization of Design Parameters

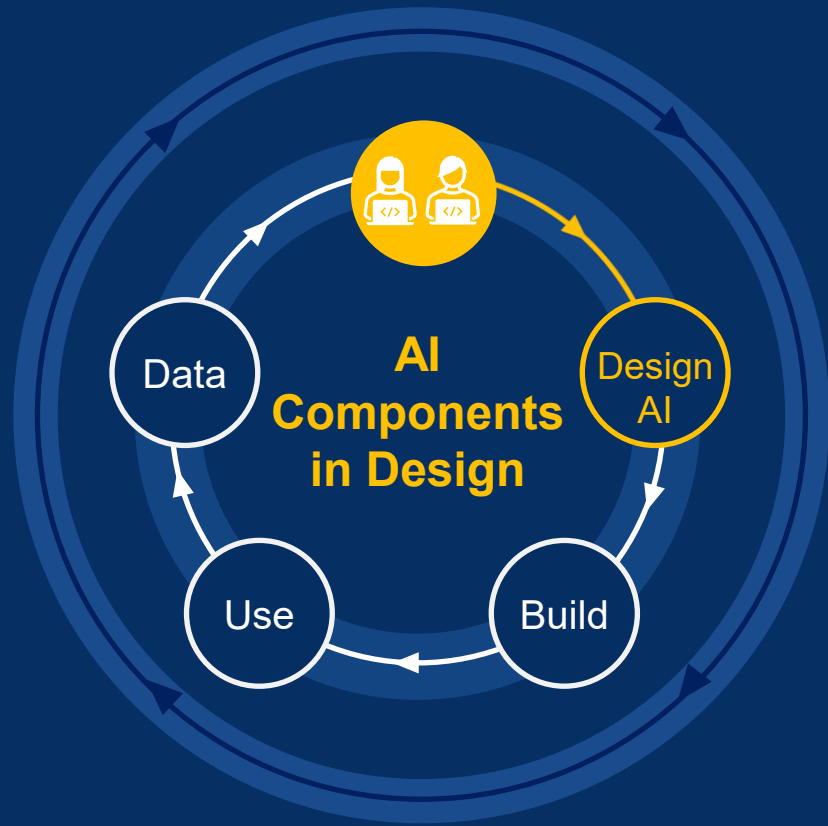


A loop within a design loop!



Approximate the physics-based model with AI

Optimization of Design Parameters

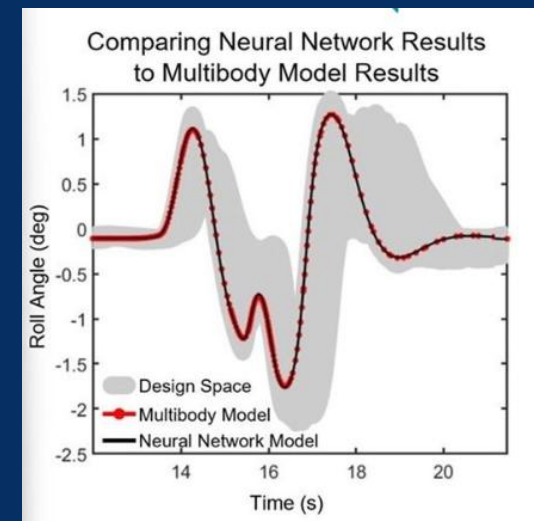


A loop within a design loop!

