Automated Tuning of Generic Embedded Controllers using Multi-objective Bayesian Optimization

Kimberly J. Chan[1], Joel A. Paulson[2], and Ali Mesbah[1]

[1]University of California, Berkeley
[2]The Ohio State University

June 3, 2022
4th NorCal Control Workshop

Berkeley
UNIVERSITY OF CALIFORNIA

Background & Motivation

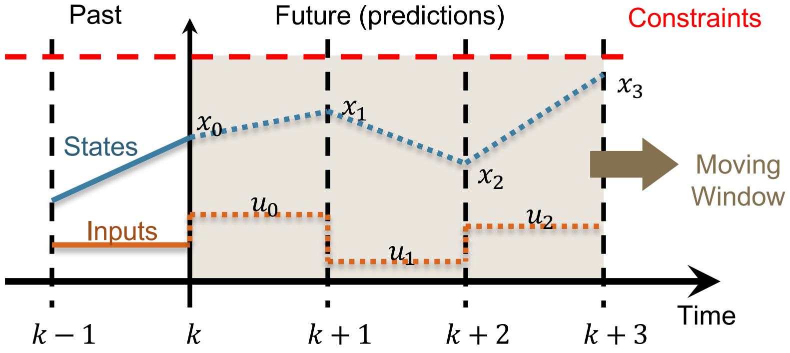
Qualities of Emerging Applications

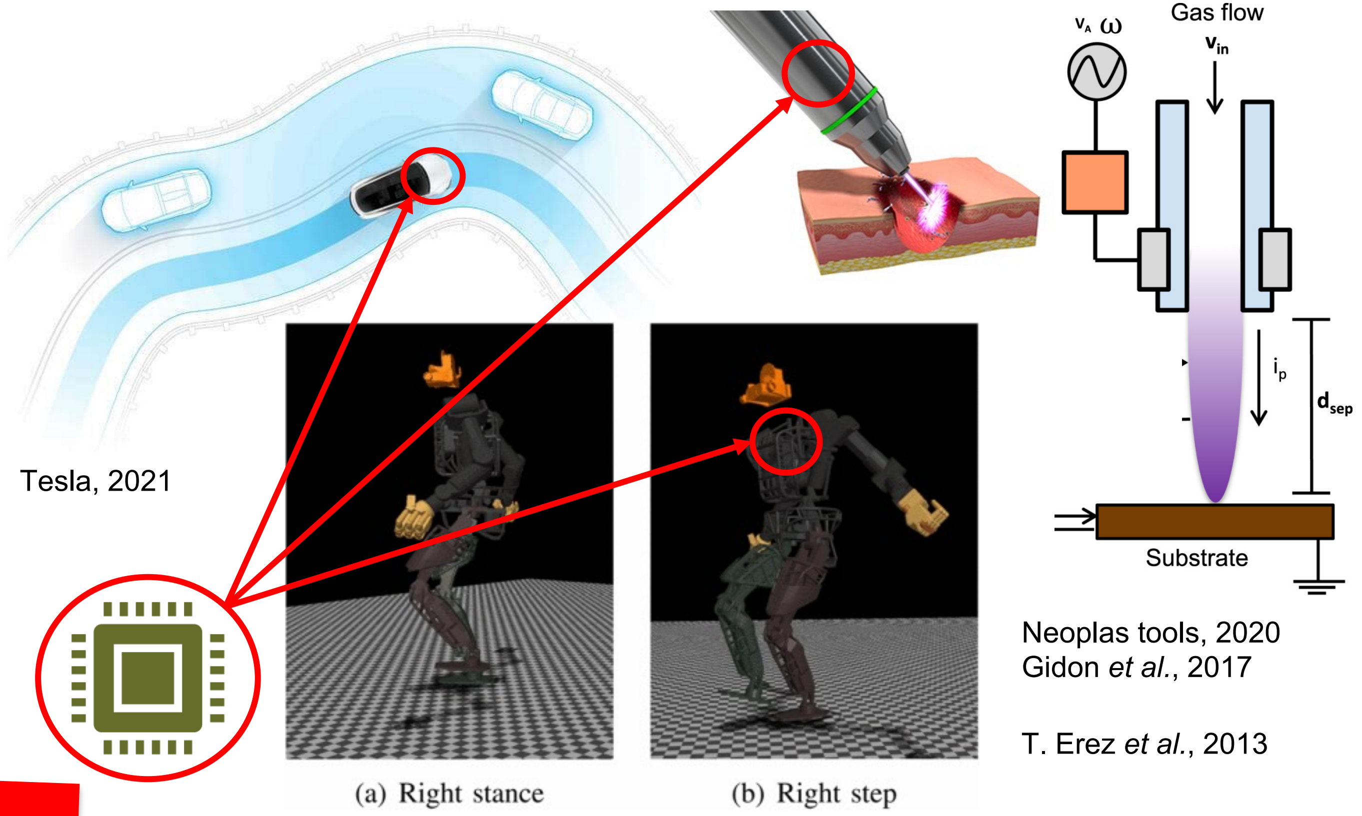
- Large-scale
- Highly nonlinear
- Fast dynamics
- Safety-critical

Challenges of Complex Control on Embedded Systems

- Real-time computation
- Resource-limited hardware
- Numerical robustness at low computational accuracy

Embedded Controller Design is a Multi-step Process!





How can the process of converting a **complex** controller to a **generic** controller that can be implemented on an **embedded** device be systematically optimized?

