

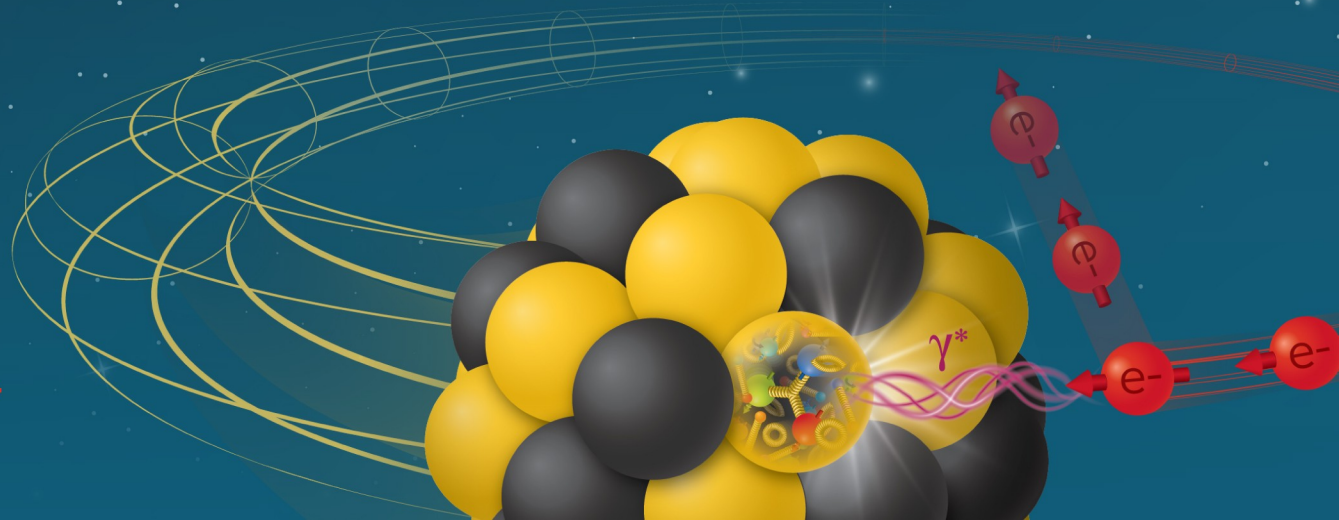
Comparative Analysis Of RF System Configurations:

Optimal Loop Tuning and Automated Loop Setup

Arshdeep Singh on behalf of the EIC LLRF Team

2026-10-13

Electron-Ion Collider



Overview

- Introduction & Overview
 - Motivation, method, evaluation
- Optimization Strategy
- Field Control Optimization
- Operational Application
- Future Work & Conclusions

Introduction/Background

- The physics of a RF system can be modelled fairly well:
 - Resonator dynamics, mechanical modes, LFD, beam interactions [1][2]
- Developed a time-domain simulator in Python. Originally to:
 - Study transient-beam loading and design a One-Turn Delay Feedback
 - Perform power requirement calculations for the EIC
- I became interested in optimal cavity control. Can use the simulation code to study:
 - Alternative feedback configurations
 - Optimal system tuning

Study Overview

- **Motivation**
 - Want to ensure optimal control of the RF feedback systems
 - Desire an automated method of configuring feedback loops while balancing both disturbance rejection and stability of all loops
- **Method**
 - Use numerical minimization techniques to optimize for reference tracking and controller effort
 - Evaluate the performance of different controllers (optimally tuned)
- **Evaluation**
 - Can compare performance based on metrics like reference tracking and disturbance rejection

