U.S. DEPARTMENT OF ENERGY | Office of NUCLEAR ENERGY

ASI | Advanced Sensors and Instrumentation

Creating a Simulation Platform for Research and Development of Advanced Control Algorithms

Advanced Sensors and Instrumentation (ASI) Annual Program Webinar

October 30 – November 2, 2023

Jake Farber, Ph.D.

Ahmad Al Rashdan Ph.D. (PI)

Idaho National Laboratory INL

Acknowledgements

Team members:

Craig Primer

Maria Coelho

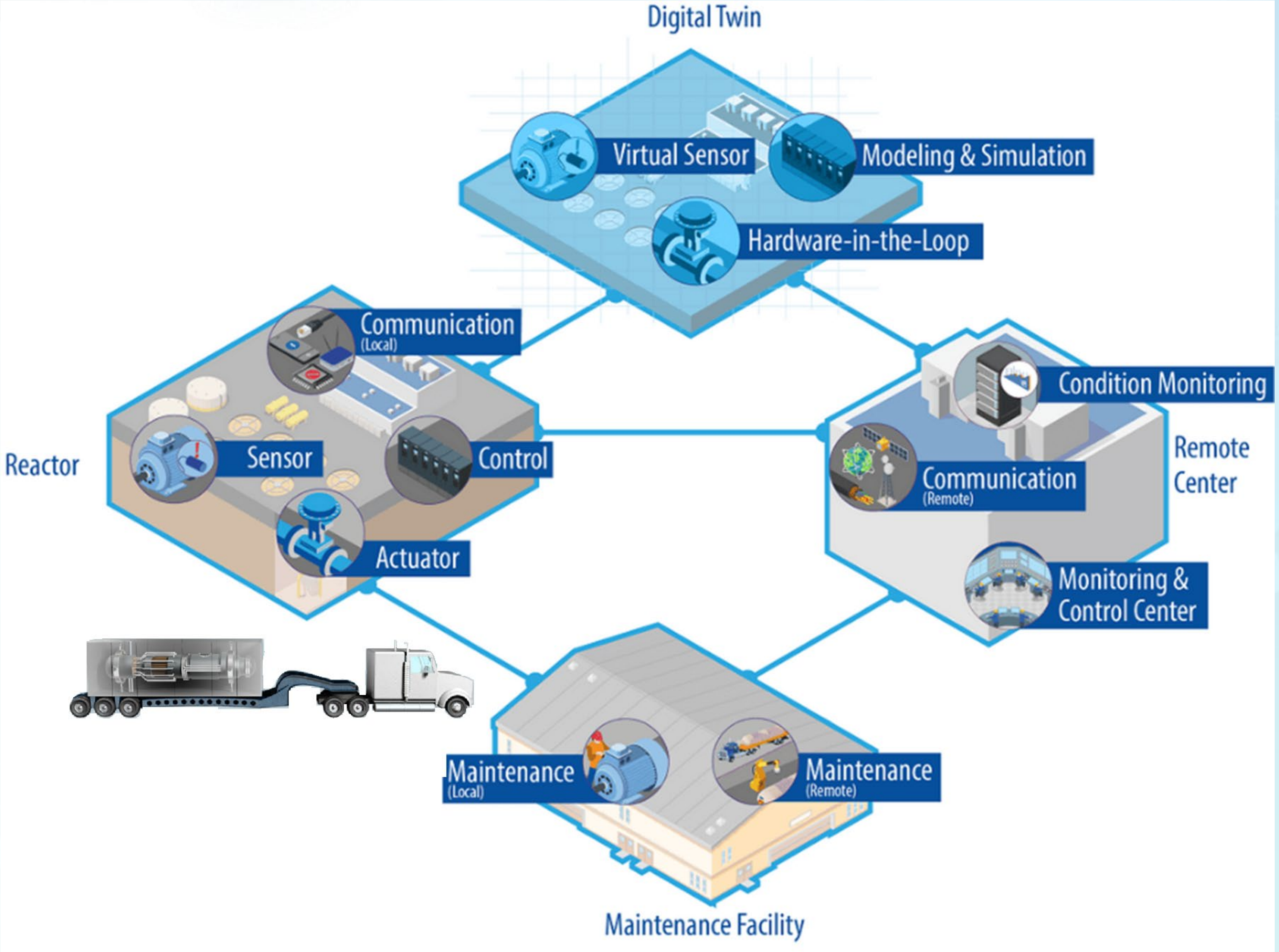
Vaibhav Yadav

Joe Oncken

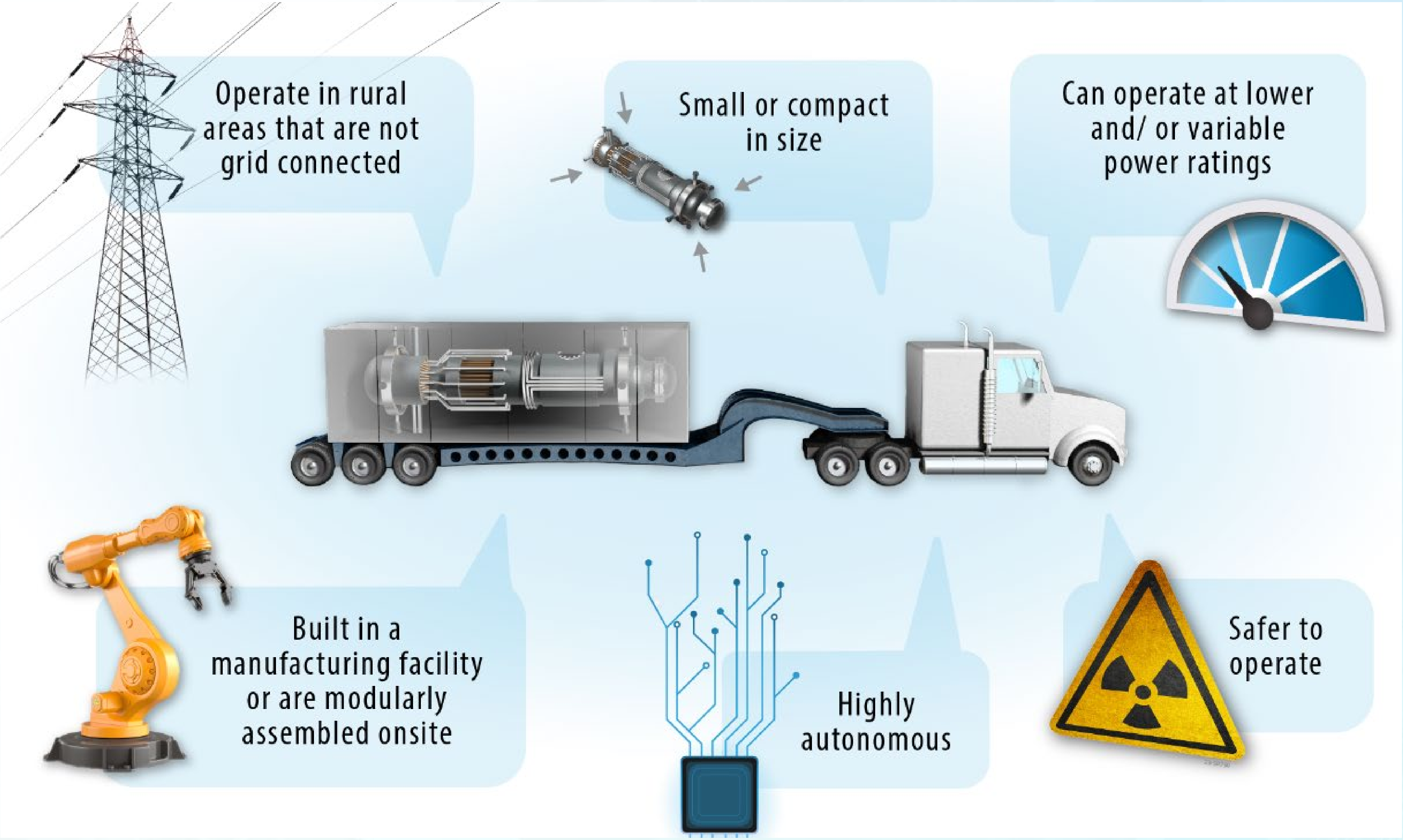
Other Idaho National Laboratory Collaborators:

Anthony Crawford

Carlo Parisi



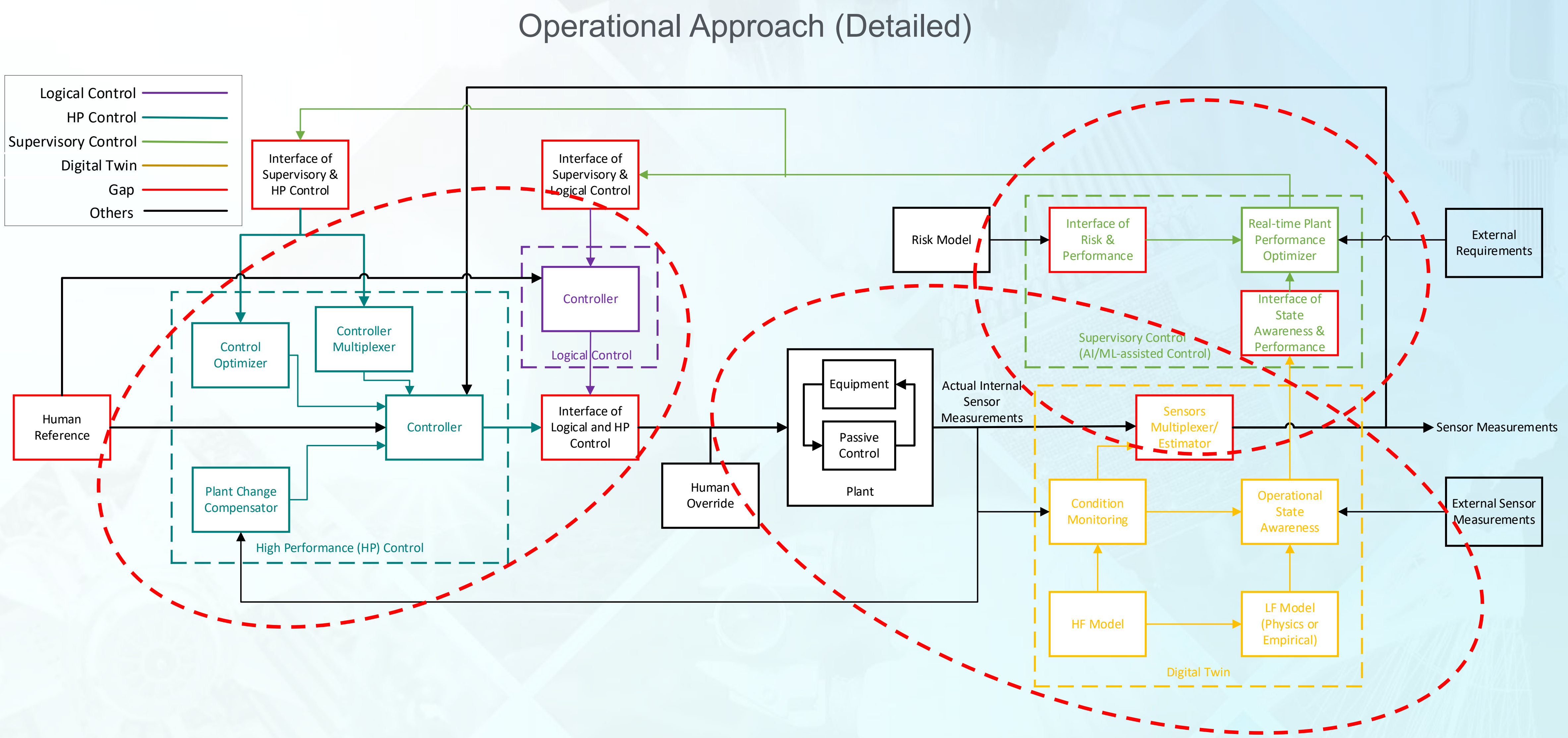
Advanced reactors will reduce construction and operating costs and increase flexibility, but will require more advanced controls and automation techniques.



