Advances in Feedback Control for Power Grid Modernization

John W. Simpson-Porco
https://www.control.utoronto.ca/~jwsimpson/

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Delft Center for Systems and Control

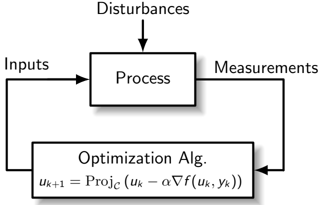
November 15, 2022

Prof. J. W. Simpson-Porco: Control Theory

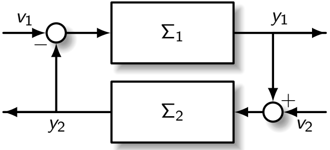
jwsimpson@ece.utoronto.ca

1. optimization and data-driven control
2. nonlinear systems
3. smart grid and energy systems

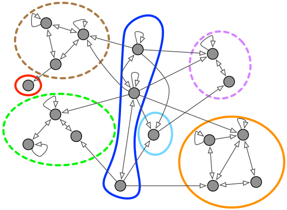
Feedback-Based Optimization



Nonlinear Systems



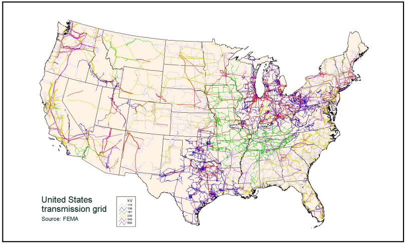
Network Dynamics & Control



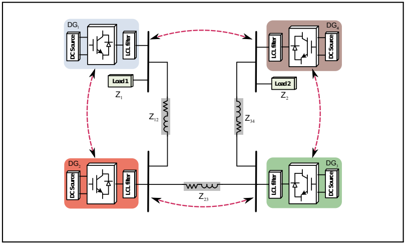
Prof. J. W. Simpson-Porco: Energy Systems

jwsimpson@ece.utoronto.ca

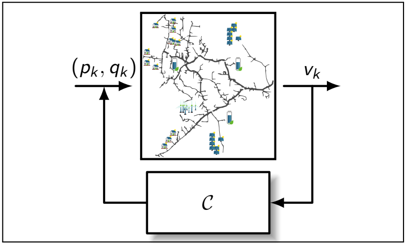
Power Flow Analysis & Algorithms



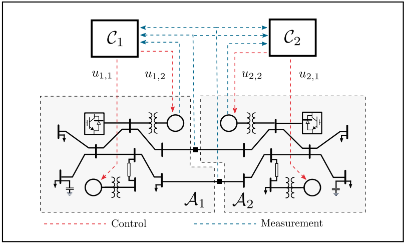
Microgrid Control & Optimization



Renewable Energy Integration

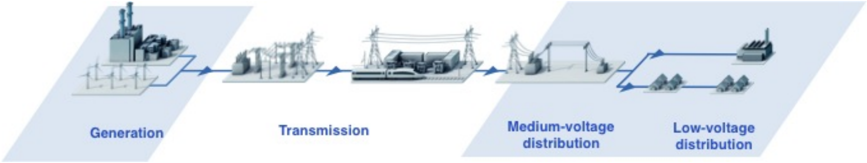


Next-Generation Hierarchical Control



The 20th Century Bulk Power System is Changing

A control engineer's view . . .



	Classical paradigm	**Modern trend**
Generation	Bulk, centralized	Small-scale, distrib.
Energy interface	Sync. generators	Power electronics
Net load uncertainty	Low	Renewable-driven
Information	Centralized	Distributed
Sensors/Actuators	Low-bandwidth	High-bandwidth

Grid Modernization Design Spec's for Control Engineers

Control engineer's #1 goal: make things **reliably boring**!

1. Coordinated Control of Many (Heterogeneous) Resources
   - **Real-time system optimization** w/ performance guarantees
   - **Scalability** to thousands of sensors/actuators
2. Grid Architecture (sensors/actuators/IT/algorithms/CPS)
   - Hierarchical **layering** across spatial and temporal scales
   - Prefer **localized** use of measurements (min. latency)
3. Practical Constraints in Power Engineering
   - Seamless integration with **legacy systems**
   - **Simple**, and congruent w/ *established power eng. principles*

Grid Modernization Design Spec's for Control Engineers

Control engineer's #1 goal: make things **reliably boring**!