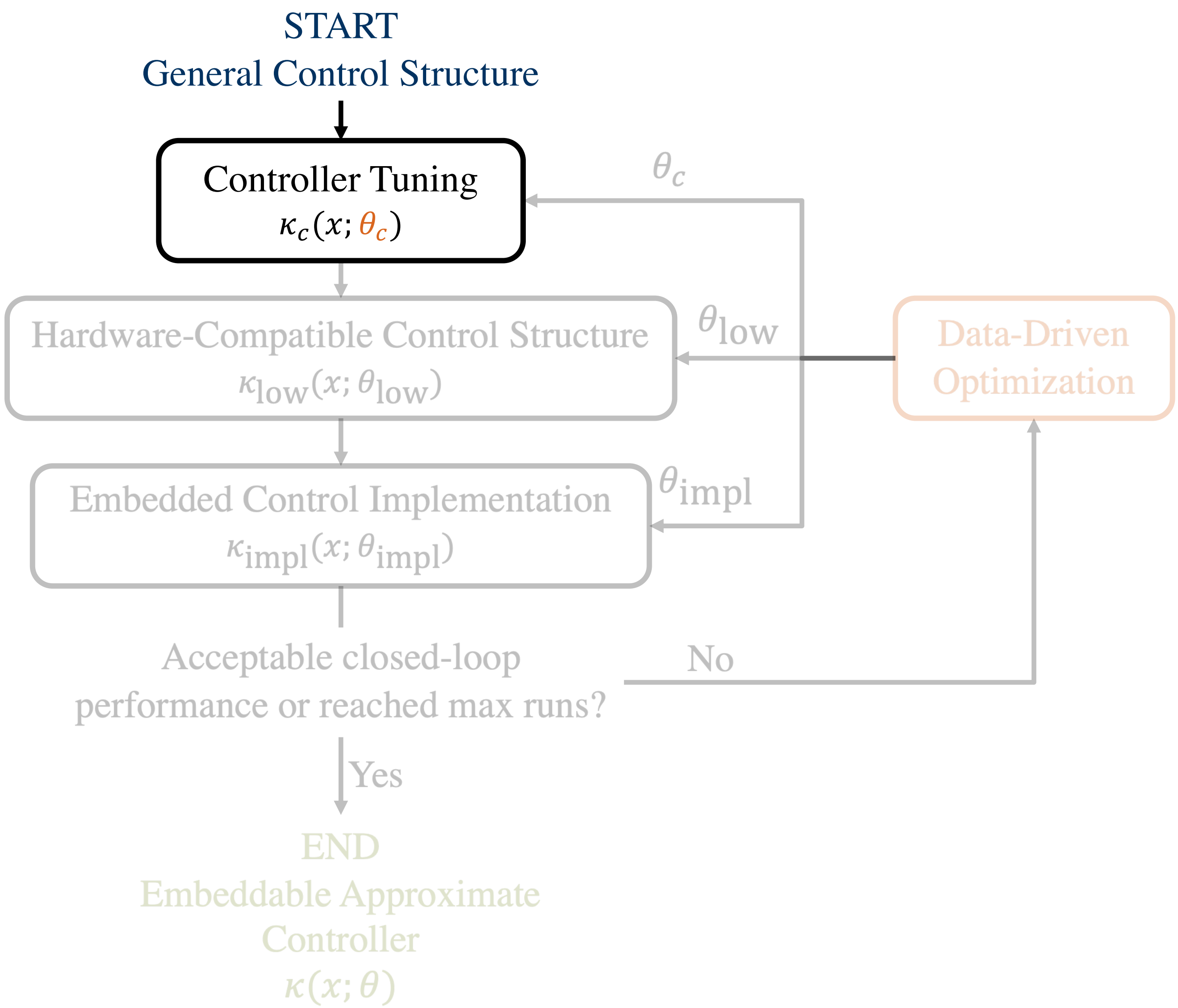
The Framework: An Assumed Workflow

Complex Controller Tuning

- Any given controller has **parameters** which can be tuned to change controller **performance**
- E.g.,
  - PID → gains and time constants
  - MPC → prediction horizon, cost weighting, etc.
  - Robust controller → uncertainty models