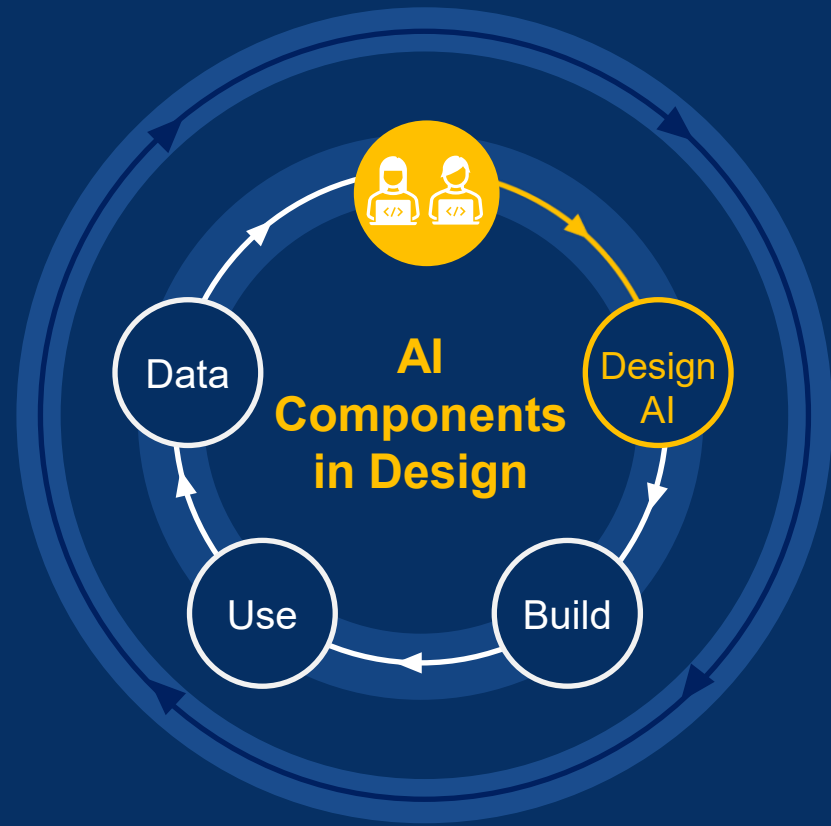
AI Reduced Order Model



5 Minutes

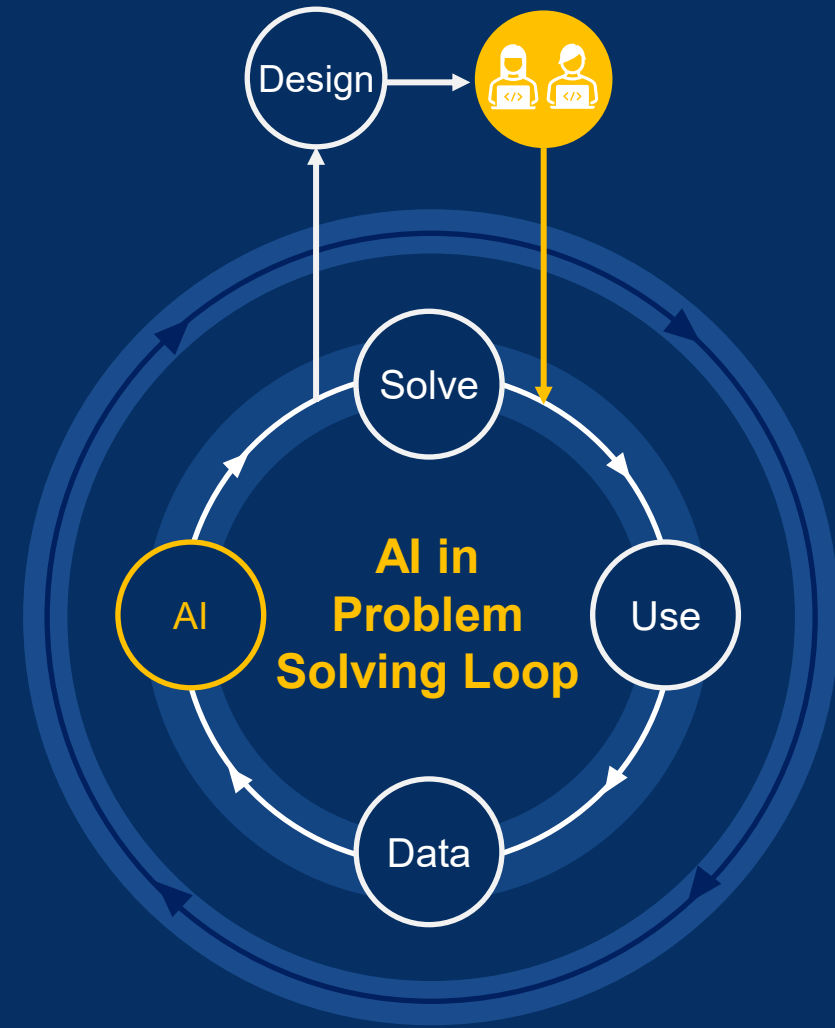
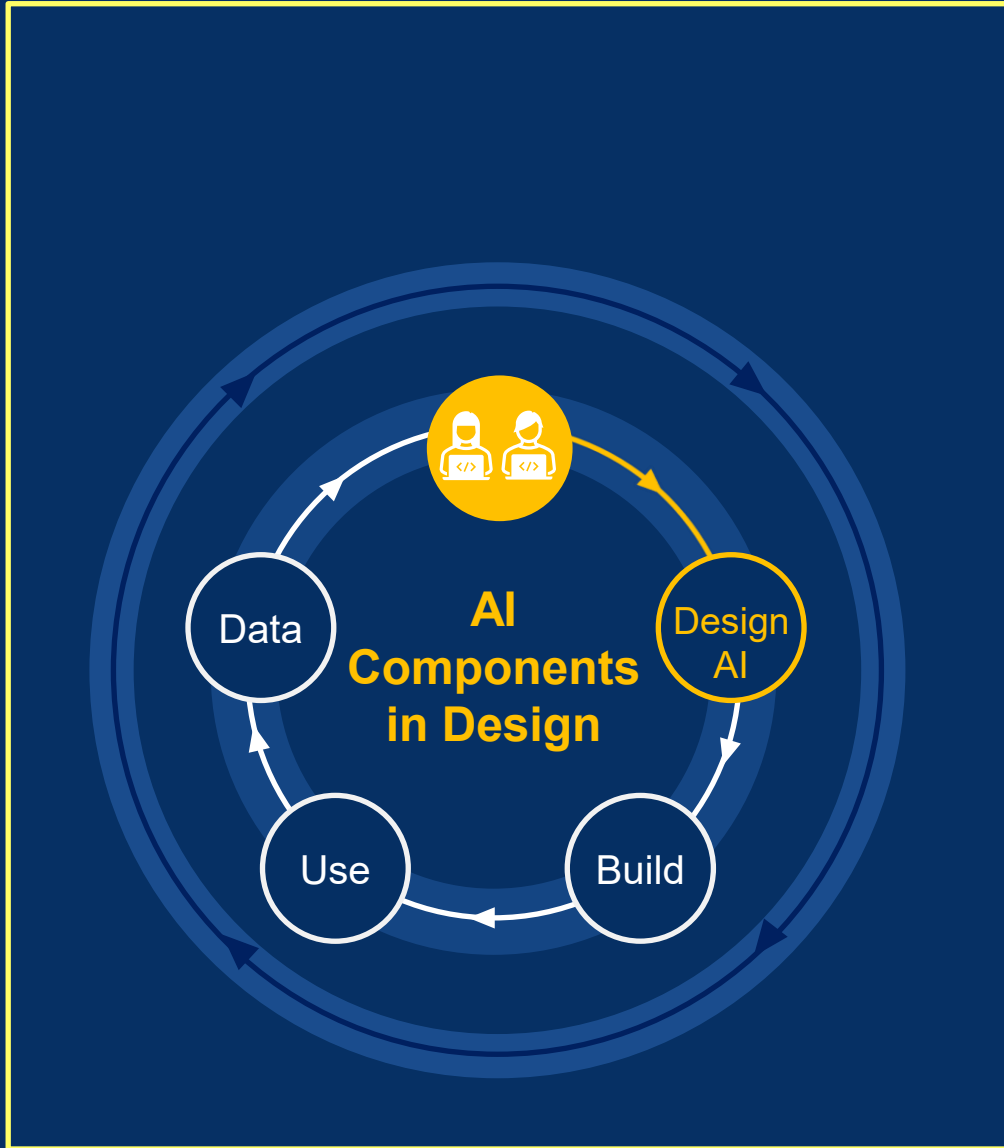