1. Coordinated Control of Many (Heterogeneous) Resources
   - **Real-time system optimization** w/ performance guarantees
   - **Scalability** to thousands of sensors/actuators
2. Grid Architecture (sensors/actuators/IT/algorithms/CPS)
   - Hierarchical **layering** across spatial and temporal scales
   - Prefer **localized** use of measurements (min. latency)
3. Practical Constraints in Power Engineering
   - Seamless integration with **legacy systems**
   - **Simple**, and congruent w/ *established power eng. principles*

Grid Modernization Design Spec's for Control Engineers

Control engineer's #1 goal: make things **reliably boring**!

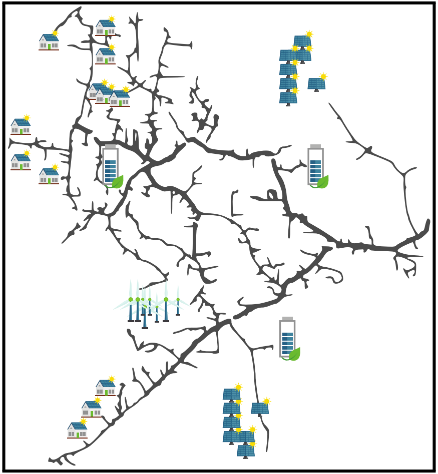
1. Coordinated Control of Many (Heterogeneous) Resources
   - **Real-time system optimization** w/ performance guarantees
   - **Scalability** to thousands of sensors/actuators
2. Grid Architecture (sensors/actuators/IT/algorithms/CPS)
   - Hierarchical **layering** across spatial and temporal scales
   - Prefer **localized** use of measurements (min. latency)
3. Practical Constraints in Power Engineering
   - Seamless integration with **legacy systems**
   - **Simple**, and congruent w/ *established power eng. principles*

Grid Modernization Design Spec's for Control Engineers

Control engineer's #1 goal: make things **reliably boring**!

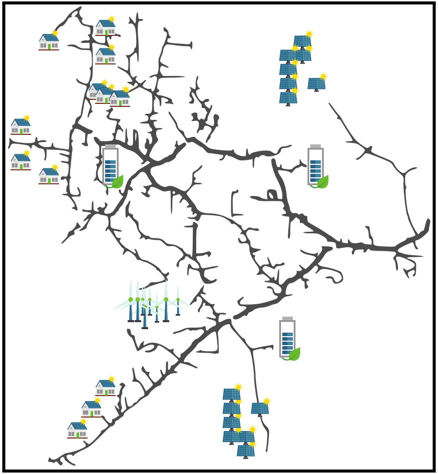
1. Coordinated Control of Many (Heterogeneous) Resources
   - **Real-time system optimization** w/ performance guarantees
   - **Scalability** to thousands of sensors/actuators
2. Grid Architecture (sensors/actuators/IT/algorithms/CPS)
   - Hierarchical **layering** across spatial and temporal scales
   - Prefer **localized** use of measurements (min. latency)
3. Practical Constraints in Power Engineering
   - Seamless integration with **legacy systems**
   - **Simple**, and congruent w/ *established power eng. principles*

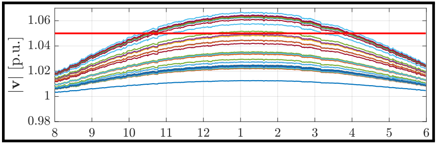
Example #1: Voltage Regulation in PV-Heavy Feeders



Want: Online control, robust w.r.t. grid model π

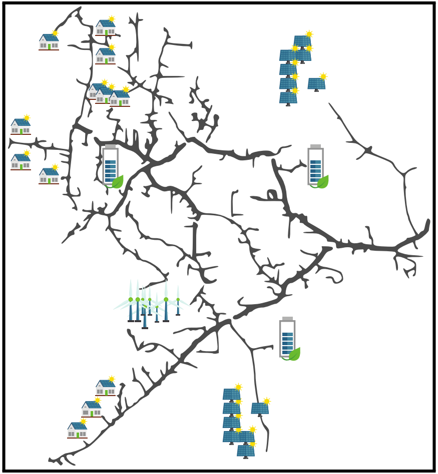
Example #1: Voltage Regulation in PV-Heavy Feeders