Control Law

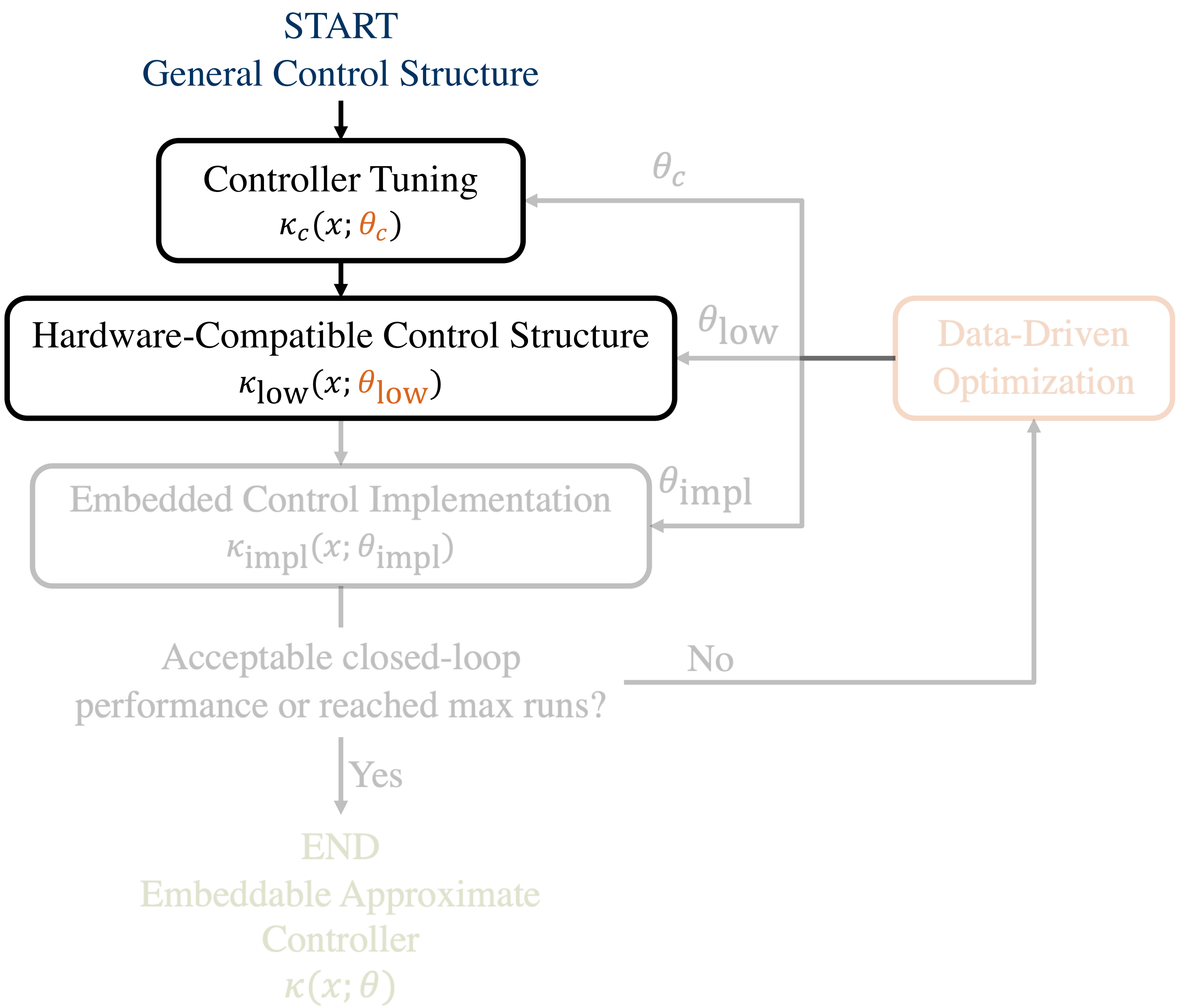
$\kappa(x;\theta)$



The Framework, *continued*

Generating a hardware-compatible controller

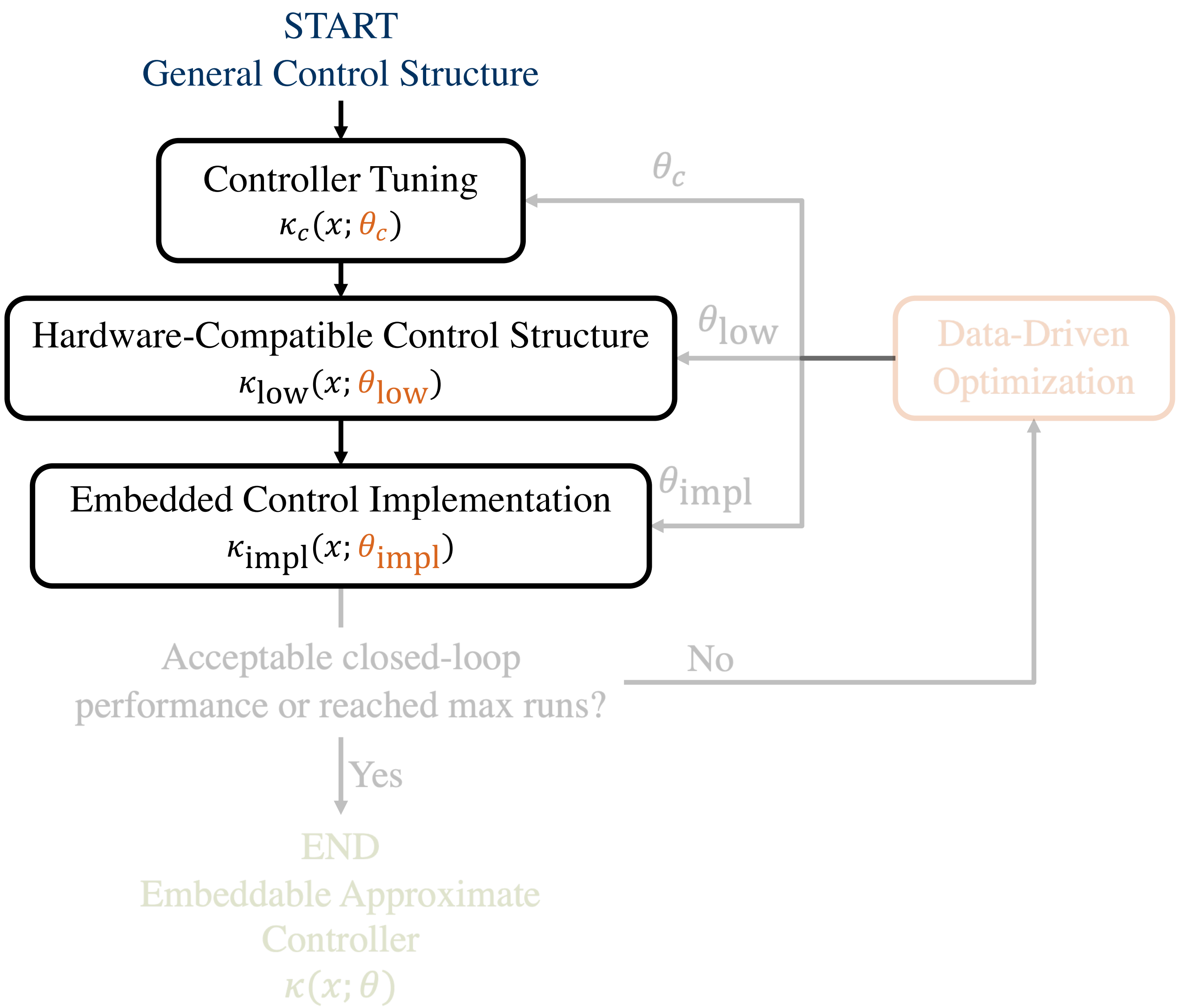
- Generic controllers are desired
- As an *intermediary*, transformation of the controller to a simple-to-evaluate controller by **approximation** is advantageous for hardware implementation
  - Simple operations
  - High accuracy with low-resource numerics