AI enhances the design process

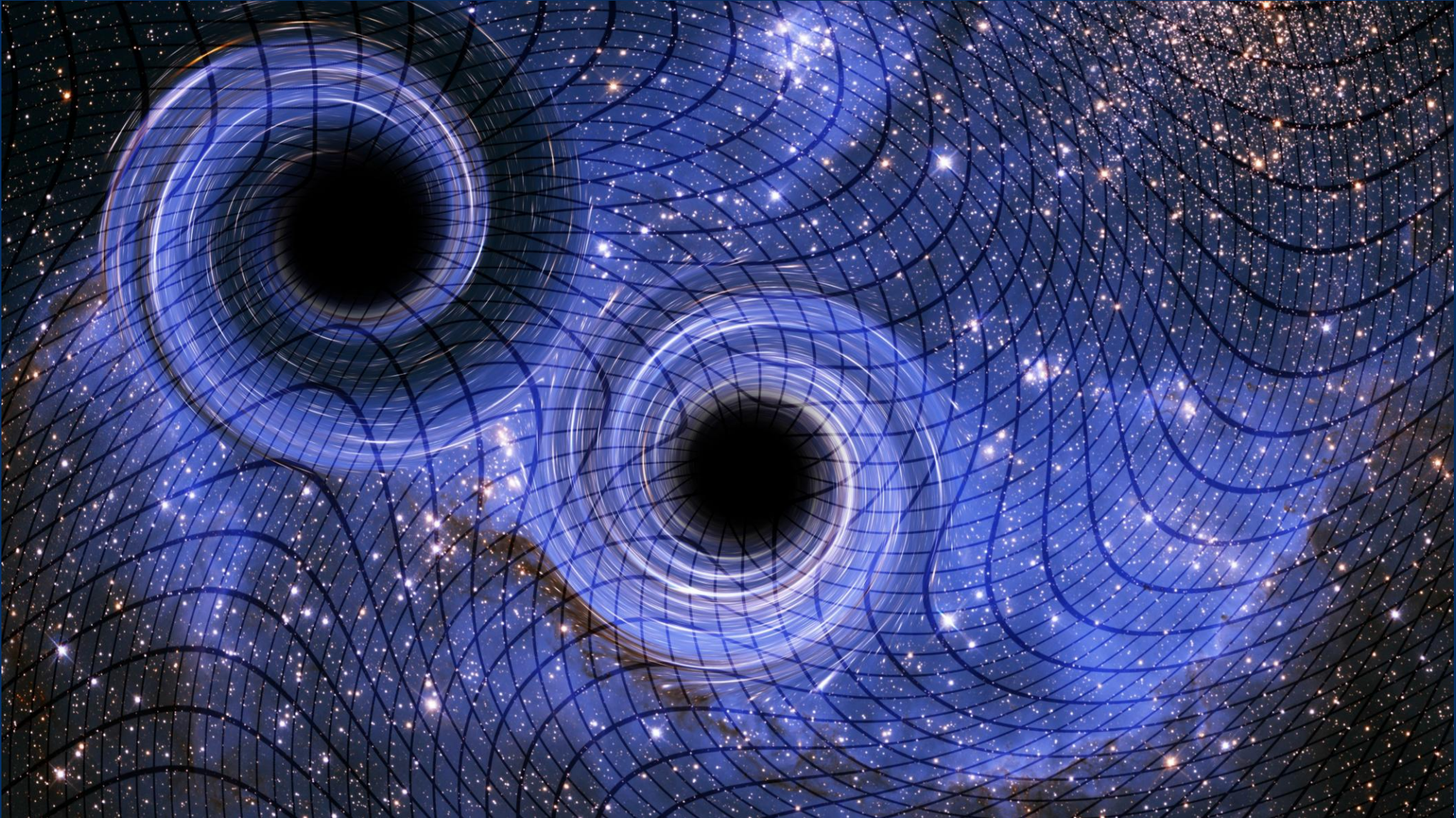


Design loops go **faster-and-faster** with AI

AI can control the problem-solving loop



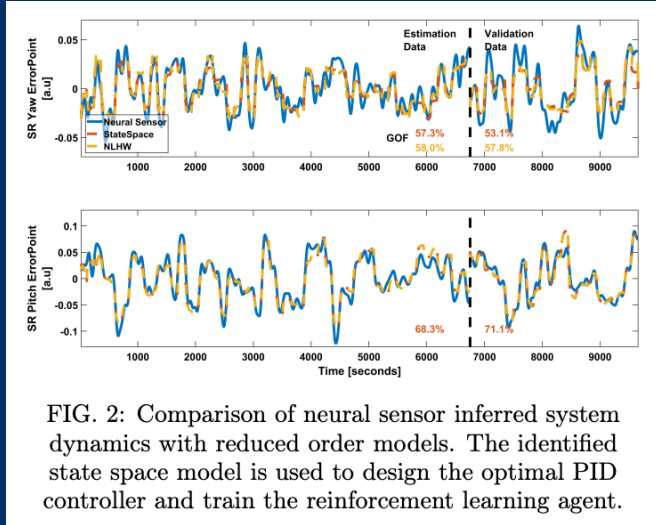
Reinforcement learning for gravitational wave detection



Source: N. Mukund et al. "Neural Sensing and Control in a Kilometer-Scale Gravitational-Wave Observatory." Phys. Rev. Applied, vol. 20, no. 6, 22 Dec. 2023, <https://doi.org/10.1103/physrevapplied.20.064041>. Accessed 22 Aug. 2025. [Licensed under CC-BY-4.0.](#)

Defining the components of the loop

System Model



Source: N. Mukund et al. "Neural Sensing and Control in a Kilometer-Scale Gravitational-Wave Observatory." Phys. Rev. Applied, vol. 20, no. 6, 22 Dec. 2023, <https://doi.org/10.1103/physrevapplied.20.064041>. Accessed 22 Aug. 2025. Licensed under CC-BY-4.0.

Reward Function

$$\text{Cost} = \sum_{j=1}^T (S_j - S_j^{\text{ref}})^T Q_j (S_j - S_j^{\text{ref}}) + (A_j - A_j^{\text{prev}})^T R_j (A_j - A_j^{\text{prev}}) \quad (5)$$

$$\text{Penalty} = W_y \left((S_j - S^{\text{min}})^2 + (S_k - S^{\text{max}})^2 \right) + W_{\text{mvrate}} \left((\dot{A}_l - \dot{A}^{\text{max}})^2 + (\dot{A}_m - \dot{A}^{\text{min}})^2 \right) \quad (6)$$

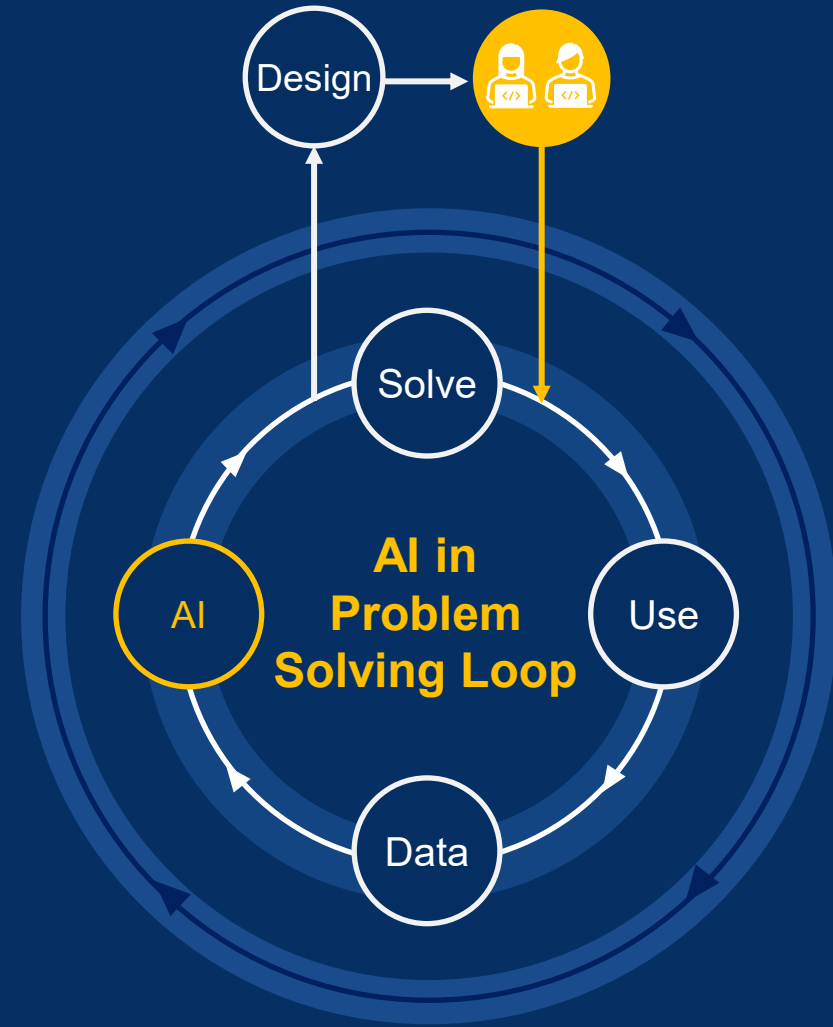
$$\forall \left(S_j < S^{\text{min}}, S_k > S^{\text{max}}, \dot{A}_l < \dot{A}^{\text{min}}, \dot{A}_m > \dot{A}^{\text{max}} \right) \quad (7)$$

$$\text{Boost} = 10 \sum_{j=1}^T (3 |S_j - S_j^{\text{ref}}| < 0.02)^2 + 10 \sum_{j=1}^T (6 |S_j - S_j^{\text{ref}}| < 0.005)^2 \quad (8)$$