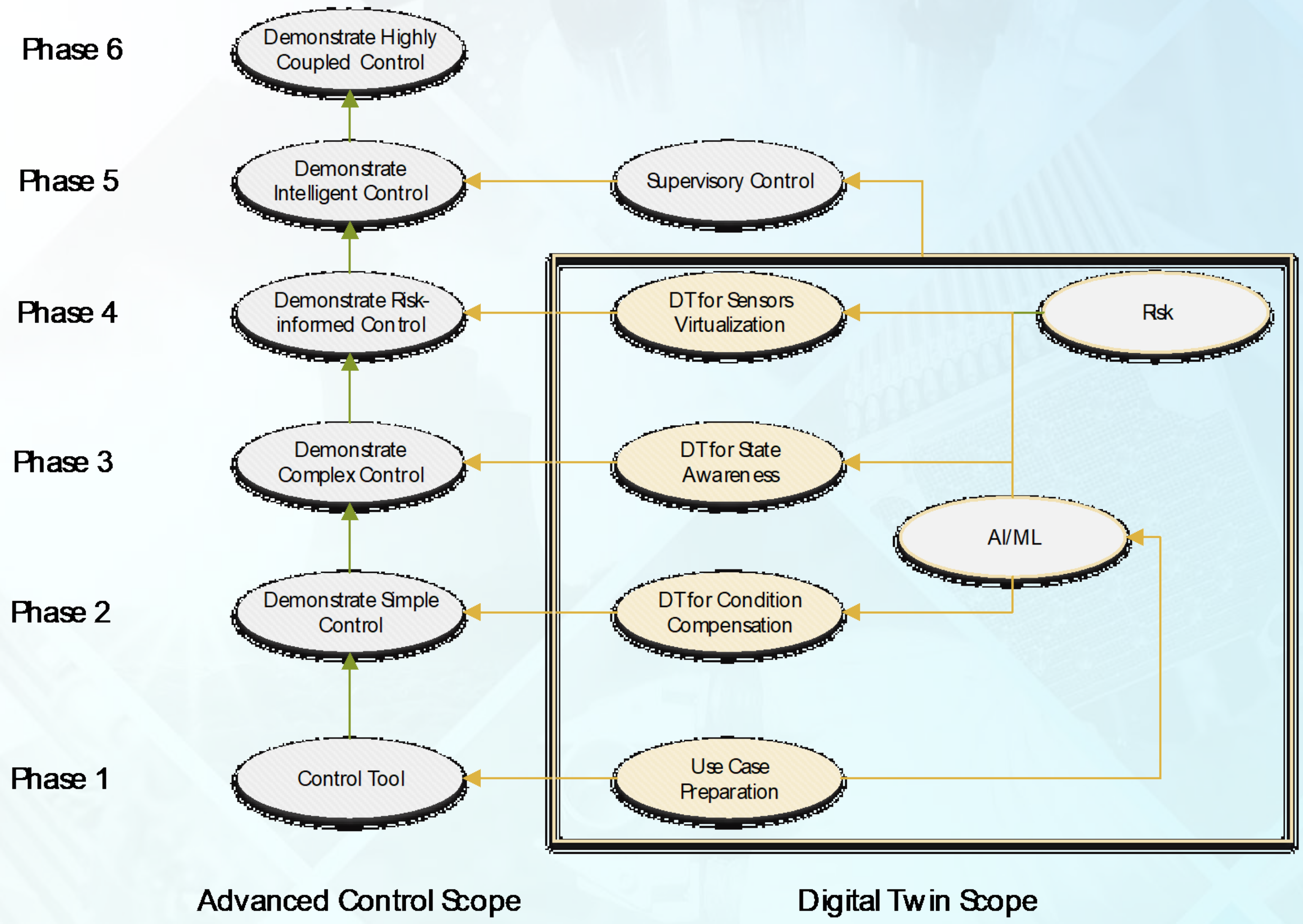
Our previous effort identified unique aspects and challenges for controlling advanced reactors, which resulted in the proposed set of control system requirements.