The Framework, *continued*

Implementing controllers on specialized hardware

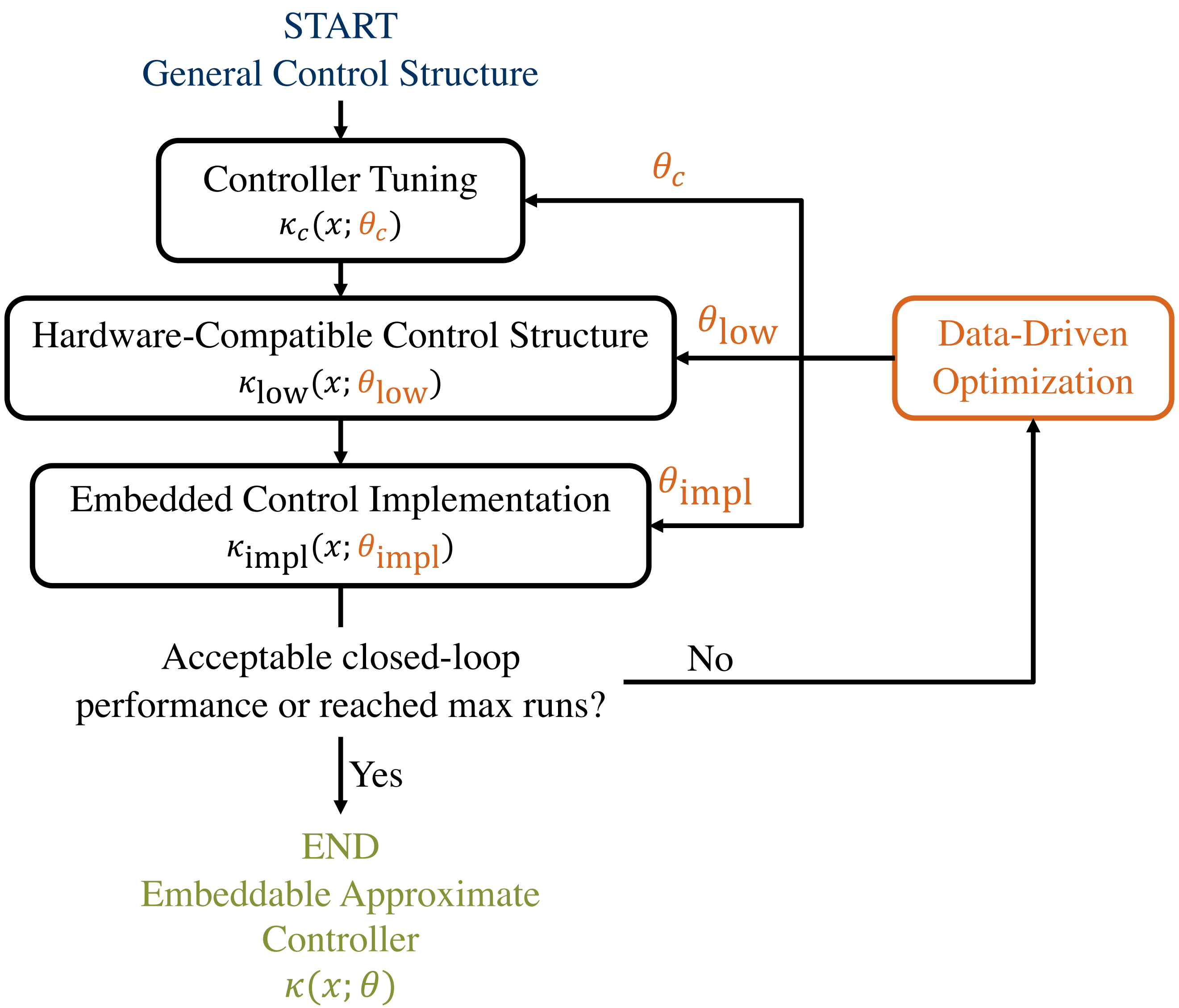
- Implementation on hardware can be difficult/require specialized training
- Automation of the process is aided by **pre-defining** the simple control law (previous step) and by strides in simplifying the hardware design process (automated code generation, high level synthesis)



The Framework, *continued*

Data-driven Optimization

- Key Idea: select **design parameters** to provide **end-to-end** controller design such that the end product is a *well-performing embeddable approximate controller*
- How to define a "well-performing" controller? → **closed-loop metrics**
- Which parameters to optimize?
- How do we do so intelligently? i.e., with limited data and/or closed-loop runs



What to use for “Data-driven Optimization”?

Bayesian Optimization

- Qualities:
  - Derivative-free/black-box
  - Can escape local optimums
  - Evaluation efficient
  - Handles noisy measurements

Assumptions on $\phi(\cdot)$:

- Difficult and/or costly to evaluate
- Not necessarily differentiable
- Noisy

$$\min_{\theta \in \Theta} \{\phi_1(\theta), \ldots, \phi_M(\theta)\}$$

*θ can consist of continuous, discrete, and/or categorical parameters!

- Method:
  - Create **surrogate model(s)** of the metrics $\phi(\cdot)$ using some initial observations

$$\mathcal{D}_n = \{(\theta_i, \{\phi_m(\theta_i)\}_{m=1}^M)\}_{i=1}^n$$

  - Optimize an **acquisition function** $\alpha(\cdot)$ to select the next sample

$$\theta_{n+1} = \arg\max_{\theta \in \Theta} \alpha_n(\theta; \mathcal{D}_n)$$

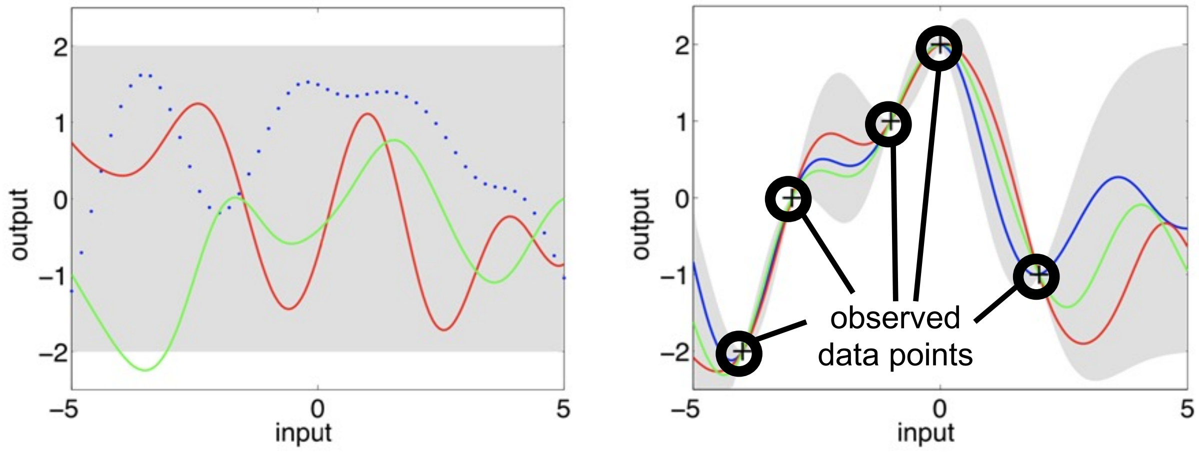
  - Common AFs: Expected Improvement (EI), Upper Confidence Bound (UCB)