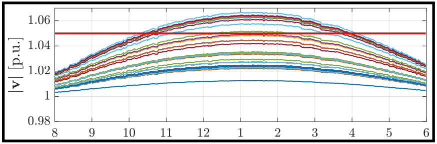




Want: Online control, robust w.r.t. grid model π

Example #1: Voltage Regulation in PV-Heavy Feeders





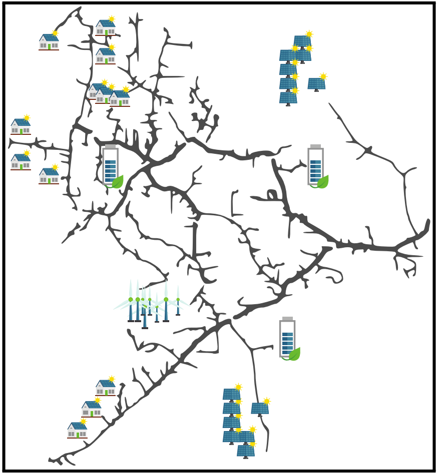
Grid Model: $v = \pi(u, w)$

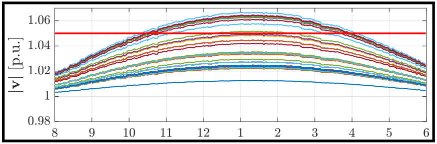
- u = controllable power
- w = uncontrollable power

$$\begin{aligned} \underset{u \in \{\text{Limits}\}}{\text{minimize}} \quad & \|u - u^{\text{nom}}\|_2^2 \\ \text{subject to} \quad & v \in [0.95, 1.05] \end{aligned}$$

Want: Online control, robust w.r.t. grid model π

Example #1: Voltage Regulation in PV-Heavy Feeders





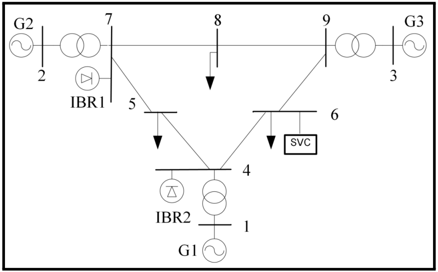
Grid Model: $v = \pi(u, w)$

- u = controllable power
- w = uncontrollable power

$$\begin{array}{ll} \underset{u \in \{\text{Limits}\}}{\text{minimize}} & \|u - u^{\text{nom}}\|_2^2 \\ \text{subject to} & v \in [0.95, 1.05] \end{array}$$

Want: Online control, robust w.r.t. grid model π

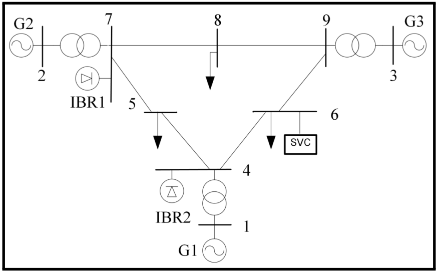
Example #2: Coordinated Voltage Control in Bulk Grid

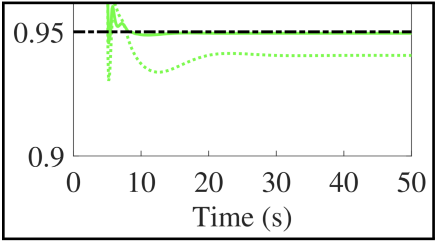


Control resources:

- Generators: $v_g^{\mathrm{ref}} \longrightarrow q_{\mathrm{g}}$
- SVC's: $v_s^{\mathrm{ref}} \longrightarrow q_{\mathrm{s}}$
- Inverters: $q_i^{\mathrm{ref}} \longrightarrow q_i$