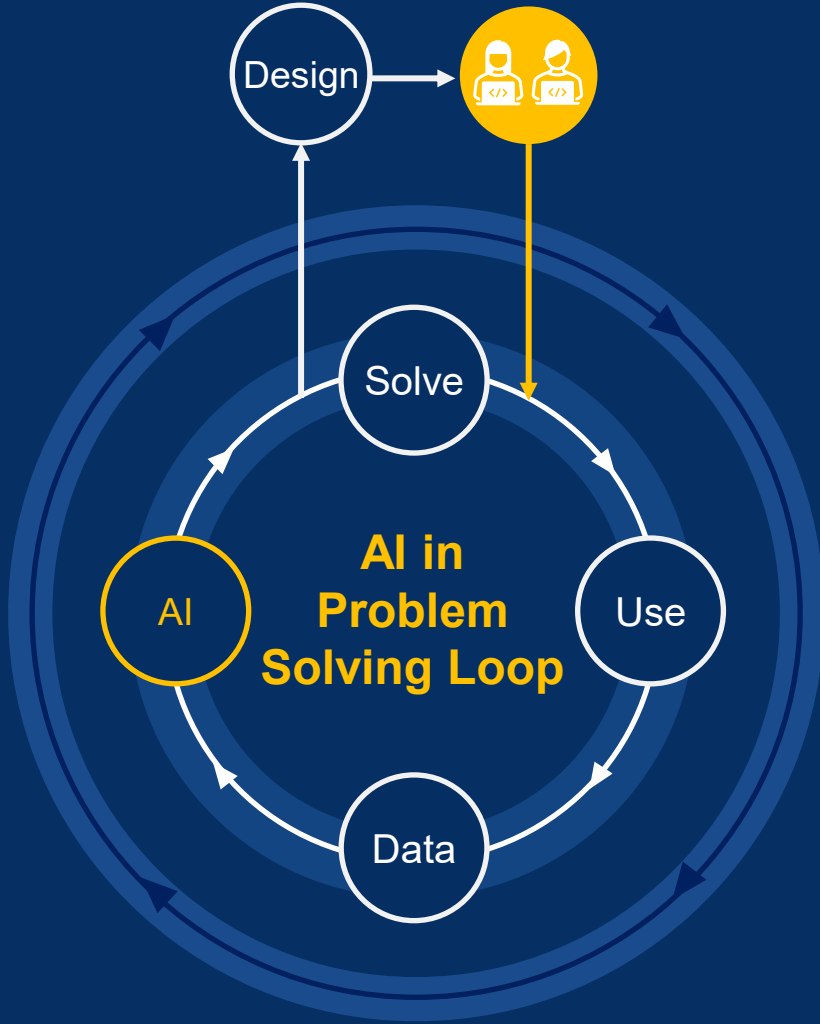
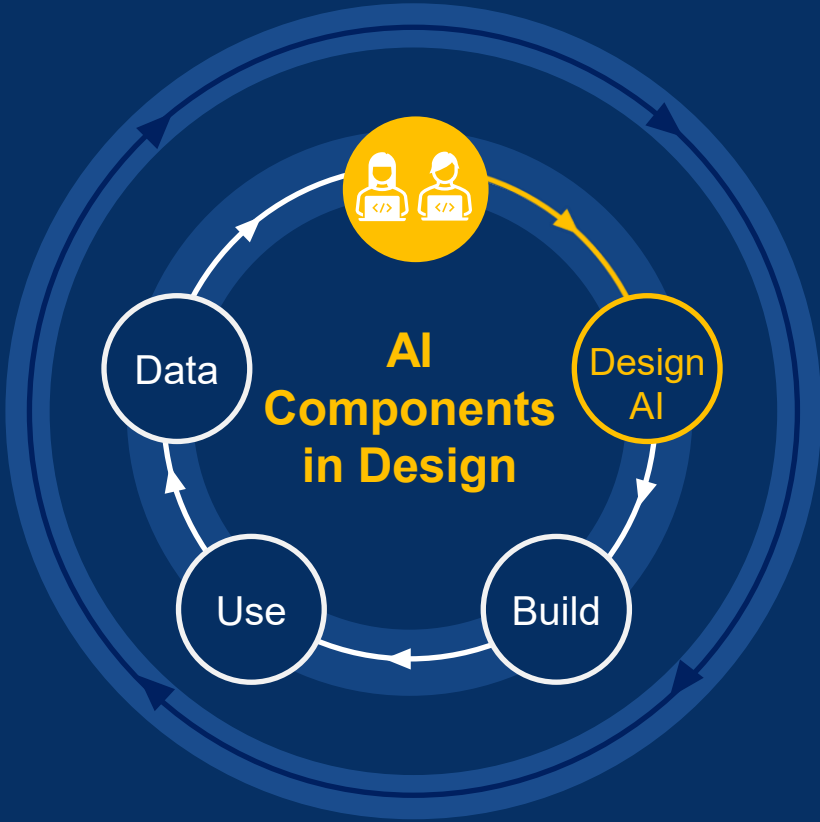
$$\text{Reward} = -(\text{Cost} + \text{Penalty}) + \text{Boost} \quad (9)$$

TABLE II: Reward function equations used to train the reinforcement learning agent. The continuous part is built using the linear quadratic regulator cost function. Discrete terms, penalty and boost, are added to penalize the violation of boundary constraints and emulate final state constraints.

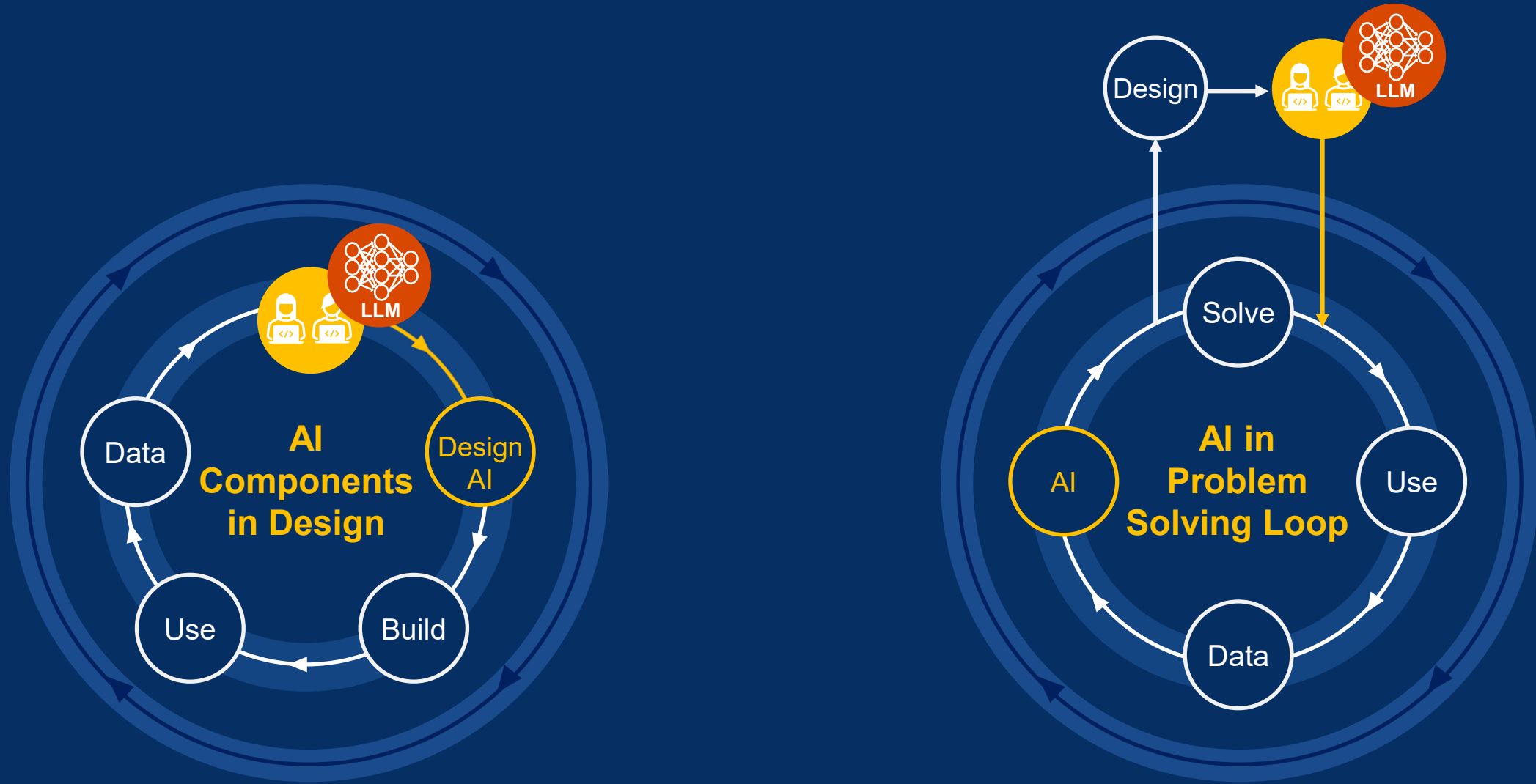
Source: N. Mukund et al. "Neural Sensing and Control in a Kilometer-Scale Gravitational-Wave Observatory." Phys. Rev. Applied, vol. 20, no. 6, 22 Dec. 2023, <https://doi.org/10.1103/physrevapplied.20.064041>. Accessed 22 Aug. 2025. Licensed under CC-BY-4.0.