Shahriari *et al.*, 2017; Frazier, 2018.

Bayesian Optimization Components

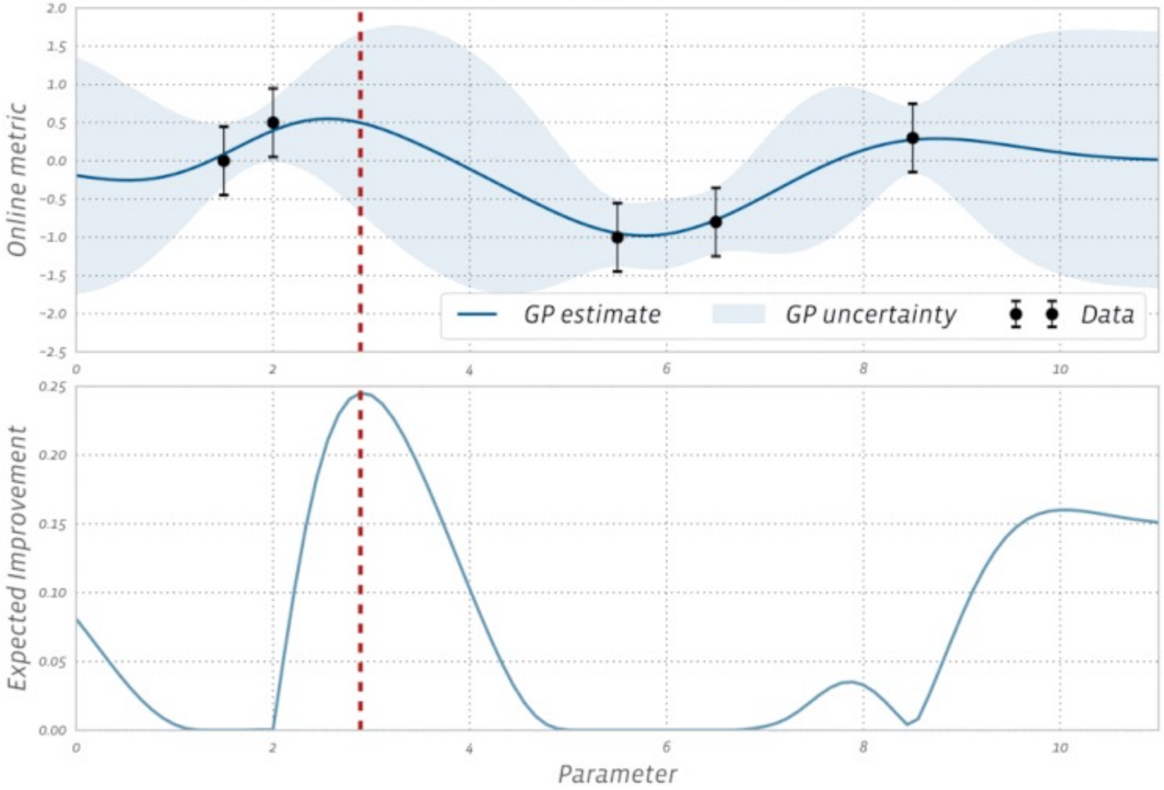
Surrogate Modeling: Gaussian Process (GP) Regression

- Learn function/metric $\phi(\cdot)$ with error bars from data
- GP is a *prior over functions*, which is updated using Bayes' theorem



Acquisition Function (AF)

- Balances sampling around current optimum and reducing uncertainty in other regions → tradeoff between exploration and exploitation

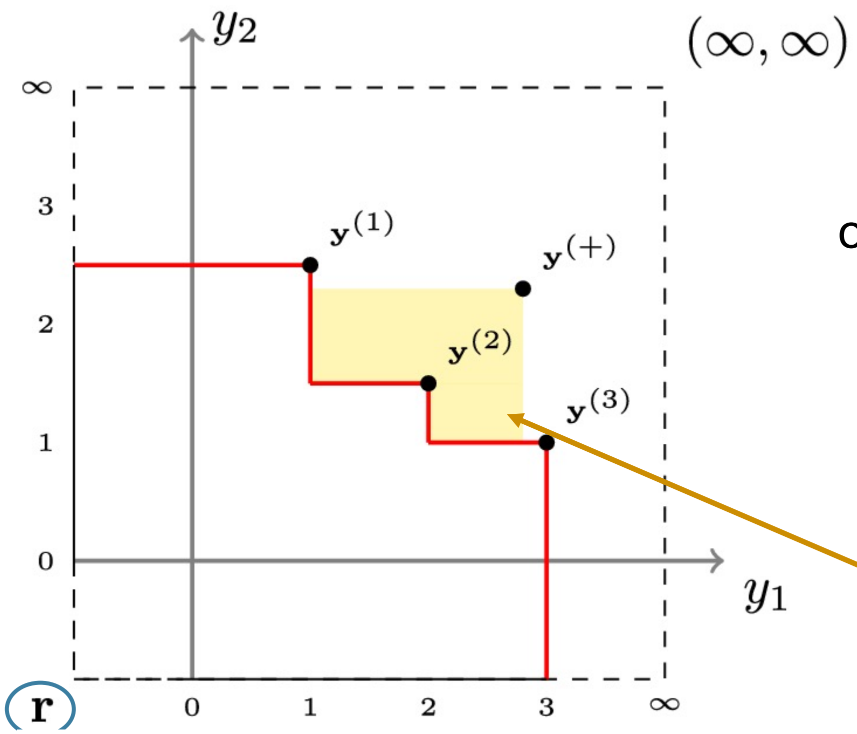


Rasmussen and Williams, 2006.