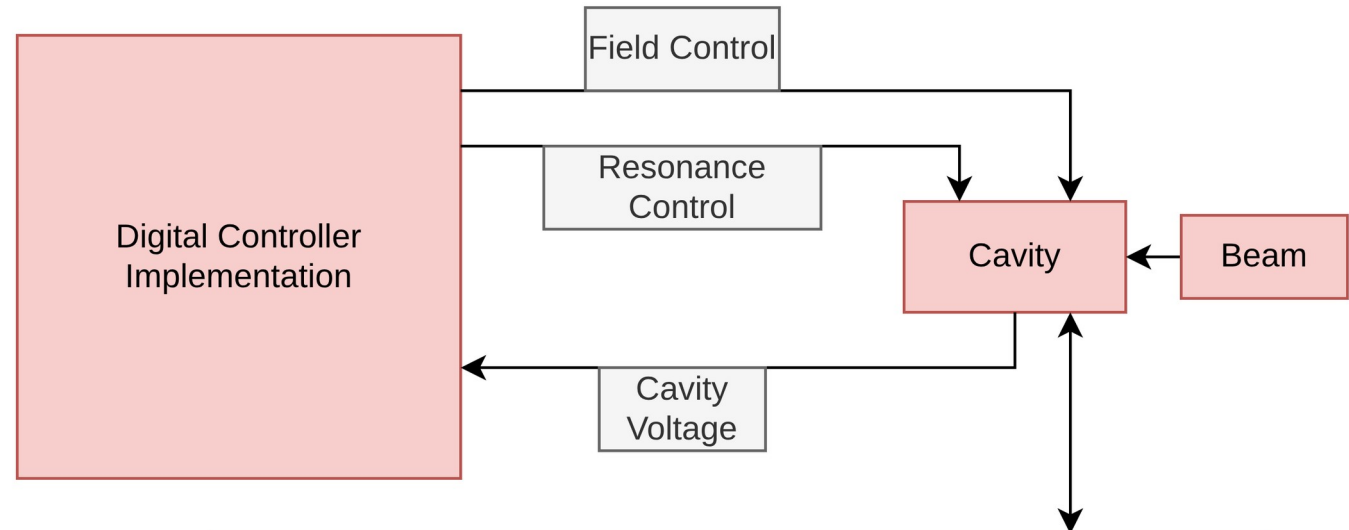
Simulation Code : pyRFControls

- This code is specifically for studying RF control systems, no particle tracking
- Code takes advantage of the Python environment (OOP, NumPy, SciPy...)
 - Components are modelled as classes. Once instantiated, components are connected, similar workflow to Simulink
 - Time domain simulator. At each time instant, signals are updated using component specific methods
 - Supports fixed-point simulation for filter design
 - Single particle dynamics (beam transfer function)
 - Phase noise propagation

```
class CavityModel:
    def __init__(
        self,
        r_q:float,
        q_0:float,
        beta:float,
        w_rf:float,
        dt:float,
        length:float|None = None,
        w_det:float = 0,
        mechanical_modes:list[MechanicalModel]|None = None,
    ):

```

RF System Overview



$$I_{gen} = \frac{V_{ant}}{2(R/Q)} \left(\frac{1}{Q_L} - 2i \frac{\Delta\omega}{\omega} \right) + \frac{dV_{ant}}{dt} \frac{1}{\omega R/Q} + \frac{1}{2} I_{beam}$$

$$\Delta\ddot{\omega}_k + \frac{2}{\tau_k} \Delta\dot{\omega}_k + \Omega_k^2 \Delta\omega_k = -K_k \Omega_k^2 E_{acc}^2 + \mu_{mp}$$

Simulator Includes:

- Cavity electrical + mechanical model
- Tuner model
 - Compares the phase between the forward power and cavity field
- Loop delay

[1][2]

RF System Overview (cont.)

- Simulations are done with a model of the 704 MHz SRF cavity used in the CeCPoP experiment at BNL
- Have measured data of the mechanical modes

Frequency [MHz]	704
R/Q (circuit def.)	250
Q_{ext}	$\sim 1.4\text{e}7$
Q_{L}	$\sim 1\text{e}10$

General Cavity Parameters

Mode Number	1	2	3	4
Mode Frequency (Hz)	91	120	180	270
Mode Q	18.6	10.7	1.6	18
Mode K (Hz/(MV/m) ²)	0.35	0.49	1.4	0.013

Mechanical Mode Parameters

Optimization Strategy

- The loss function is defined as:

$$L(\theta) = \lambda_0 \left| I_{g,ref} - \frac{V_c(\theta)}{2R_L} \right|^2 + \lambda_1 \left| \frac{d\psi_{det}(\theta)}{dt} \right|$$

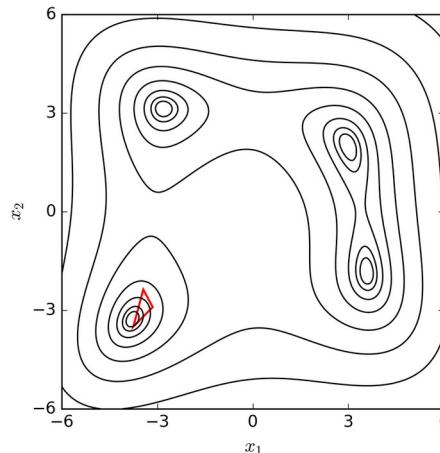
Enforce reference tracking

Smooth tuner response, ensure coupled loop stability

- θ is the vector of tunable parameters
- $I_{g,ref}$ is simply a step function
 - General broadband signal (excites mechanical modes), don't want to overfit to any specific disturbances
- $V_c(\theta)$ is the output of the simulation
- $\Psi_{det}(\theta)$ is the detuning amount
- λ_0 and λ_1 are weighing coefficients

Optimization Strategy (cont.)

- Minimization strategy:
 - Seeking a global minimum for the loss function
 - We are dealing with a non-linear numerical optimization problem with many parameters
 - First utilize a stochastic-evolutionary algorithm (Differential-Evolution) [3]
 - Slow, but effective at landing us in the region of a global solution
 - Then utilize the Nelder-Mead algorithm [4]
 - Improve upon the global solution to find the local minimum
- Both algorithms are conveniently implemented in SciPy!