Embedding intelligence in engineering design



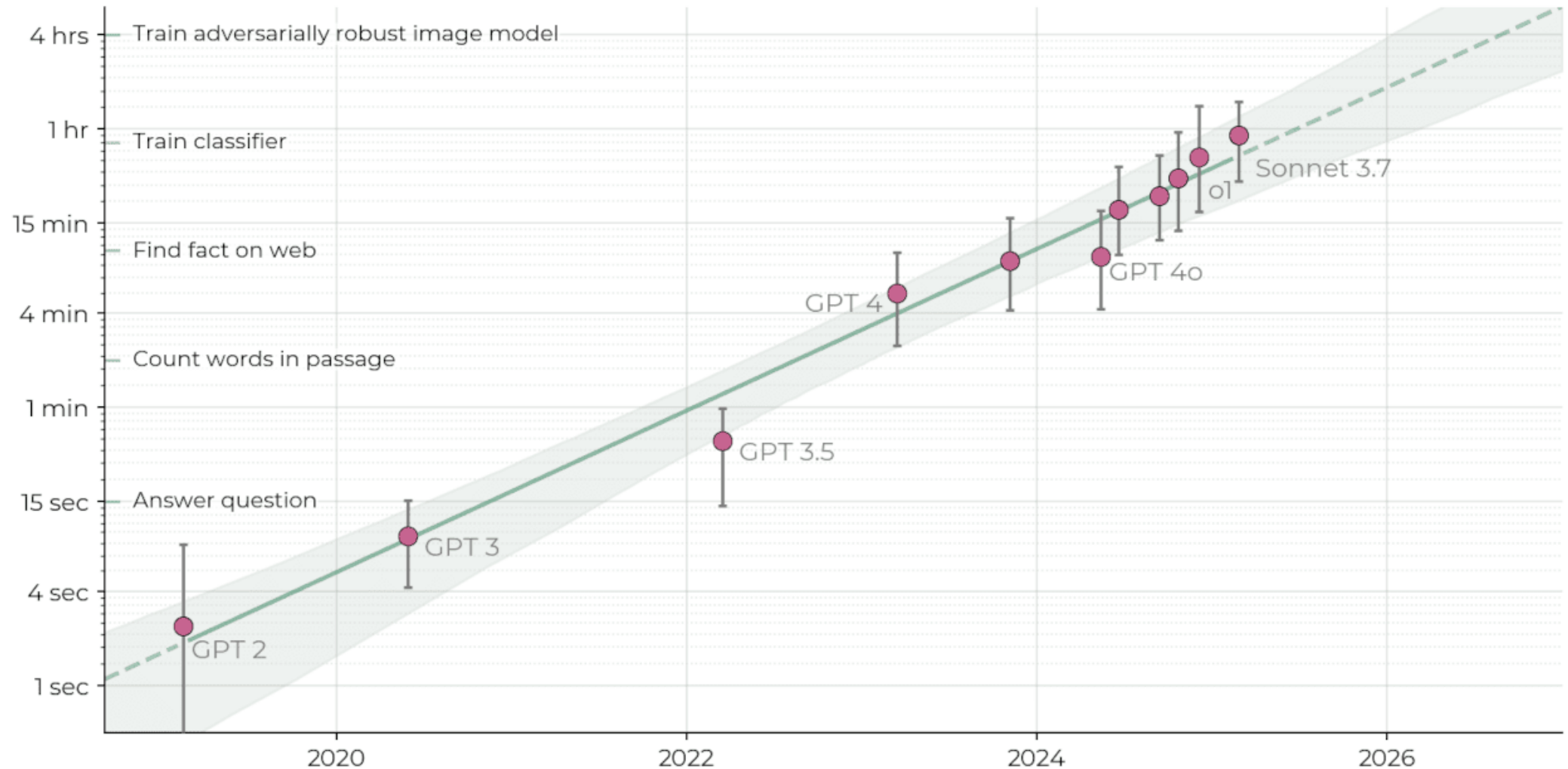
Generative AI assists engineers throughout design loops



The length of tasks AI can do is doubling every 7 months



Task length (at 50% success rate)



CC-BY

Model release date

metr.org

Generative AI pushes the boundaries

The MATLAB Editor interface shows the Copilot tool generating code for a plot. A tooltip above the code reads: "Put both lines on the same chart, with torque on the left and current on the right. Generated code might be inaccurate. Validate before use." The code in the editor is as follows:

```

1 % Create a new figure
2 figure;
3
4 % Create axes for the current and torque
5 yyaxis left
6 plot(time, current, 'LineWidth', 1);
7 ylabel('Current (A)');
8 grid on
9 title('Motor Current and Load Torque');
10
11 yyaxis right
12 plot(time, -torque, 'LineWidth', 1);
13 ylabel('Torque (N*m)');
14 ylim([-3e-4 5e-5]);
15
16 % Add x-axis label
17 xlabel('Time (s)');
18

```

At the bottom, the Command Window displays the instruction: ">> Press **F9** to generate code with Copilot".

The MATLAB Simulation interface displays a "Wind Turbine" model. The model includes several sub-diagrams: "Supervisory Control", "Turbine coupled with Grid and Transformer", "Turbine Pitch Controller", and "MPPT, Derating with Limiter Control". A "Results" block shows inputs like Pitch Brake State, Generator Power, and Wind Speed (m/s).

The Copilot Chat (Beta) window on the right provides the following suggestions:

Can you provide specific suggestions on improving the control system, the state machine, and the physical system model of the wind turbine? List the tools that I should use, including the links to them.

I will analyze your `sscv_wind_turbine` model to provide specific suggestions for improving the control system, state machine, and physical system model. Additionally, I will gather documentation on relevant tools that can assist you in these improvements.

Please hold on while I perform these tasks.

Analyzed the `sscv_wind_turbine` model for suggestions on improvements. Retrieved documentation on tools for improving control systems and models.

Based on the analysis of your `sscv_wind_turbine` model, here are specific suggestions for improving the control system, state machine, and physical system model:

Control System Improvements

- Advanced MPPT Algorithms:** Implement sophisticated Maximum Power Point Tracking (MPPT) methods, such as Incremental Conductance or Fuzzy Logic Control, to enhance tracking efficiency under varying wind conditions.
- Adaptive Control Strategies:** Introduce adaptive control techniques that adjust control parameters in real-time based on environmental changes to improve responsiveness and stability.

State Machine Enhancements

- State Transition Optimization:** Minimize unnecessary state changes by analyzing the transition conditions and implementing hysteresis to prevent oscillations between states.
- Fault Detection and Recovery:** Integrate mechanisms for detecting faults and abnormal conditions, allowing for effective transitions to safe operational states.

Physical System Model Enhancements

- Enhanced Mechanical Modeling:** Incorporate detailed mechanical models that account for non-linearities and dynamic interactions, such as friction losses.
- Energy Storage Integration:** Consider integrating energy storage solutions (like batteries) to buffer power output fluctuations and enhance grid stability.