Unique Aspect	Challenge	Control Requirement
Regulatory Requirements	Artificial Intelligence/Machine Learning (AI/ML) control may not meet regulatory requirements, such as deterministic and explainable behavior	Include an interface control layer between the plant and any AI/ML decision making
Operating Environment	Instrumentation and Control (I&C) equipment will endure harsh environments for extended periods, increasing probabilities of failures	Identify and compensate for sensor, communication, and electronics failures
High Consequence	Manual investigation to reduce uncertainty and avoid shutdown may not be feasible	Incorporate risk elements to prevent unnecessary loss of power generation
Highly Coupled	Compact and simpler designs will produce strongly coupled systems, making "isolated" control less feasible	Integrate highly coupled control loops and state-awareness methods
Evolving Knowledge	Novel concepts of physics and operation will be used that may not be fully understood or validated	Incorporate robustness into the control loop design
Operating History	There will be limited operating history with which to make operational decisions	Use software models to identify and react to or track unanticipated physical phenomena Define the human role and allowable human interventions

To meet these control requirements, that effort also proposed a hierarchical control framework that integrates advanced control, supervisory control, and digital twins.



Our future research plan incrementally incorporates additional capabilities until we ultimately demonstrate highly coupled and autonomous operations.

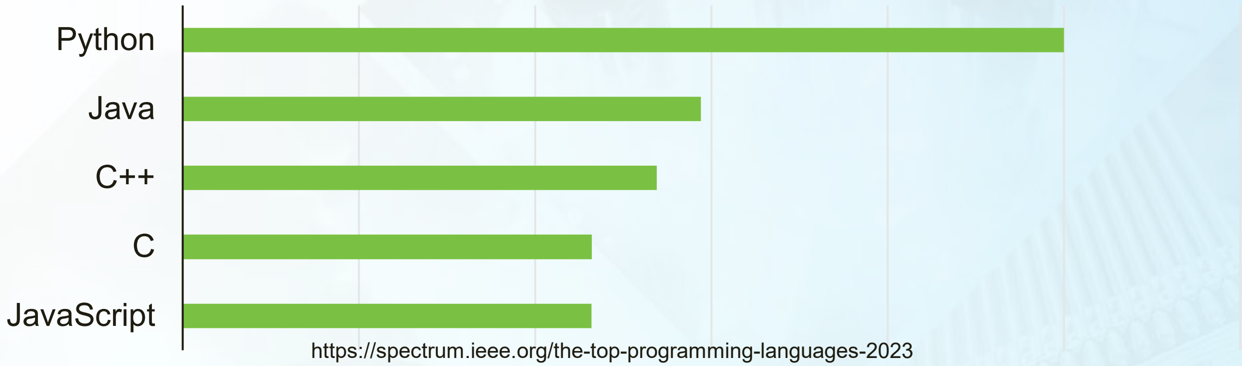


Motivation and Objectives

- To integrate the framework, existing software tools surveyed are proprietary and contain closed-source code, are unqualified or limited in functionality, have limited interfacing to other tools, and/or require specialized knowledge of tools
- This effort created a simulation platform that can integrate autonomous-control-enabling technologies and methods, allowing for faster and more efficient development and transfer of ideas
- The software platform is called the Control and Optimization Modular Modeling Application for Nuclear Deployment (COMMAND), and it is designed to be:
  - Accessible
  - Modular and interfaceable
  - High performing

COMMAND was developed using the Python programming language because of its popularity, active community, and open-source and cross-platform compatibility.