Multi-objective Bayesian Optimization

Find the *Pareto optimal solutions* that form the **Pareto frontier** (a boundary in which improvement in one objective means degrading the other(s))

- Common competing objectives: performance vs robustness vs resource utilization
- Multi-objective AFs: Expected Hypervolume Improvement (EHVI)

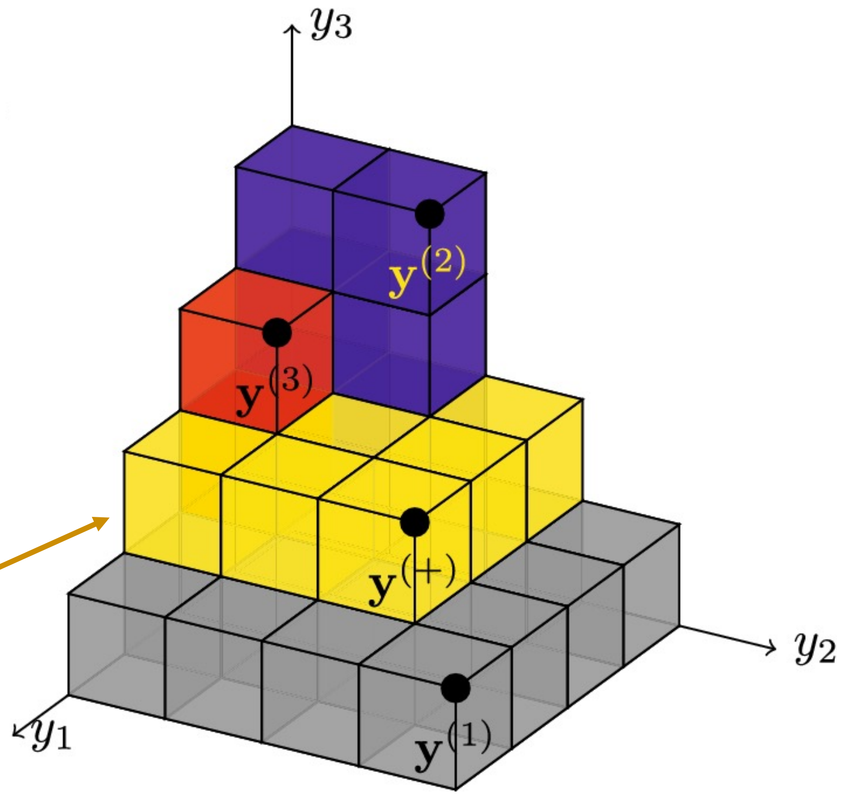


Reference point

Currently observed Pareto optimal points: $y^{(1)}$, $y^{(2)}$, $y^{(3)}$

$y^{(+)}$ newly observed point

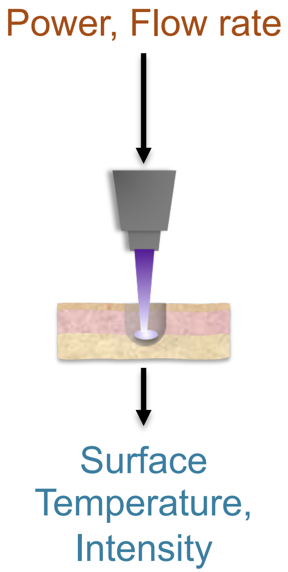
Yellow regions indicate **hypervolume improvement**

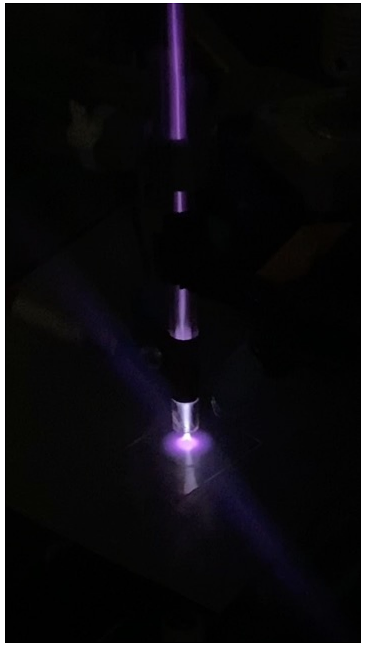


Khan, Goldberg, and Pelikan, 2002
Yang, Emmerich, Deutz, and Bäck, 2019

*diagrams assume maximization

Test Case: Thermal Dose Delivery of Cold Plasma





Thermal Dose Definition

- Equivalent response of biological substrates subjected for a given amount of time to a thermal stress at a reference temperature

$$\mathrm{CEM}(t) = \int_0^t \underbrace{K^{43-T(\tau)} dt}_{\text{Highly nonlinear}}$$

Integral (non-retractable)