Recommended Tools and Documentation

Ask Copilot

Validate generated output before use. [Deeper Insights \(2/40\)](#)

What can generative AI do for engineers?

How do I... ?

Explain & Advise

Make a ...

Create & Improve

Check that ...

Verify & Automate

Embedding intelligence in MATLAB, Simulink, and Polyspace

MATLAB Copilot

R2025a

Simulink Copilot

R2026a

Polyspace Copilot

R2026a

How do I... ?

Explain & Advise

Make a ...

Create & Improve

Check that ...

Verify & Automate

HOME PLOTS APPS PROJECT LIVE EDITOR INSERT VIEW Search (⌘ Space) Seth

New Open Save Print Export Compare Go To Find Bookmark

Text Code Control Task Copilot Refactor Code Issues Debugger Generate Test Find Tests Run Section Run and Advance Run to End Run Step Stop

FILE NAVIGATE TEXT CODE ANALYZE TEST SECTION RUN

/ > Users > sdeland > Documents > MATLAB > projects > AlforAutomotive > VirtualSensorExample >

script.mlx * x +

/Users/sdeland/Documents/MATLAB/projects/AlforAutomotive/VirtualSensorExample/script.mlx

6

```
VariableNames=["Voltage","Current","Temperature","V_avg","I_avg","SOC"])
```

t = 669956x6 table

	Voltage	Current	Temperature	V_avg	I_avg	SOC
1	0.3851	0.7510	0.3031	0.3851	0.7510	0.2064
2	0.3852	0.7510	0.3046	0.3851	0.7510	0.2064
3	0.3852	0.7510	0.3061	0.3852	0.7510	0.2064
4	0.3852	0.7510	0.3076	0.3852	0.7510	0.2064
5	0.3852	0.7510	0.3091	0.3852	0.7510	0.2064
6	0.3852	0.7510	0.3106	0.3852	0.7510	0.2064
7	0.3852	0.7510	0.3120	0.3852	0.7510	0.2064
8	0.3852	0.7510	0.3135	0.3852	0.7510	0.2064
9	0.3852	0.7510	0.3150	0.3852	0.7510	0.2064

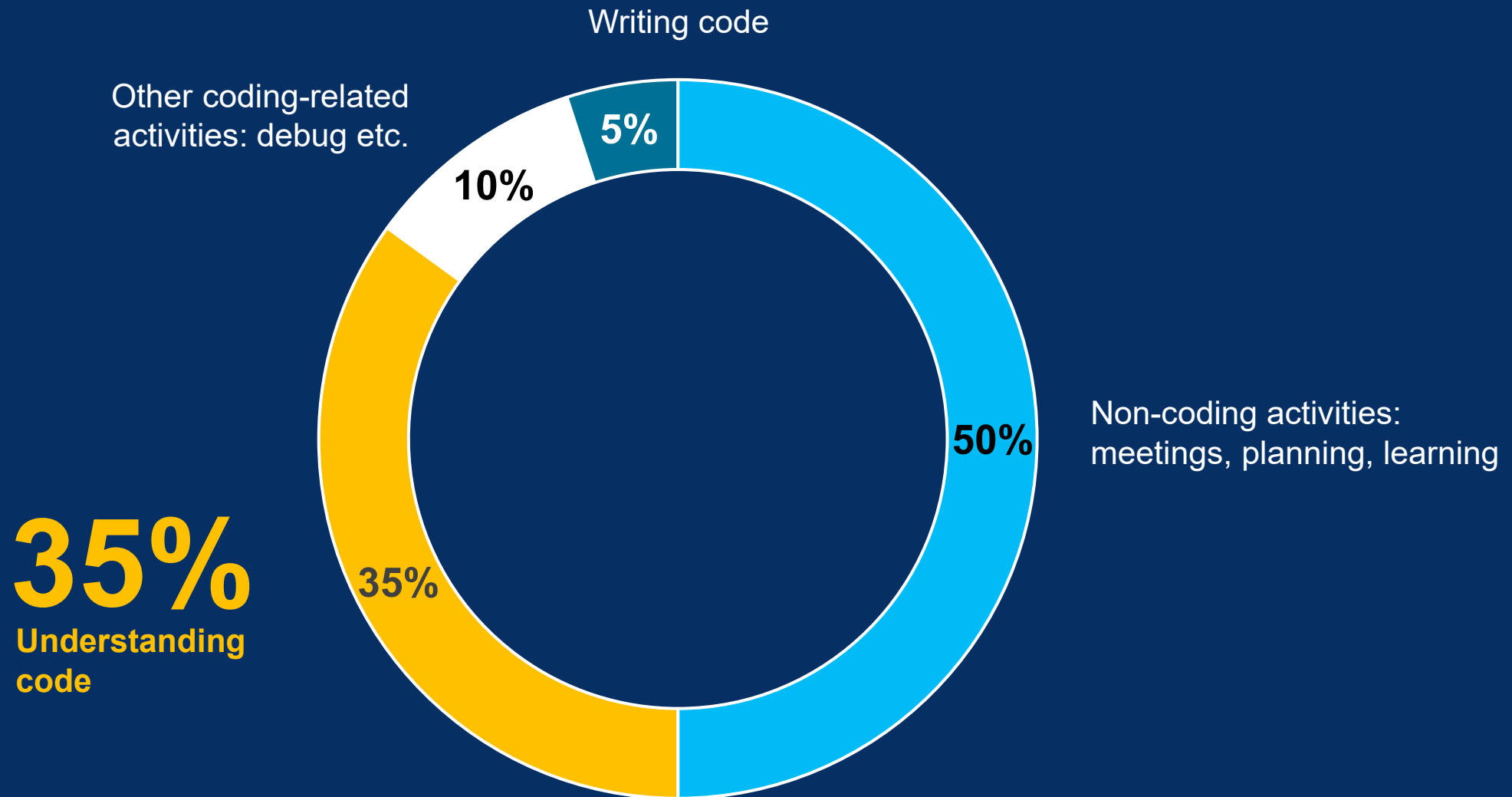
References

[1] Kollmeyer, Phillip, Carlos Vidal, Mina Naguib, and Michael Skells. "LG 18650HG2 Li-Ion Battery Data and Example Deep Neural Network XEV SOC Estimator Script." Mendeley, March 5, 2020. <https://doi.org/10.17632/CP3473X7XV.3>.

7

|

Authoring code is only one part of the story



Based on Povarov, Nikita (2022, May 20) *AI for Software Developers: a Future or a New Reality?*. InfoQ. <https://www.infoq.com/articles/ai-for-software-developers/>

```
str = regexprep(str, "[0-9](?=(?:[0-9]{3})+(?![0-9]))", "$&," );
```

MATLAB R2025a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Print Compare Go To Find Bookmark Copilot Refactor Profiler Code Issues Generate Test Find Tests Run Section Run Step Stop

FILE NAVIGATE CODE ANALYZE TEST SECTION RUN

/ > Users > sdeland > Documents > MATLAB > Work

formatNumbersAsStrings.m × +

```

1 function str = formatNumbersAsStrings(num)
2     num = round(num);
3     str = string(num);
4     str = regexprep(str, '[0-9](?=(?:[0-9]{3})+(?![0-9]))', '$&,');
5 end

```

Copilot Chat

Copilot

Hello! Copilot is here to answer your questions, help write and explain code, and even identify code issues. Learn more about [Copilot](#).

Remember that Copilot sometimes writes code and text that seems accurate, but is not. Make sure to verify any received code and give feedback on the results to help improve the responses.

Shuffle Example Prompts

- Read "peppers.png" and display as grayscale
- Plot a sine wave and add a vertical line using vline

Ask Copilot

Validate generated output before use.