**Visualization of the Nelder-Mead
algorithm in 2 dimensions
[Wikipedia Article]**

Field Control Optimization

- Options besides the traditional PID exist
 - Active Disturbance Rejection Controller (ADRC), Linear Quadratic Regulator (LQR)

Name	Description	Strengths	Weaknesses
PID	Uses proportional, integral, and derivative action on tracking error	Simple (tuning and stability analysis), low computational cost, can remove steady-state error	Does not use plant dynamics, derivative is noise sensitive, sensitive to IQ coupling
ADRC	Uses an extended-state observer to estimate and eliminate disturbances	Strong disturbance rejection, does not require high model accuracy, removes steady-state error, less sensitive to IQ coupling	More complicated implementation, less direct tuning and stability analysis
LQR	Model-based feedback that minimizes a quadratic cost on state deviation and control effort	Analytic gain calculation, handles IQ coupling by design, guaranteed stability	Requires an accurate linear plant model, model errors can lead to steady-state error

[5][8]

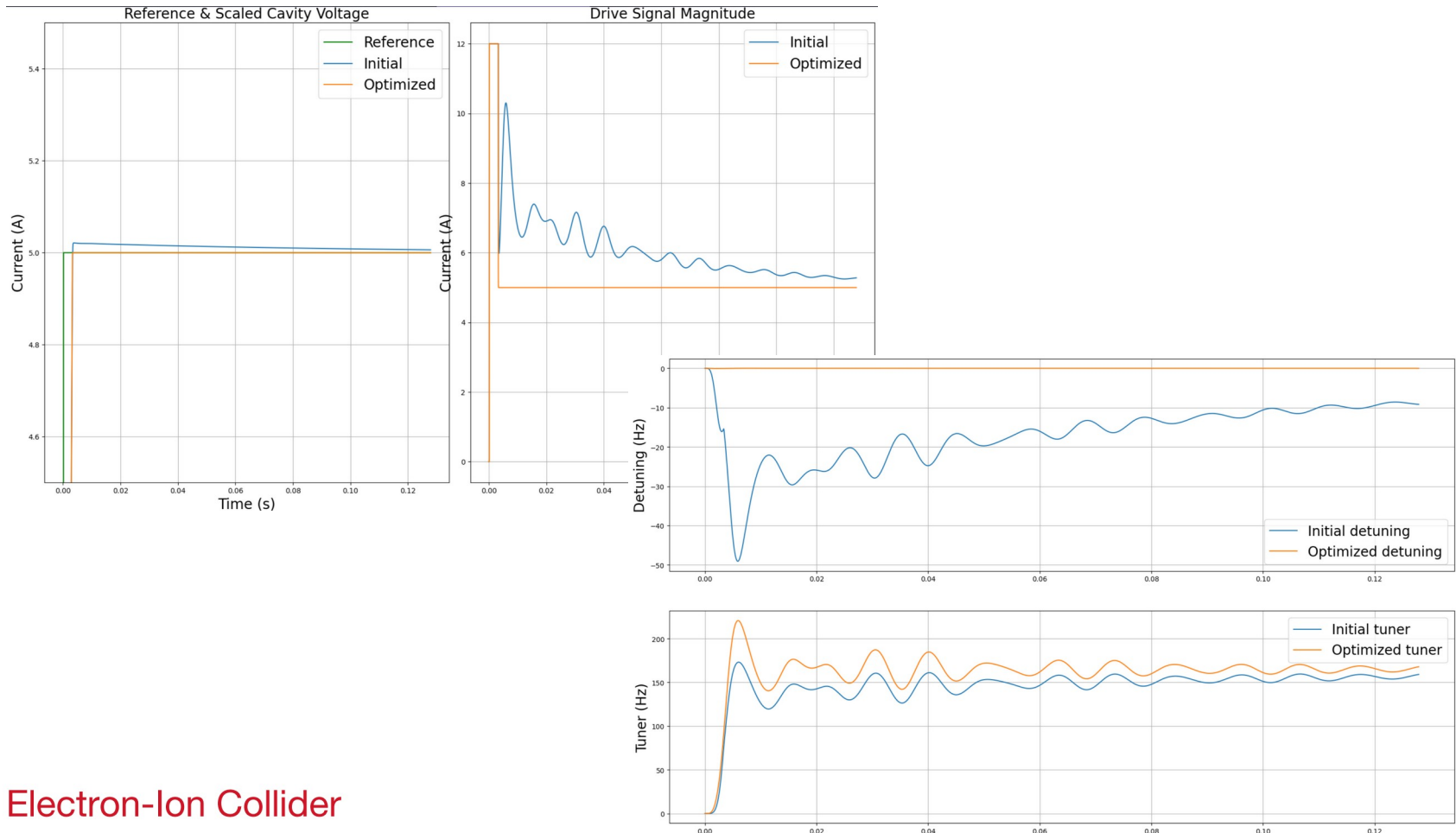
- This study only focuses on the PID and ADRC