Example #2: Coordinated Voltage Control in Bulk Grid

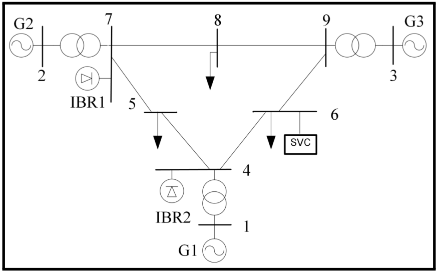


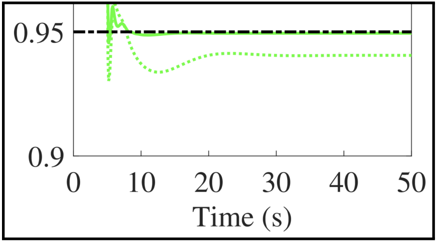


Control resources:

- Generators: $v_g^{\mathrm{ref}} \longrightarrow q_{\mathrm{g}}$
- SVC's: $v_s^{\mathrm{ref}} \longrightarrow q_{\mathrm{s}}$
- Inverters: $q_i^{\mathrm{ref}} \longrightarrow q_i$

Example #2: Coordinated Voltage Control in Bulk Grid

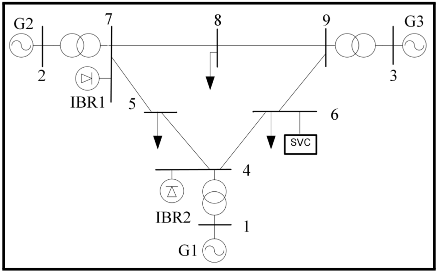


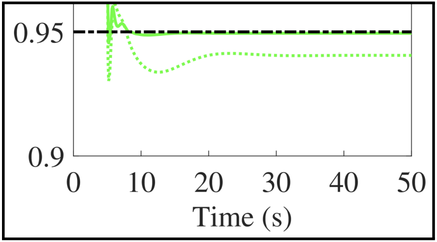


Control resources:

- Generators: $v_g^{\mathrm{ref}} \longrightarrow q_{\mathrm{g}}$
- SVC's: $v_s^{\mathrm{ref}} \longrightarrow q_{\mathrm{s}}$
- Inverters: $q_i^{\mathrm{ref}} \longrightarrow q_i$
- $u =$ vector of references
- $q =$ vector of power outputs

Example #2: Coordinated Voltage Control in Bulk Grid





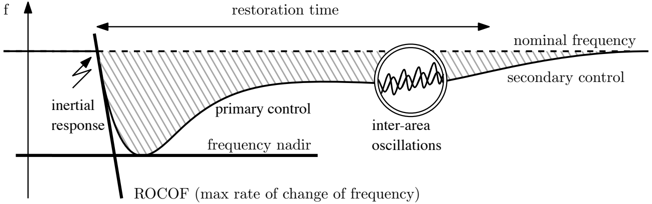
Control resources:

- Generators: $v_g^{\mathrm{ref}} \longrightarrow q_{\mathrm{g}}$
- SVC's: $v_s^{\mathrm{ref}} \longrightarrow q_{\mathrm{s}}$
- Inverters: $q_i^{\mathrm{ref}} \longrightarrow q_i$
- $u =$ vector of references
- $q =$ vector of power outputs

Model:
$$\begin{aligned} \dot{x} &= f(x, u, w) \\ (v, q) &= h(x, u, w) \end{aligned}$$

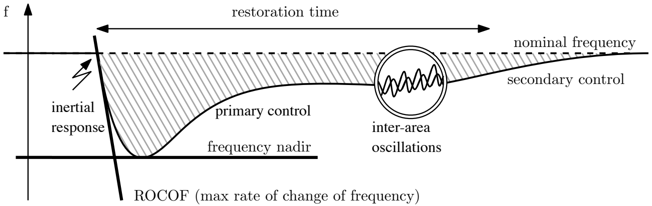
$$\begin{aligned} \underset{u \in \{\text{Limits}\}}{\text{minimize}} \quad & \|q - q^{\mathrm{nom}}\|_2^2 \\ \text{subject to} \quad & v \in [0.95, 1.05] \\ & q \in [q_{\min}, q_{\max}] \end{aligned}$$