IEEE Top Programming Languages 2023



https://spectrum.ieee.org/the-top-programming-languages-2023

Popularity

Open Source

open source initiative®

python™

Active Community

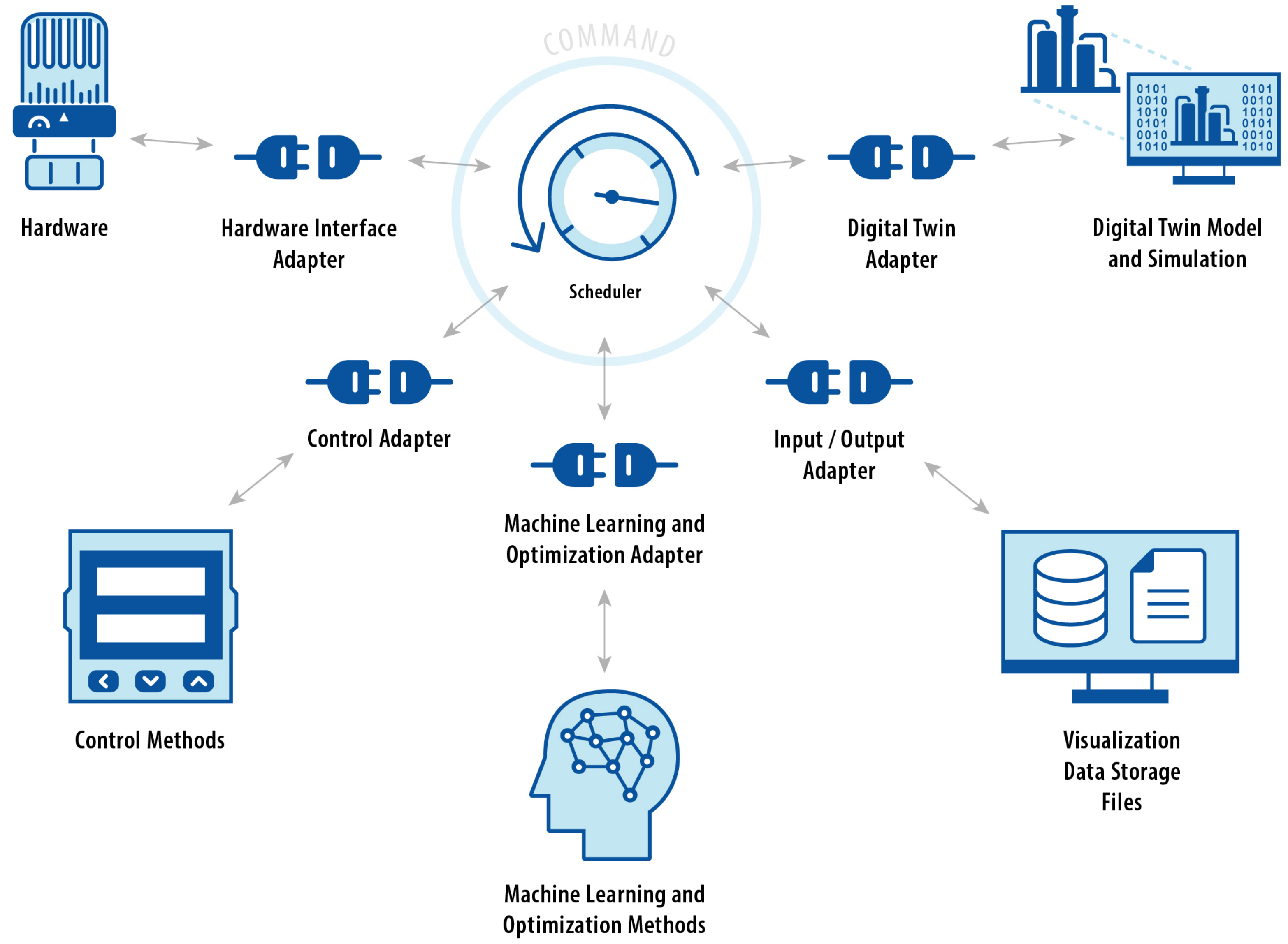
NumPy

pandas

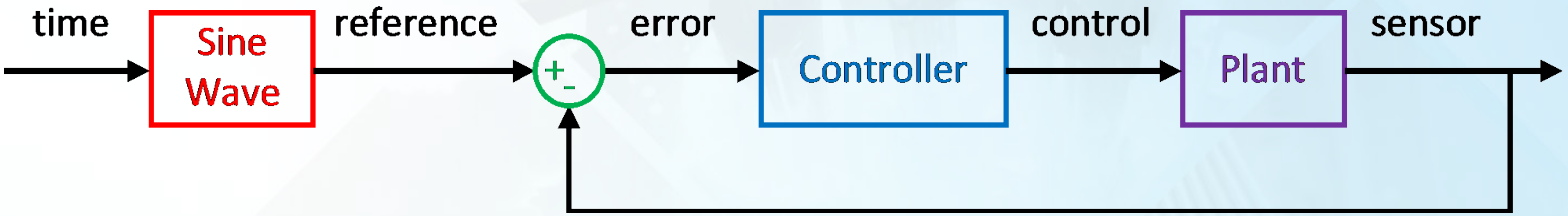
TensorFlow

Cross Platform

Graphical Layout



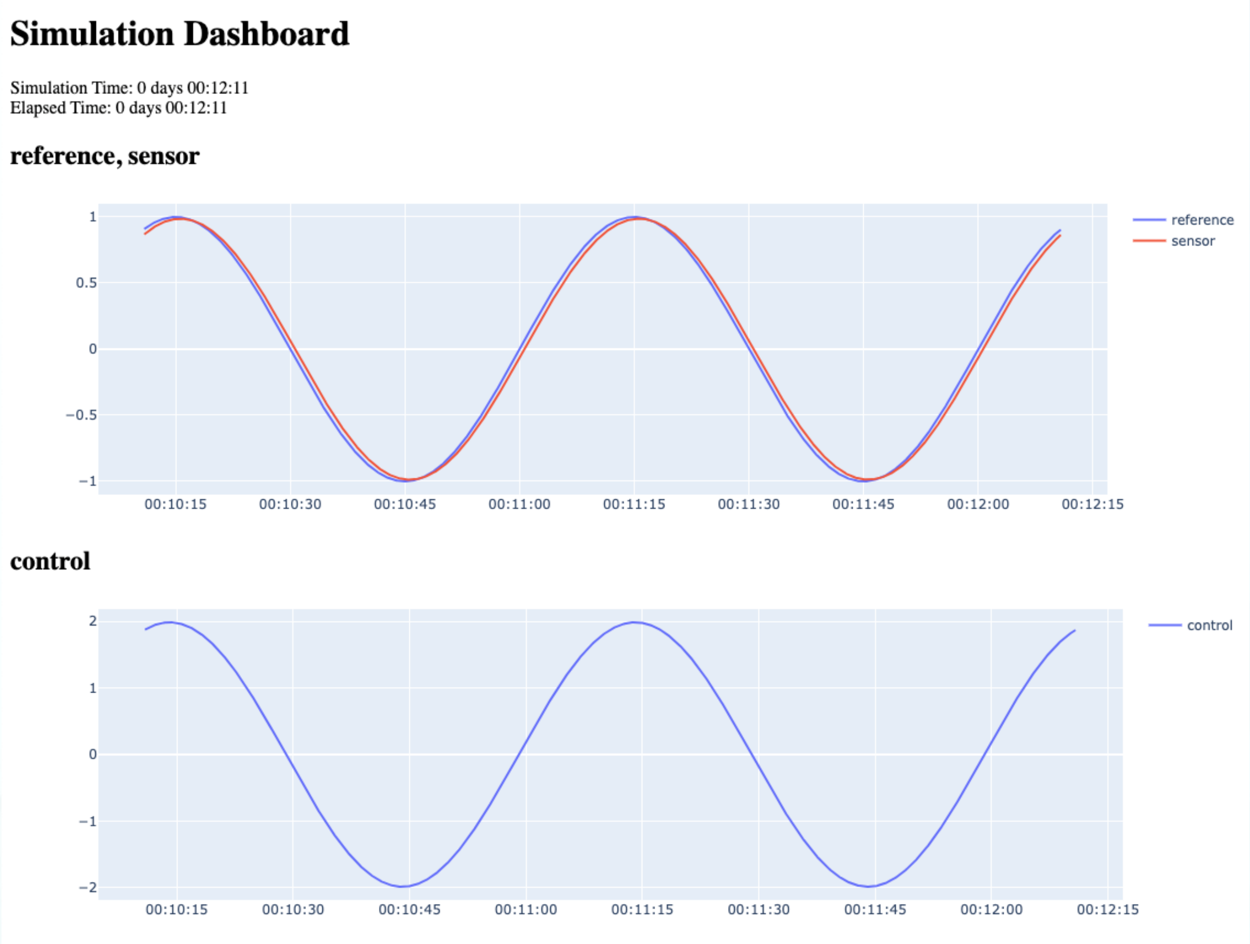
To create a simulation, the basic building blocks—variables and systems—are combined in a Python script, with connections defined by system inputs and outputs.




```
from command import *
if __name__ == '__main__':
    s = base.Simulator(time_step=0.1)
    s.add([
        base.Variables('reference', 'error', 'control', 'sensor'),
        base.SineWave(name='sine1', inputs='sim_time', outputs='reference', period=60),
        base.Subtraction(name='subtraction1', inputs1='reference', inputs2='sensor', outputs='error'),