Example of Optimization : PID

- Both field and resonance control use PID controllers, thus:

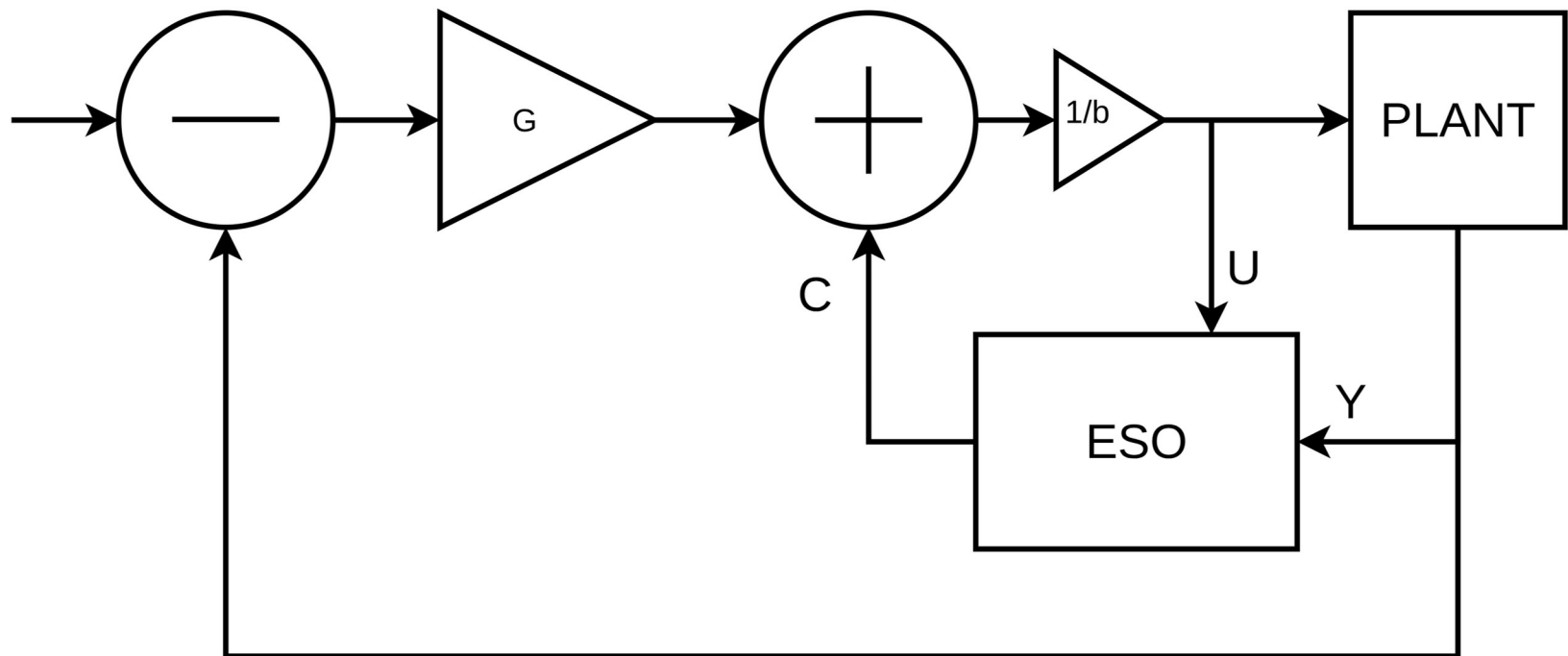
$$\theta = [K_{PF}, K_{IF}, K_{DF}, K_{PR}, K_{IR}, K_{DR}]$$

Field Control
Resonance Control



Overview : ADRC

- An ADRC extends a proportional feedback loop using an extended-state observer (ESO) [5]
 - The ESO calculates disturbances in real time and modules the forward drive accordingly
 - ADRCs generally have better disturbance rejection and stability than PIDs



Developing The ESO

- Following the derivation in [6]
- Start with a well known result, separating the cavity differential equation into real and imaginary parts and assuming $\dot{V}_C \ll \omega_g V_C$

$$\dot{V}_{cR} + \omega_{1/2} V_{cR} + \Delta\omega V_{cI} = \omega_{1/2} V_{gR}$$

$$\dot{V}_{cI} + \omega_{1/2} V_{cI} - \Delta\omega V_{cR} = \omega_{1/2} V_{gI}$$

- First looking at only the real part of V_C , define $y = V_{cR}$, $u = V_{gR}$, $b = \omega_{1/2}$:

$$\dot{y} = f + bu$$

- f is the total disturbance state, including detuning, beam loading, etc.
- Rewriting in matrix form, first we define $x_1 = y$ and $x_2 = f$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \dot{f}$$

Developing The ESO (cont.)