Example #3: Secondary Frequency Control in Bulk Grid

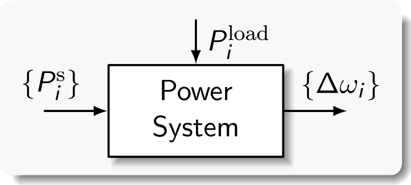


- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$

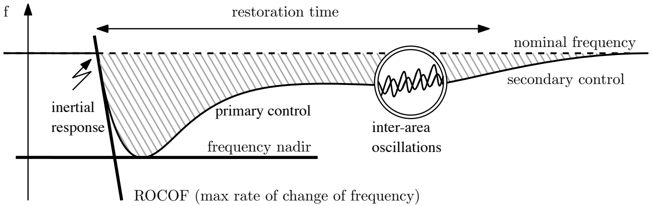
Example #3: Secondary Frequency Control in Bulk Grid



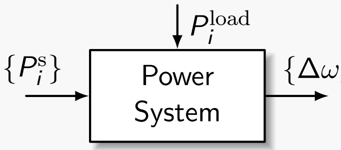
- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$



Example #3: Secondary Frequency Control in Bulk Grid

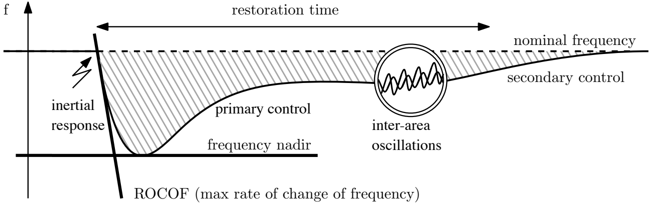


- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$

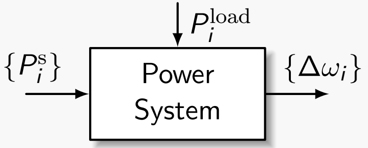


$$
\begin{array}{ll}
\underset{P_i^{\mathrm{s}} \in \{\text{limits}\}}{\text{minimize}} & \sum_{i=1}^{n} C_i(P_i^{\mathrm{s}}) \\
\text{subject to} & \Delta\omega_i = 0 \\
 & \text{(System dynamics)}
\end{array}
$$

Example #3: Secondary Frequency Control in Bulk Grid



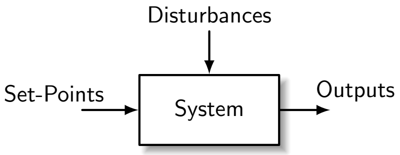
- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$



$$\begin{aligned} \underset{P_i^{\text{s}} \in \{\text{limits}\}}{\text{minimize}} \quad & \sum\nolimits_{i=1}^{n} C_i(P_i^{\text{s}}) \\ \text{subject to} \quad & \Delta\omega_i = 0 \\ & \text{(System dynamics)} \end{aligned}$$

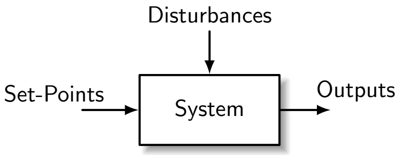
Want: Fast resource-allocating control loops (*architecture?*)

Offline vs. Online Optimization



Goal: Real-time regulation of system to an **optimal constrained operating point**.

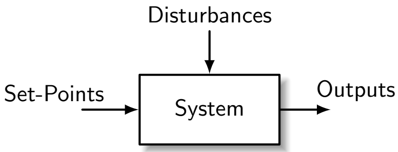
Offline vs. Online Optimization



Goal: Real-time regulation of system to an **optimal constrained operating point**.

Example: Optimal dispatch of generation in AC electric power systems, subject to frequency and voltage regulation constraints.

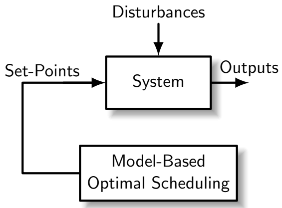
Offline vs. Online Optimization



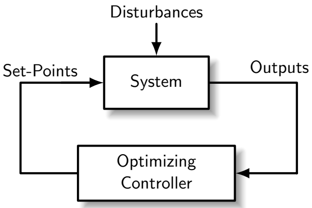
Goal: Real-time regulation of system to an **optimal constrained operating point**.