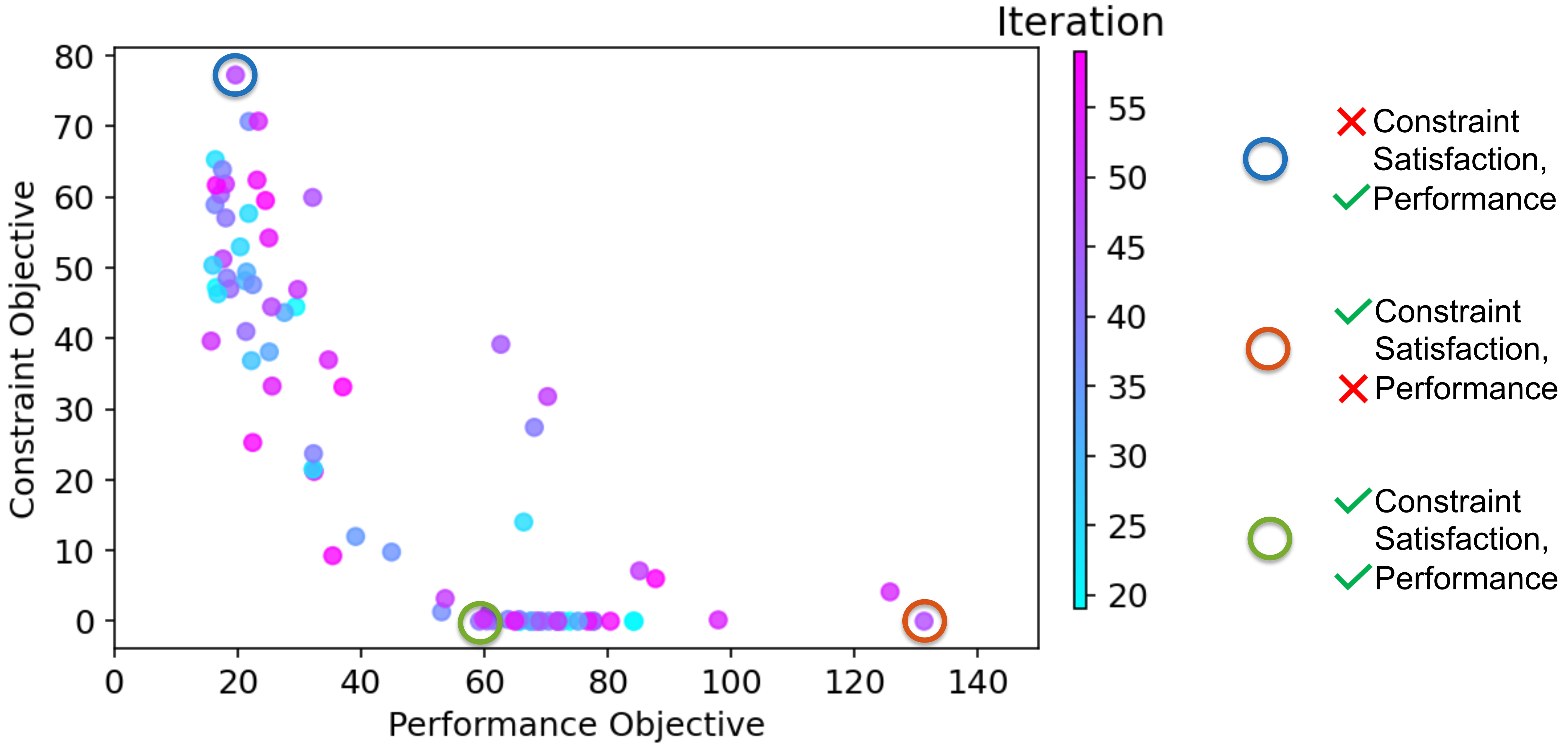
Objectives

- **Performance**: achieve desired dose optimally (quickly)
  - Tracking **cost** (i.e., minimize this)
- **Constraint Satisfaction**: remain below potentially dangerous operating temperature
  - Degree of constraint **violation** (also minimize)

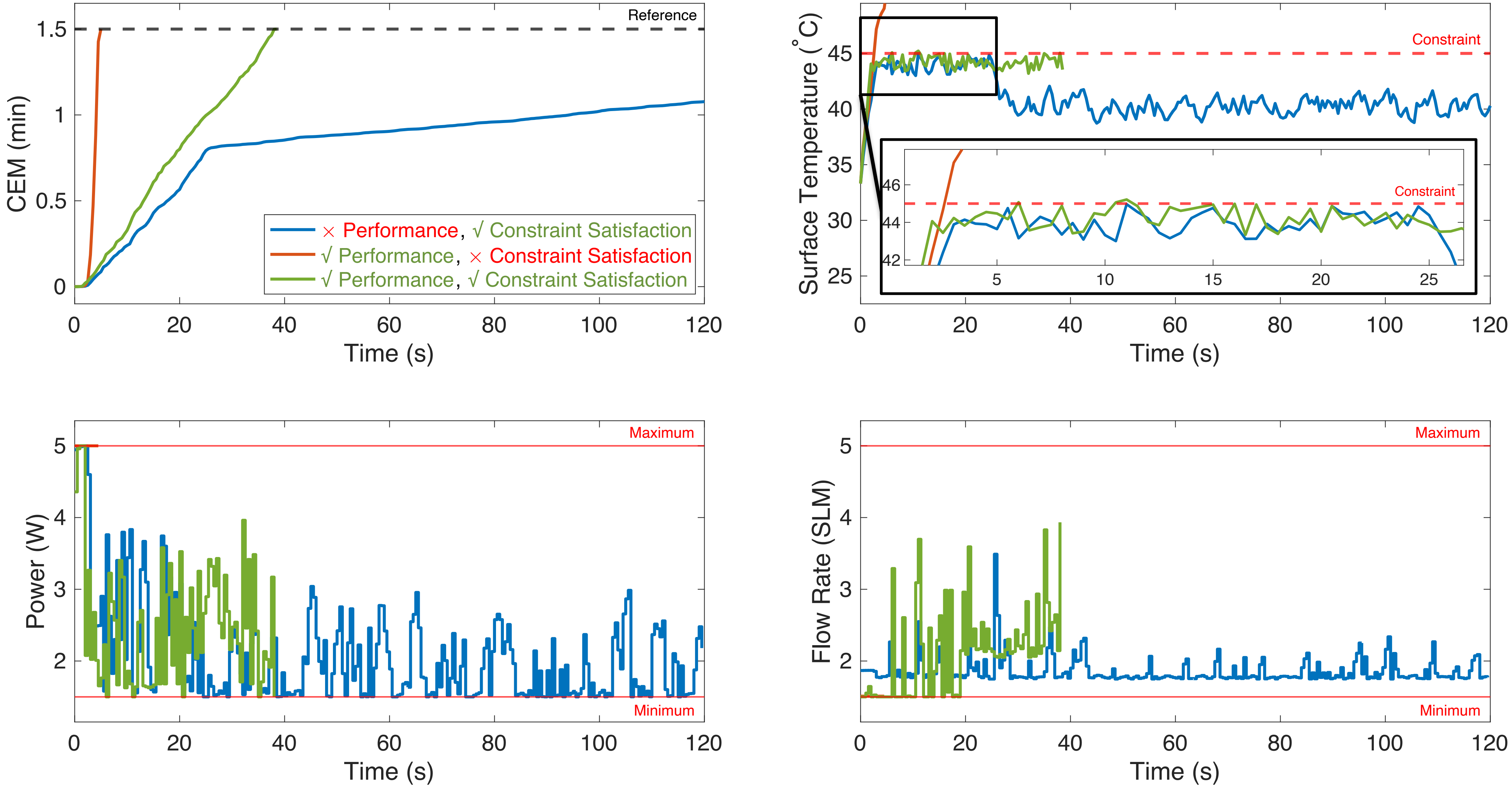
$$\sum_k (\mathrm{CEM}_{sp} - \mathrm{CEM}(k))^2$$