Editor: 110% UTF-8 LF formatNumbersAsStrings Ln 4 Col 66

HOME

PLOTS

APPS

PROJECT

EDITOR

PUBLISH

VIEW



New



Open



Save



Print



Compare

FILE



Go To



Find

Bookmark

NAVIGATE



Copilot



Refactor

CODE



Profiler



Code Issues



Debugger

ANALYZE



Generate Test

TEST



Find Tests



Run Section

SECTION

/ > Users > sdeland > Documents > MATLAB > Work

formatNumbersAsStrings.m × +

/Users/sdeland/Documents/MATLAB/Work/formatNumbersAsStrings.m

```
1 function str = formatNumbersAsStrings(num)
2     num = round(num);
3     str = string(num);
4     str = regexprep(str, '[0-9](?=(?:[0-9]{3})+(?![0-9]))', '$&');
5 end
```



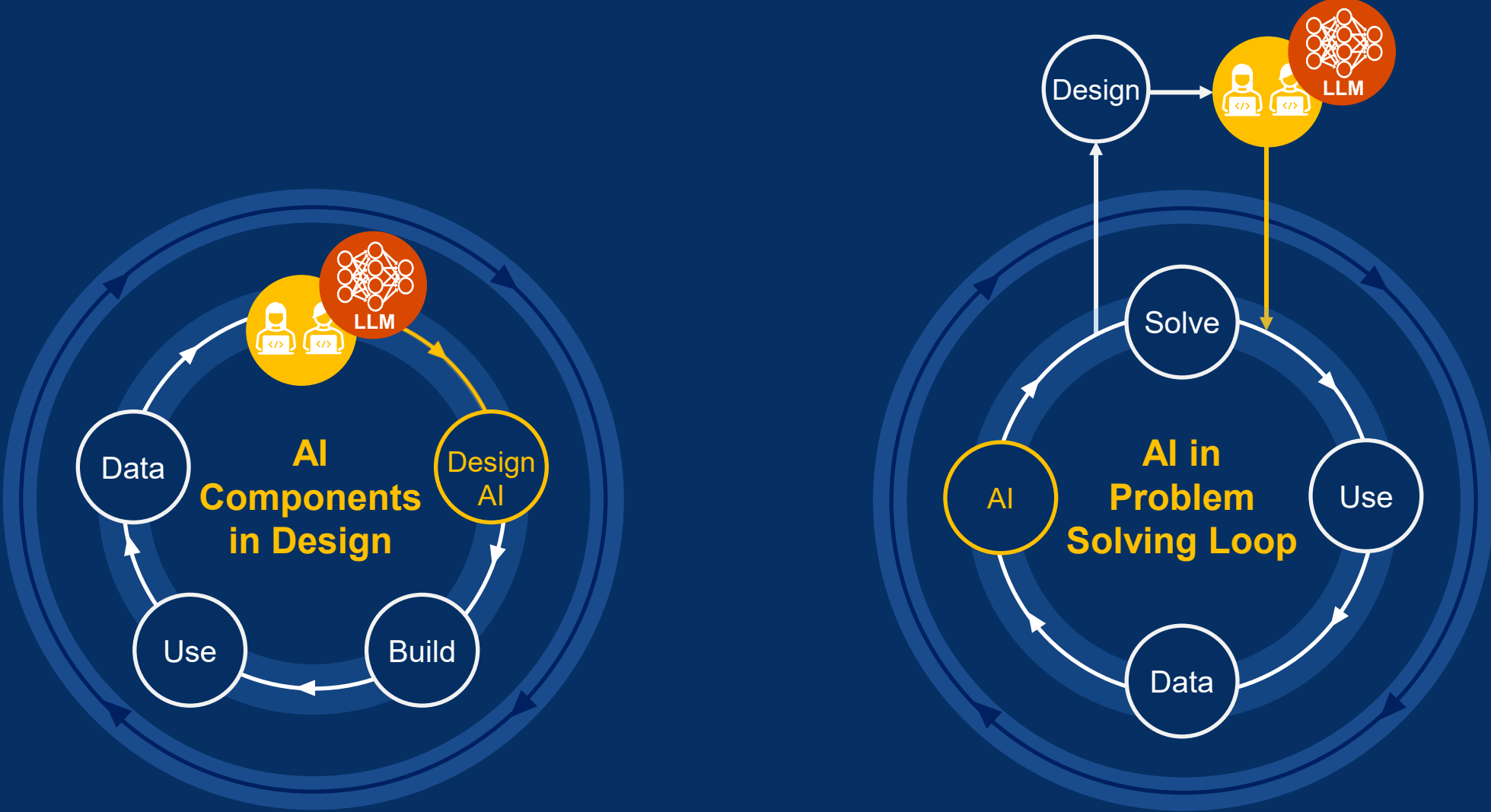
```
5      %      num - numeric value to be formatted
6      %
7      % Output Arguments:
8      %      str - formatted string representation of the number
9
10     % Round the number to the nearest integer
11     num = round(num);
12     % Convert the rounded number to a string
13     str = string(num);
14     % Insert commas as thousand separators in the string
```

Command Window

>> ✨ Press  to generate code with Copilot



Design loops go faster with **Generative AI**



Bringing Generative AI to Simulink

The screenshot displays the Simulink environment for a wind turbine model. The main workspace shows four sub-diagrams: Supervisory Control, Turbine Pitch Controller, Turbine coupled with Grid and Transformer, and MPPT, Derating with Limiter Control. A Copilot chat window on the right contains the following text:

Tianyi

Can you provide specific suggestions on improving the control system, the state machine, and the physical system model of the wind turbine? List the tools that I should use, including the links to them.

Copilot
Reasoned for 4 sec >

I will analyze your sscv_wind_turbine model to provide specific suggestions for improving the control system, state machine, and physical system model. Additionally, I will gather documentation on relevant tools that can assist you in these improvements.

Please hold on while I perform these tasks.

Analyzed the sscv_wind_turbine model for suggestions on improvements. Retrieved documentation on tools for improving control systems and models.

Based on the analysis of your sscv_wind_turbine model, here are specific suggestions for improving the control system, state machine, and physical system model:

Control System Improvements

- Advanced MPPT Algorithms:** Implement sophisticated Maximum Power Point Tracking (MPPT) methods, such as Incremental Conductance or Fuzzy Logic Control, to enhance tracking efficiency under varying wind conditions.
- Adaptive Control Strategies:** Introduce adaptive control techniques that adjust control parameters in real-time based on environmental changes to improve responsiveness and stability.

State Machine Enhancements

- State Transition Optimization:** Minimize unnecessary state changes by analyzing the transition conditions and implementing hysteresis to prevent oscillations between states.
- Fault Detection and Recovery:** Integrate mechanisms for detecting faults and abnormal conditions, allowing for effective transitions to safe operational states.

Physical System Model Enhancements

- Enhanced Mechanical Modeling:** Incorporate detailed mechanical models that account for non-linearities and dynamic interactions, such as friction losses.
- Energy Storage Integration:** Consider integrating energy storage solutions (like batteries) to buffer power output fluctuations and enhance grid stability.