        control.PIDController(name='pid1', inputs='error', outputs='control', kp=10, ki=3, kd=0, tau=0),
        control.TransferFunction(name='tf1', inputs='control', outputs='sensor', tf_matrices=[[1], [0.5, 1]]),
        io.Historian(name='historian', path='sensor_data.csv'),
        io.Visualization(name='visualization', plot_dicts=[
            {'variables': ['reference', 'sensor']},
            {'variables': ['control']},
        ])
    ])
    s.compile()
    s.start()
```

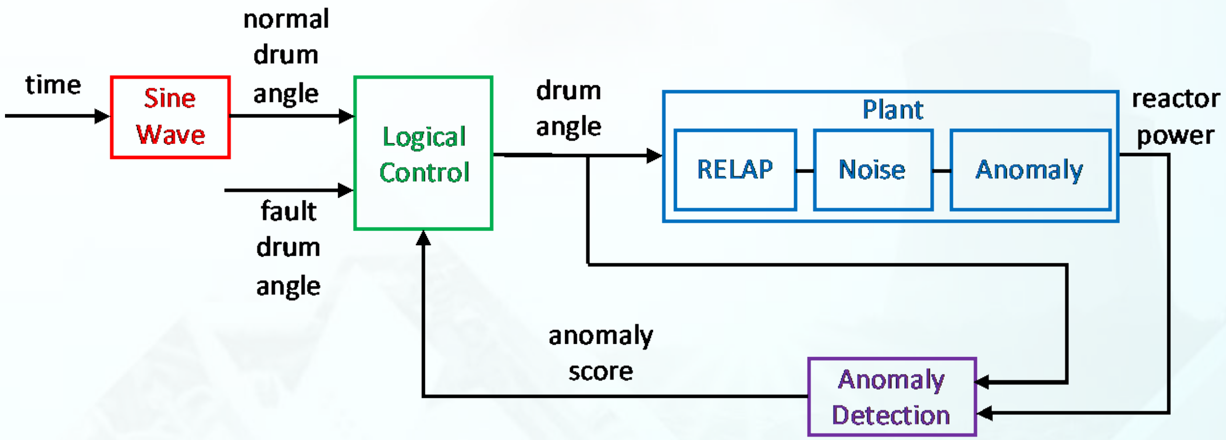
- Uses text input and output files
- Groups components by modules
- Connects systems through their inputs and outputs keywords
- Allows simulation components to be added in any order
- Enables parallel processing

Running the simulation creates both a visualization platform for viewing its progression in real-time and a CSV output file for reporting and postprocessing.