First order plant equation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f$$

$$y = x_1 \quad f = x_2$$

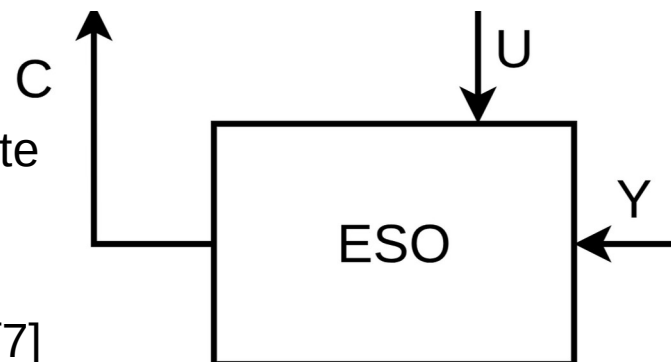
- The goal of the ESO is the eliminate f in real time

First order ESO equation

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u + \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} (y - \hat{y})$$

$$\hat{y} = z_1 \quad \hat{f} = z_2$$

- Estimated disturbance, observer gains
- The ESO uses the drive signal, plant output and an estimate of the plant output to calculate the disturbance
- Fixed point implementation demonstrated in [7]



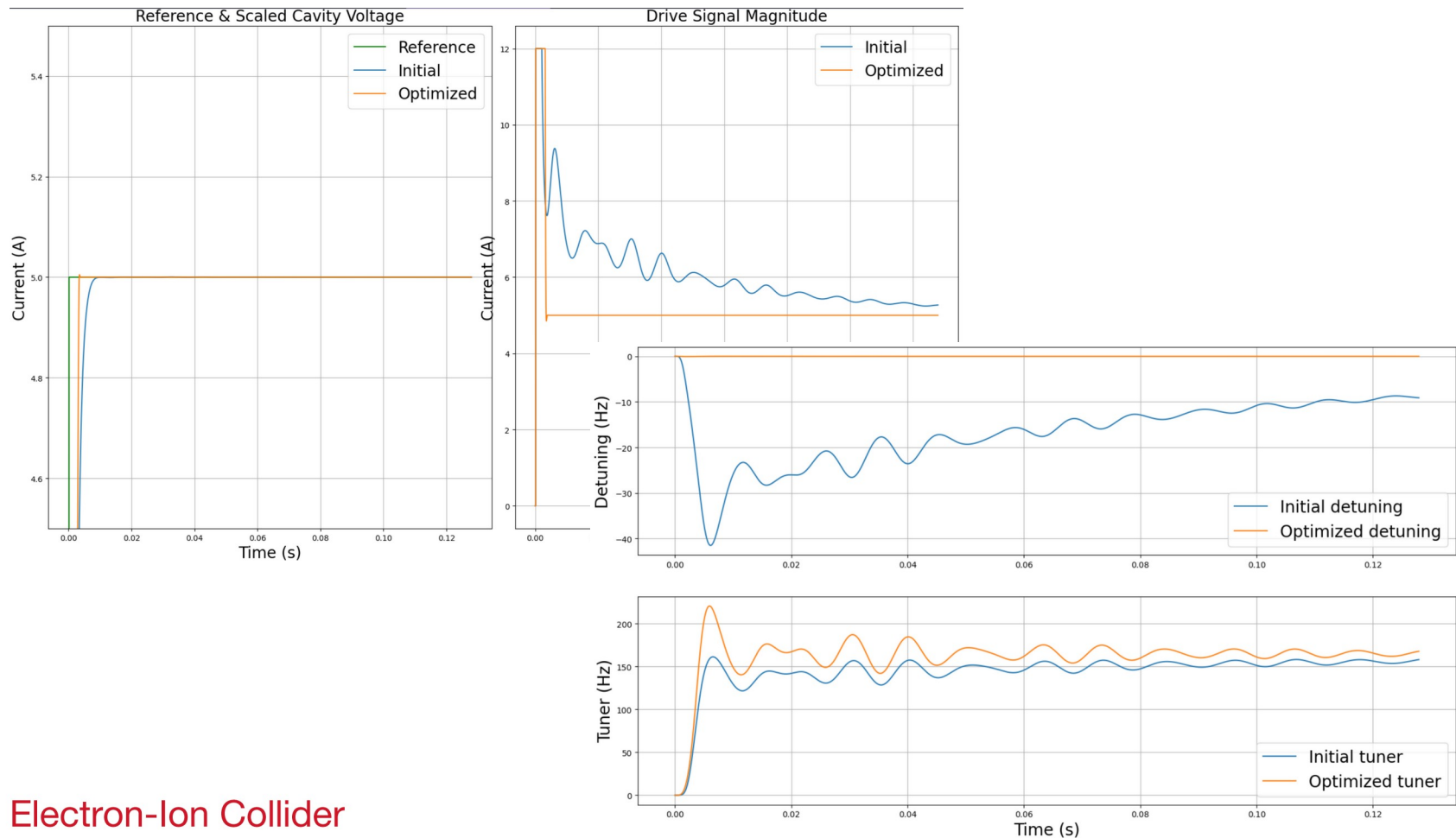
Tuning Strategy : ADRC