Recommended Tools and Documentation

Ask Copilot

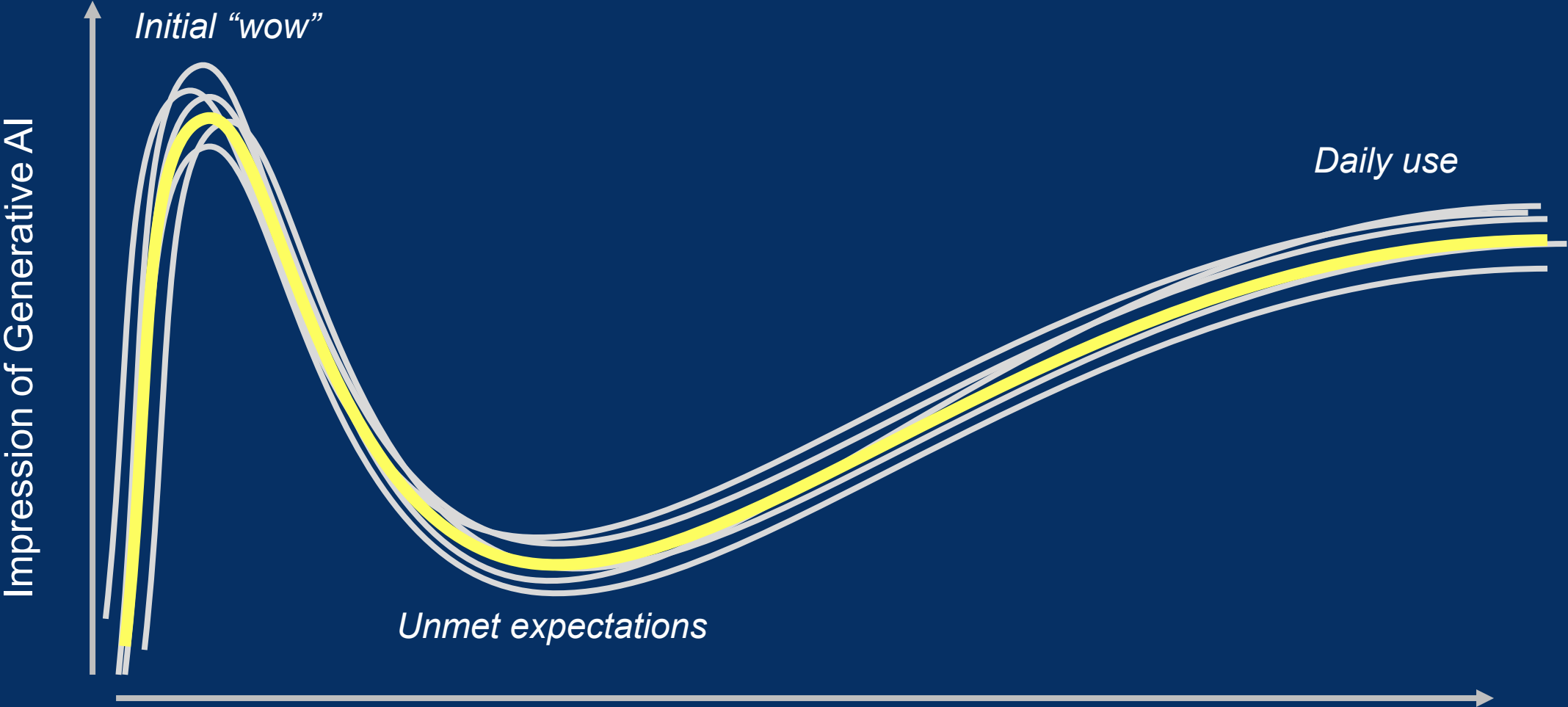
Validate generated output before use. Deeper Insights (2/40)

Learning how to use Generative AI in engineering

“Like learning to work with a new colleague.”

- Memorized the internet
- Completes tasks almost instantly
- Excels at simple tasks and does not complain
- Can be overconfident in more complex tasks

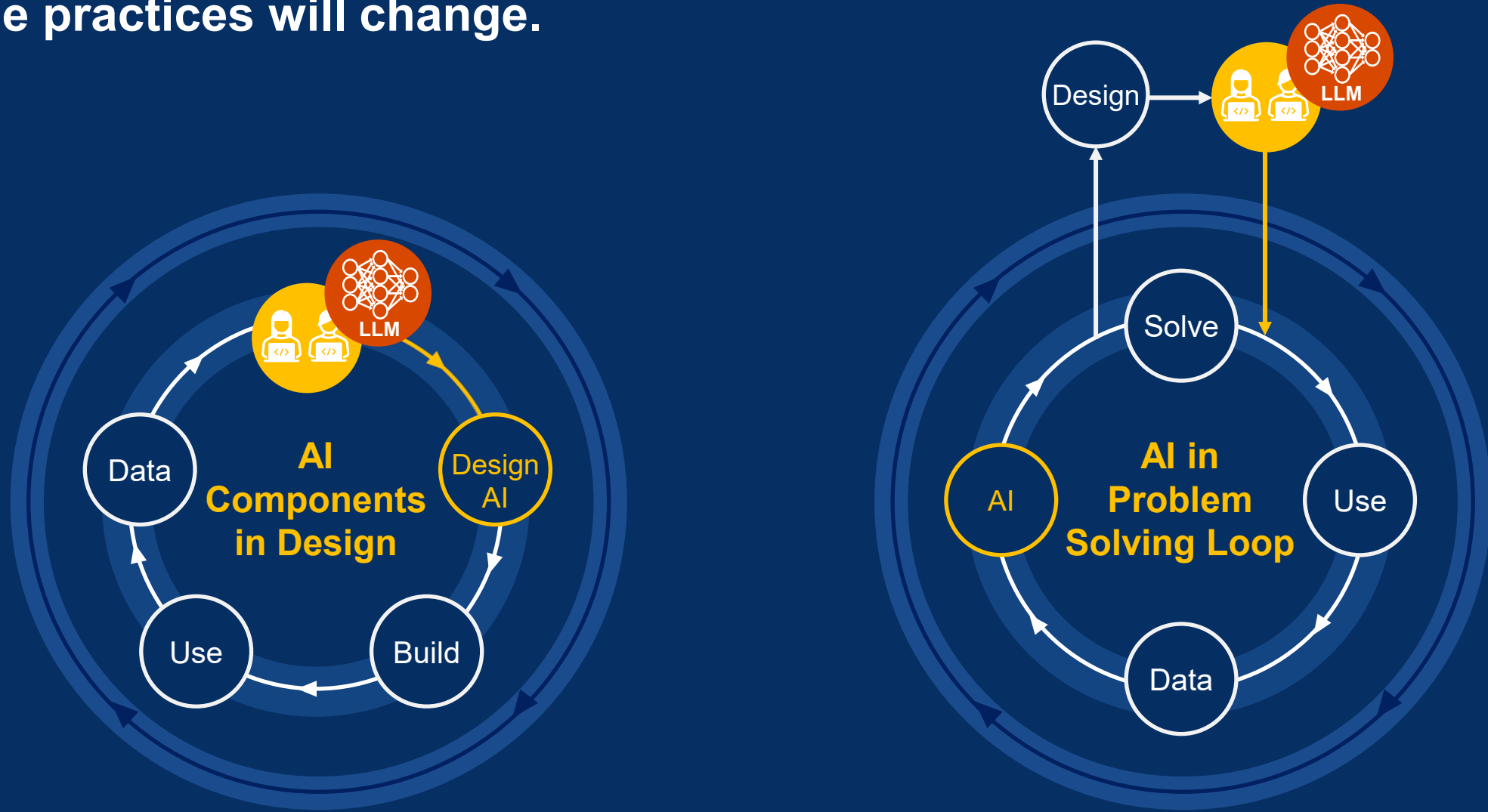
Learning to use Generative AI takes time



Engineers should learn to harness Generative AI

- Get hands-on experience
- Adapt your skillset
- This will evolve over years, but it's critical that engineers act now

The principles of engineering design will not change, but the practices will change.



Engineers need to become more goal-oriented

MATLAB EXPO



© 2025 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