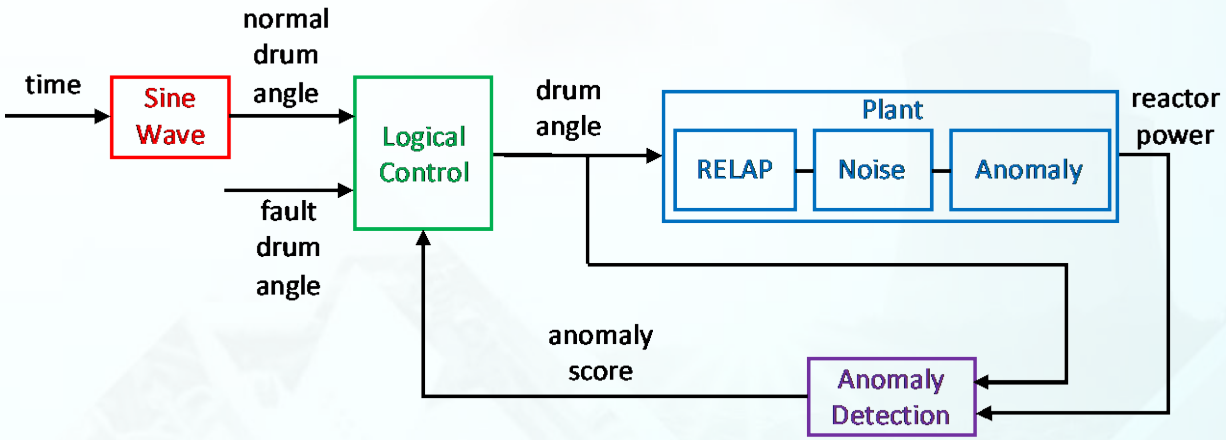
sim_time	control	error	reference	sensor
0	0.0000	0.0000	0.0000	0.0000
0.1	0.1047	0.0105	0.0105	0.0000
0.2	0.2078	0.0205	0.0209	0.0005
0.3	0.3015	0.0292	0.0314	0.0022
0.4	0.3805	0.0362	0.0419	0.0056
0.5	0.4418	0.0413	0.0523	0.0111

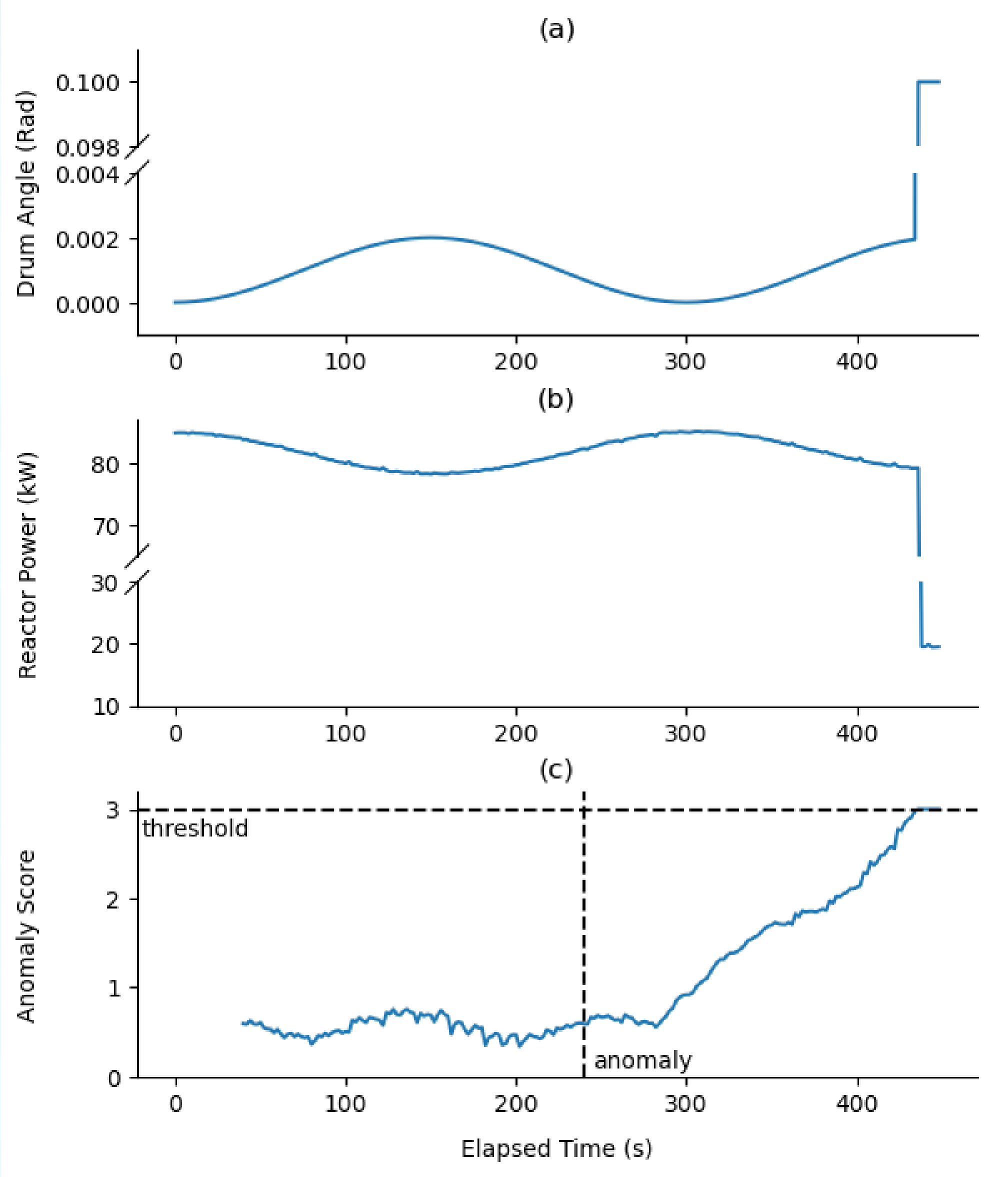
We implemented a use case that integrated RELAP5-3D, logical control, an anomaly, and anomaly detection to simulate detecting the anomaly and shutting the reactor down.




```
import numpy as np
from command import *

def scaled_cosine(x):
    return 0.001 * (1 - np.cos(x * 2 * np.pi / 300))

if __name__ == '__main__':

    input_variables = [{'card_number': 20509670, 'word_number': 3, 'reference_value': 0.5, 'offset': 0.5}]
    output_variables = [{'location': 0, 'variable': 'rktpow']

    s = base.Simulator(time_step=2)
    s.add([
        base.Variables('normal_drum_angle', 'drum_angle', 'anomaly_score',
            'reactor_power1', 'reactor_power2', 'reactor_power'),
        base.Constant('score_threshold', value=3),
        base.Constant('fault_drum_angle', value=0.1),
        base.UserDefined(name='user1', inputs='sim_time', outputs='normal_drum_angle', function=scaled_cosine),
        control.LogicalOperator(name='logic1', comp1='anomaly_score', op='<=', comp2='score_threshold',
            return1='normal_drum_angle', return2='fault_drum_angle', outputs='drum_angle'),
        dt.RELAP53D(name='marvel', inputs='drum_angle', outputs='reactor_power1', relap_file_directory='RELAP/',
            input_variables=input_variables, output_variables=output_variables),
        base.AddGaussianNoise(name='noise1', inputs='reactor_power1', outputs='reactor_power2', std=50),
        base.AddRamp(name='anomaly1', inputs='reactor_power2', outputs='reactor_power',
            start_time=4*60, slope=200/60),
        mlo.AnomalyDetector(name='detector1', inputs=['drum_angle', 'reactor_power'], outputs='anomaly_score',
            historical_data_file='unfaulted_data.csv', method='pca', pca_var=0.99, window_size=20),
        io.Historian(name='historian', path='faulted_data.csv', time_step=10),
        io.Visualization(name='visualization', update_time=1, n_time_steps=200, plot_dicts=[
            {'variables': ['drum_angle']},
            {'variables': ['reactor_power']},
            {'variables': ['anomaly_score']}
        ]),
    ])
    s.compile()
    s.start()
```