Example: Optimal dispatch of generation in AC electric power systems, subject to frequency and voltage regulation constraints.

Offline Optimization



Online Optimization



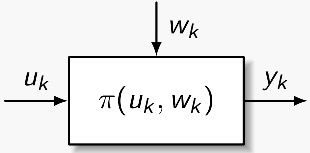
Framework #1

"The optimization algorithm approach"

Key Ingredients: Convex analysis/opt, robust control.

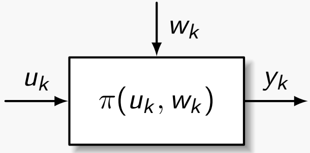
1. IEEE CDC: "*Towards robustness guarantees for feedback-based optimization*"
2. IEEE TPWRS: "*Measurement-Based Fast Coordinated Voltage Control for Transmission Grids* . . ."
3. IEEE CSS-L: "*Low-Gain Stability of Projected Integral Control for Input-Constrained Discrete-Time Nonlinear Systems*"

Feedback-Based Optimization of Memoryless Systems

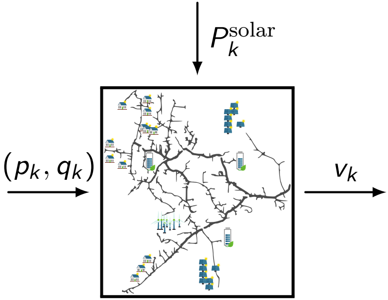


- System π **uncertain**
- Disturbance w **unmeasured**
- Output y **measured**

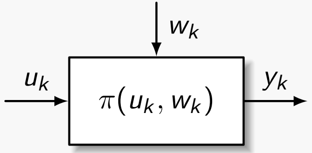
Feedback-Based Optimization of Memoryless Systems



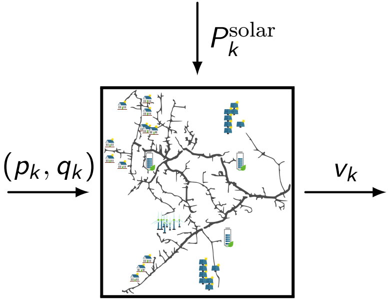
- System π **uncertain**
- Disturbance w **unmeasured**
- Output y **measured**



Feedback-Based Optimization of Memoryless Systems



- System π **uncertain**
- Disturbance w **unmeasured**
- Output y **measured**



$$\begin{aligned} \underset{u \in \mathcal{U}}{\text{minimize}} \quad & f(u) + g(y) \\ \text{subject to} \quad & y = \pi(u, w) \end{aligned}$$

Sample Assumptions:

- $\mathcal{U}$ is closed convex
- π is C^1 in u
- f, g are C^2 cvx, Lipschitz ∇

Feedback-Based Optimization of Memoryless Systems

The key steps on one slide.

$$\begin{aligned} \underset{u\in\mathcal{U}}{\text{minimize}} \quad & f(u)+g(y) \\ \text{subject to} \quad & y=\pi(u,w) \end{aligned} \quad \Longrightarrow \quad \underset{u\in\mathcal{U}}{\text{minimize}} \quad f(u)+g(\pi(u,w))$$

Offline Projected Gradient Descent:

$$u_{k+1}=\mathrm{Proj}_{\mathcal{U}}\left\{u_k-\alpha\left(\nabla f(u_k)+\partial\pi(u_k,w_k)^{\mathsf{T}}\nabla g(\pi(u_k,w_k))\right)\right\}$$

Approximate Offline Projected Gradient Descent:

$$u_{k+1}=\mathrm{Proj}_{\mathcal{U}}\left\{u_k-\alpha\left(\nabla f(u_k)+\boldsymbol{\Pi}^{\mathsf{T}}\nabla g(\pi(u_k,w_k))\right)\right\}$$

Approximate Online Projected Gradient Descent:

$$u_{k+1}=\mathrm{Proj}_{\mathcal{U}}\left\{u_k-\alpha\left(\nabla f(u_k)+\boldsymbol{\Pi}^{\mathsf{T}}\nabla g(\boldsymbol{y_k})\right)\right\}$$