- We have 4 tunable parameters L_1 , L_2 (the observer gains), b (plant gain), and G (feedback gain)
- Will use a pole placement technique [6]:
 - Choosing $G = \omega_c$ to place controller pole at $-\omega_c$
 - Choosing $L_1 = 2\omega_{ob}$, $L_2 = \omega_{ob}^2$ to place ESO pole at $-\omega_{ob}$
- Can define the observer to have $\omega_{ob} = \alpha \omega_c$, where α is some positive number
- We are left with b which will depend on our system
- Overall, left with 3 tunable parameters, ω_c , b , and α

Optimal Tuning : ADRC

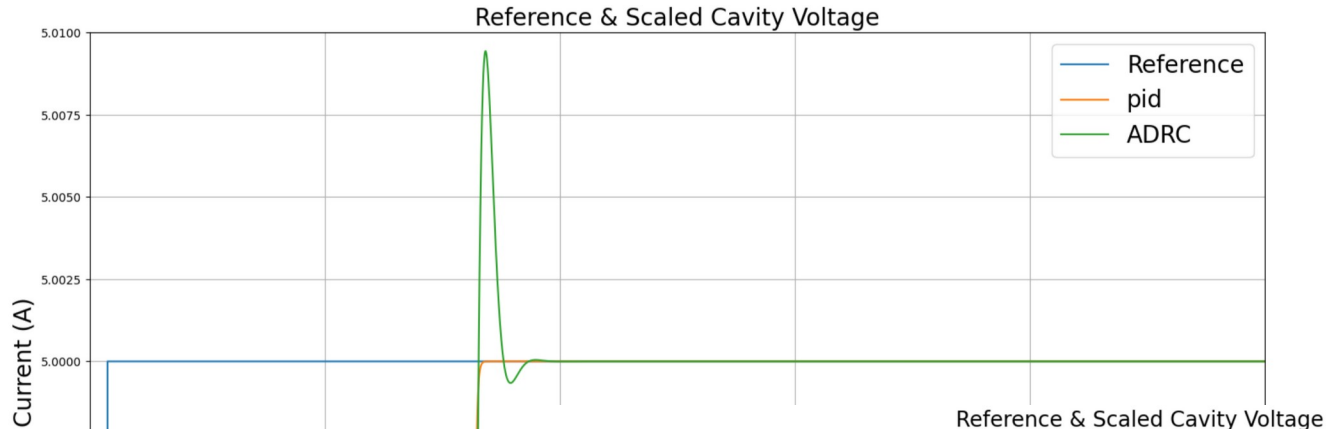
- Both field control uses an ADRC while resonance control still uses a PID:

$$\theta = [\omega_c, b, \alpha, K_{PR}, K_{IR}, K_{DR}]$$

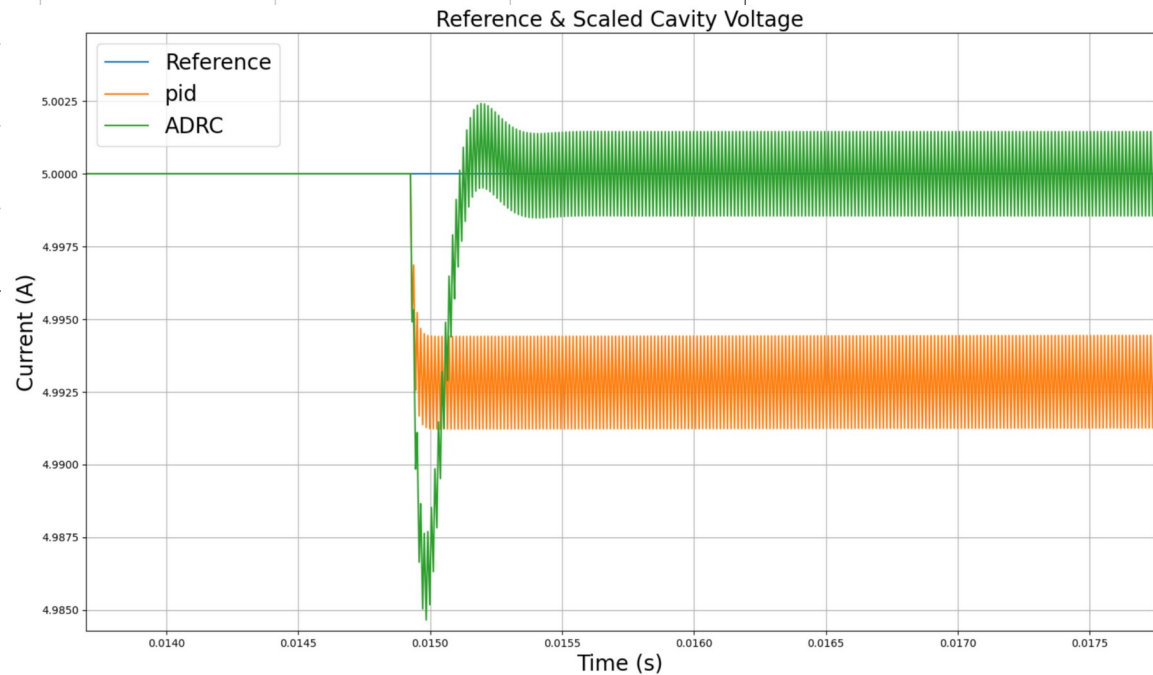


Comparing the PID and ADRC

- Comparing the step response and disturbance rejection



To the left : Step response



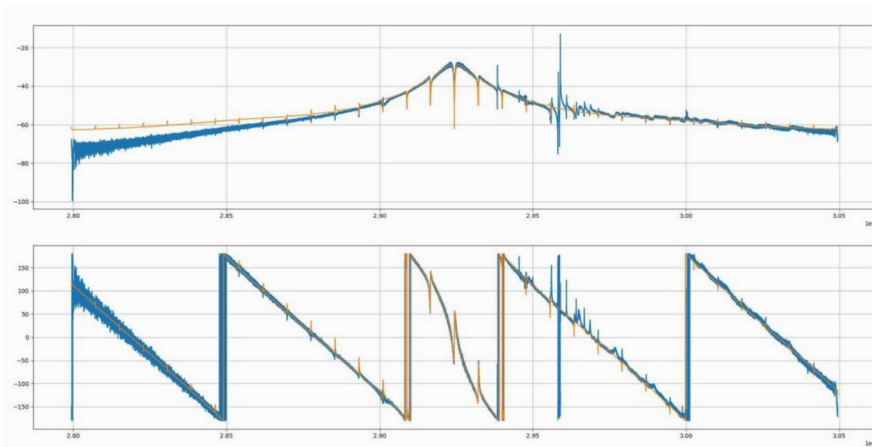
To the right : Loop reaction to
beam-loading

Problems With The Optimization Approach

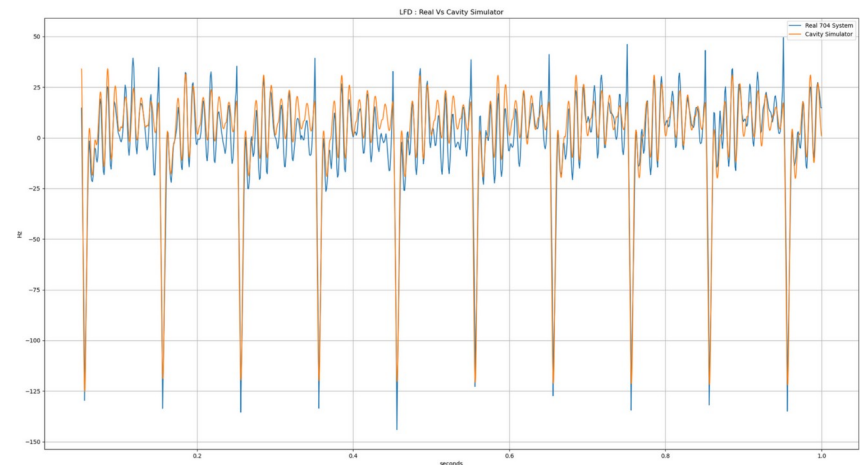
- Optimizing to a step function is a good general case, however:
 - May want to optimize under a multitude of cases and take a parameter average
 - With beam-loading, at injection, after bunch splitting, at store, etc.
 - Can also change parameters in real time
- This optimization approach is very brute force
 - Have to the simulation at every optimization step
 - Can take a while to find a global minimum
 - Could try algorithms that take into account gradients (BFGS)
 - *Could also write the simulator in a faster language...*
- Should take physics restrictions into account
 - Proportional gain, impedance reduction, and gain/phase margin are coupled quantities

Operational Applications

- To use this optimization method, need to know the system parameters reasonably well
- Not all system parameters are known before hand:
 - “Easy” parameters : Cavity Q_L , amplifier gains
 - Uncertain parameters : Loop delay, losses (cable, splitters, etc.)
- Can “probe” the system, extract parameters by fitting a model
 - In either frequency or time domain