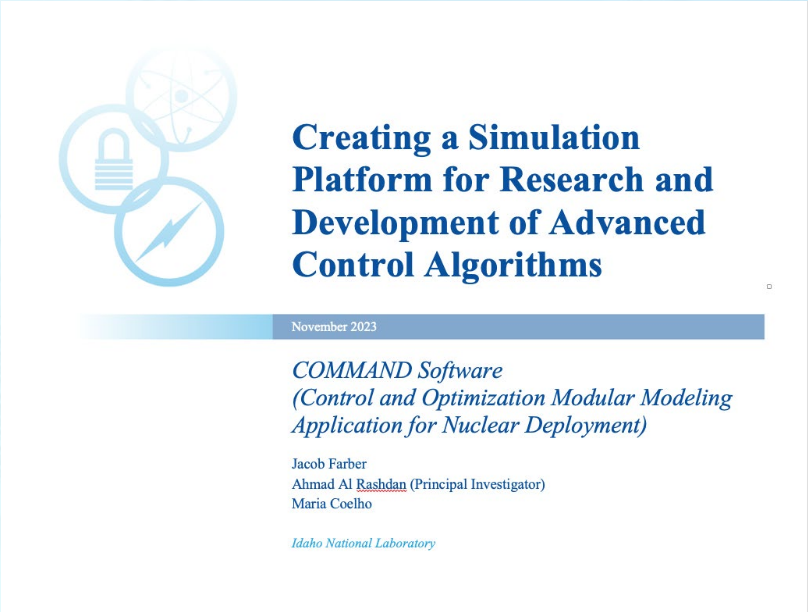
We implemented a use case that integrated RELAP5-3D, logical control, an anomaly, and anomaly detection to simulate shutting the reactor down after detecting an anomaly.





Conclusions

- Filling a gap in the existing software tools, COMMAND is a flexible and scalable simulation software platform written in Python
- Its key benefit is its ability to integrate autonomous-control-enabling technologies and algorithms, allowing for faster and more efficient development and transfer of ideas
- The current effort laid down the software foundations and infrastructure and demonstrated the software with the MARVEL use case


Creating a Simulation Platform for Research and Development of Advanced Control Algorithms

November 2023

COMMAND Software
(Control and Optimization Modular Modeling Application for Nuclear Deployment)

Jacob Farber
Ahmad Al Rashdan (Principal Investigator)
Maria Coelho

Idaho National Laboratory

Thank You