$$\sum_{T(k)>T_{\max}} (T(k) - T_{\max})^2$$

Results: The Pareto Frontier



Results: Closed-loop Simulations



Conclusions + Upcoming and Future Work

- Bayesian optimization is a powerful framework for systematic and automated controller tuning under system uncertainty
- Multi-objective formulation formalizes inherent tradeoffs in controller performance

- Expansion to adaptive control techniques
- Expansion to "robustifying” approximated controllers
- Expansion towards system-on-chip (SoC) architectures
- Explore reinforcement learning techniques for automated controller tuning

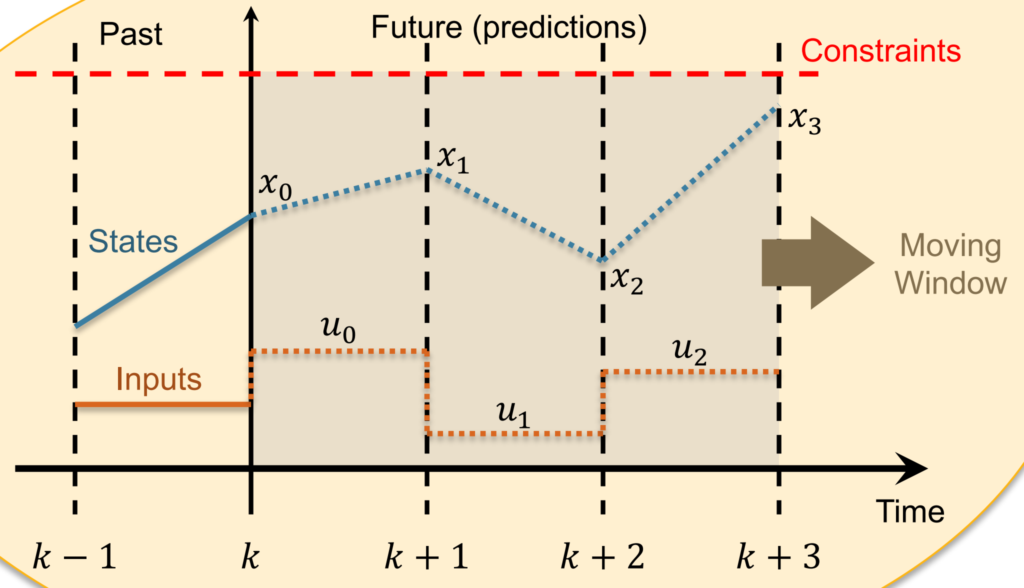
Supplementary

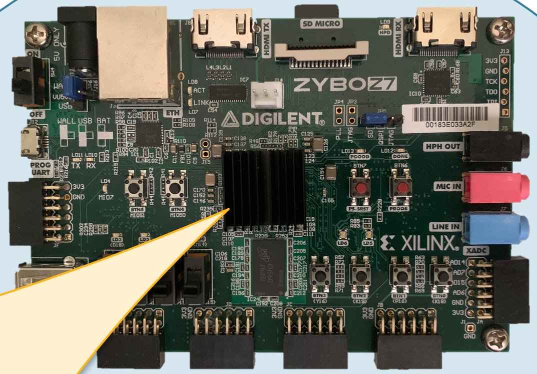
Background & Motivation

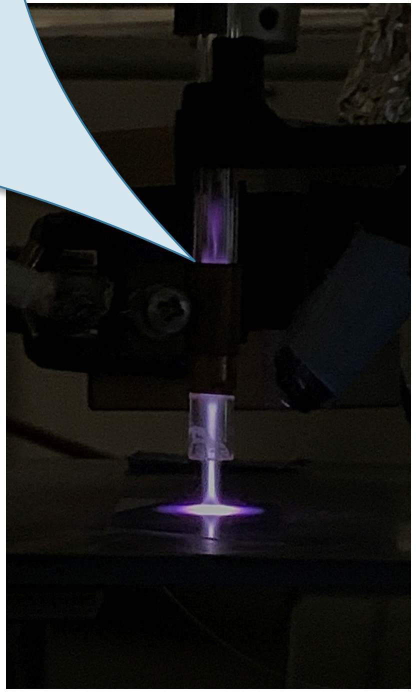
- Embedded control is necessary for a variety of emerging applications
- Emerging applications are complex and thus need complex control
- Embedded control relies on low-resource hardware and/or specialized hardware (e.g., GPUs)
- The process of designing a controller and then embedding it involves a variety of steps that changes the initial complex-form controller (highly accurate, computationally heavy) to a low-level embeddable controller (low-load, quick-to-evaluate).
- Within the steps involves a variety of controller design parameters, i.e., decisions, that affect the final controller.
- Thus, there is a need for an intelligent strategy to provide an end-to-end means to tune the process from a controller’s algorithmic design to its embedded implementation.

Embedded Controller Design is a Multi-step Process

How can a **generic** controller be optimally **tuned** to be implemented on an **embedded** device?







Results: Closed-loop Simulations