Feedback-Based Optimization of Memoryless Systems

The key steps on one slide.

$$\begin{array}{ll} \underset{u \in \mathcal{U}}{\text{minimize}} & f(u) + g(y) \\ \text{subject to} & y = \pi(u, w) \end{array} \quad \Longrightarrow \quad \underset{u \in \mathcal{U}}{\text{minimize}} \quad f(u) + g(\pi(u, w))$$

Offline Projected Gradient Descent:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \partial \pi(u_k, w_k)^\mathsf{T} \nabla g(\pi(u_k, w_k)) \right) \right\}$$

Approximate Offline Projected Gradient Descent:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \mathbf{\Pi}^\mathsf{T} \nabla g(\pi(u_k, w_k)) \right) \right\}$$

Approximate Online Projected Gradient Descent:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \mathbf{\Pi}^\mathsf{T} \nabla g(y_k) \right) \right\}$$

Feedback-Based Optimization of Memoryless Systems

The key steps on one slide.

$$\begin{aligned} \underset{u \in \mathcal{U}}{\text{minimize}} \quad & f(u) + g(y) \\ \text{subject to} \quad & y = \pi(u, w) \end{aligned} \quad \Longrightarrow \quad \underset{u \in \mathcal{U}}{\text{minimize}} \quad f(u) + g(\pi(u, w))$$

Offline Projected Gradient Descent:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \partial \pi(u_k, w_k)^\mathsf{T} \nabla g(\pi(u_k, w_k)) \right) \right\}$$

Approximate Offline Projected Gradient Descent:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \mathbf{\Pi}^\mathsf{T} \nabla g(\pi(u_k, w_k)) \right) \right\}$$

Approximate Online Projected Gradient Descent:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \mathbf{\Pi}^\mathsf{T} \nabla g(\mathbf{y}_k) \right) \right\}$$

Feedback-Based Optimization of Memoryless Systems

The key steps on one slide.

$$\begin{aligned}\underset{u\in\mathcal{U}}{\text{minimize}} \quad & f(u)+g(y) \\ \text{subject to} \quad & y=\pi(u,w)\end{aligned} \qquad \Longrightarrow \qquad \underset{u\in\mathcal{U}}{\text{minimize}} \quad f(u)+g(\pi(u,w))$$

Offline Projected Gradient Descent:

$$u_{k+1}=\mathrm{Proj}_{\mathcal{U}}\left\{u_k-\alpha\left(\nabla f(u_k)+\partial\pi(u_k,w_k)^{\mathsf{T}}\nabla g(\pi(u_k,w_k))\right)\right\}$$

Approximate Offline Projected Gradient Descent:

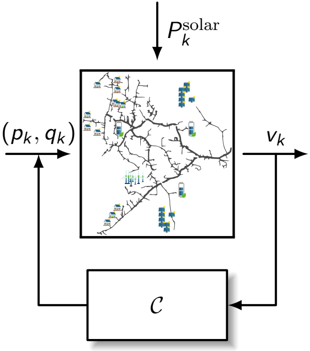
$$u_{k+1}=\mathrm{Proj}_{\mathcal{U}}\left\{u_k-\alpha\left(\nabla f(u_k)+\mathbf{\Pi}^{\mathsf{T}}\nabla g(\pi(u_k,w_k))\right)\right\}$$

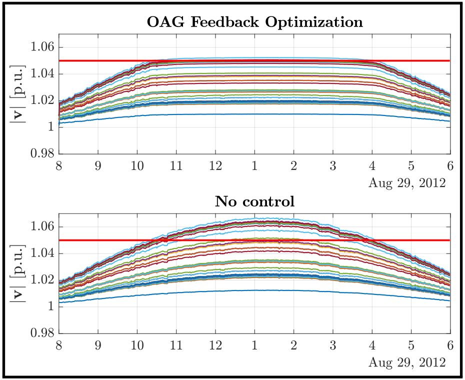
Approximate Online Projected Gradient Descent:

$$u_{k+1}=\mathrm{Proj}_{\mathcal{U}}\left\{u_k-\alpha\left(\nabla f(u_k)+\mathbf{\Pi}^{\mathsf{T}}\nabla g(\boldsymbol{y_k})\right)\right\}$$