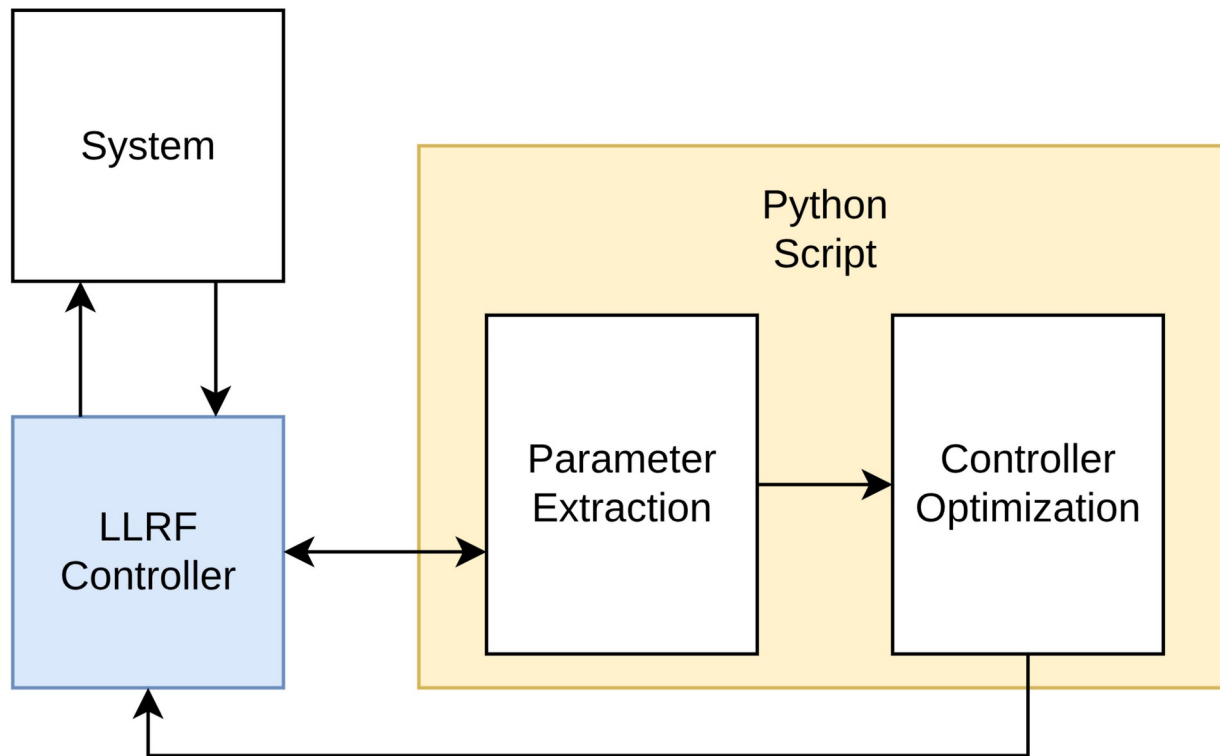
Parameter Extraction In Frequency-Domain



Parameter Extraction In Time-Domain

Operational Applications (cont.)

- Once the parameters are extracted, they can be used in the simulator
- Now, the controller optimization can proceed
- Whole process can be automated using Python scripting



Conclusions and Future Work

- Work is still in its early stages
 - The loss function is too simple
 - Should be able to enforce stability (> 60 degrees phase margin, for example)
 - Need more informed bounds for the tunable parameters
 - Need methodology to compare stability of different controllers and the multiple loops
 - Perform evaluation of LQR
- Using system identification techniques and numerical optimization, we are able to configure all loops involved in a RF control system
- Would appreciate feedback and suggestions. Thank you for listening, any questions?

References

- [1] J. Tuckmantel, Cavity-Beam-Transmitter Interaction Formula Collection with Derivation
- [2] J. R. Delayen, PONDEROMOTIVE INSTABILITIES AND MICROPHONICS – A TUTORIAL
- [3] Storn, R and Price, K, Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization, 1997, 11, 341 – 359
- [4] Gao, F. and Han, L. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. 2012. Computational Optimization and Applications. 51:1, pp. 259-277
- [5] J. Han, From PID to Active Disturbance Rejection Control
- [6] J. Vincent, On active disturbance rejection based control design for superconducting RF cavities
- [7] S. Zhao, Fixed-Point Implementation of Active Disturbance Rejection Control for Superconducting Radio Frequency Cavities
- [8] <https://en.wikipedia.org>, Linear Quadratic Regulator

Acknowledgments

- Thanks to the EIC RF team for their advice, support, and feedback. S. V. Andres, M. Blaskiewicz, K. Mernick, F. Severino, G. Narayan, W. Xu, B. Xiao, S. Mai, A. Fahey and others.