Example #2: Voltage Regulation in PV-Heavy Feeders

Punchline: Maintains voltage in limits, minimizes PV curtailment, provably robust to large model variations





Convergence of Approx. Gradient Descent

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}}\Big\{u_k - \alpha F_w(u_k)\Big\}$$

$$F_w(u_k) = \nabla f(u_k) + \mathbf{\Pi}^\mathsf{T} \nabla g(\pi(u_k, w_k))$$

Theorem from VI Literature: Suppose that F_w is ρ-strongly monotone and L-Lipschitz continuous w.r.t. inner product $\langle x, y \rangle_P = x^\mathsf{T} P y$ with $P \succ 0$. Then

$$\alpha < \frac{2\rho}{L^2} \quad \Longrightarrow \quad \text{Global exp. convergence}$$

Problem: $F_w(u)$ is **uncertain**. How can we systematically check if F_w satisfies these assumptions?

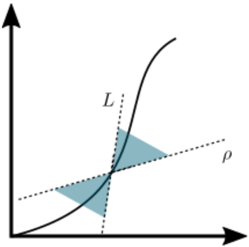
Convergence of Approx. Gradient Descent

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}}\Big\{u_k - \alpha F_w(u_k)\Big\}$$

$$F_w(u_k) = \nabla f(u_k) + \mathbf{\Pi}^\mathsf{T} \nabla g(\pi(u_k, w_k))$$

Theorem from VI Literature: Suppose that F_w is ρ-strongly monotone and L-Lipschitz continuous w.r.t. inner product $\langle x, y \rangle_P = x^\mathsf{T} P y$ with $P \succ 0$. Then

$$\alpha < \frac{2\rho}{L^2} \quad \implies \quad \text{Global exp. convergence}$$



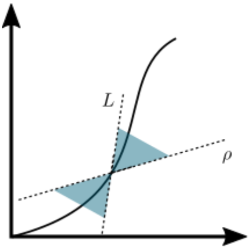
Problem: $F_w(u)$ is **uncertain**. How can we systematically check if F_w satisfies these assumptions?

Convergence of Approx. Gradient Descent

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}}\Big\{u_k - \alpha F_w(u_k)\Big\}$$
$$F_w(u_k) = \nabla f(u_k) + \mathbf{\Pi}^\mathsf{T} \nabla g(\pi(u_k, w_k))$$

Theorem from VI Literature: Suppose that F_w is ρ-strongly monotone and L-Lipschitz continuous w.r.t. inner product $\langle x, y \rangle_P = x^\mathsf{T} P y$ with $P \succ 0$. Then

$$\alpha < \frac{2\rho}{L^2} \quad \Longrightarrow \quad \text{Global exp. convergence}$$



Problem: $F_w(u)$ is **uncertain**. How can we systematically check if F_w satisfies these assumptions?

Convergence of Approx. Gradient Descent

$$\begin{aligned} \text{Mapping:} \quad & F_w(u) &&= \nabla f(u) + \mathbf{\Pi}^\mathsf{T} \nabla g(\pi(u,w)) \\ \text{Jacobian:} \quad & \partial F_w(u) &&= \nabla^2 f(u) + \mathbf{\Pi}^\mathsf{T} \nabla^2 g(\pi(u,w)) \partial \pi(u,w) \end{aligned}$$

Proposition: Given $P \succ 0$, equivalent statements:

1. F_w is ρ-strongly mono. and L-Lipschitz w.r.t $\langle \cdot, \cdot \rangle_P$ on $\mathcal{U}$
2. the following matrix inequality holds:

$$\begin{bmatrix} \partial F_w(u) \\ I \end{bmatrix}^\mathsf{T} \left(\underbrace{\begin{bmatrix} 2 & -(\rho + L) \\ -(\rho + L) & 2\rho L \end{bmatrix}}_{:= X_{\rho,L}} \otimes P \right) \begin{bmatrix} \partial F_w(u) \\ I \end{bmatrix} \preceq \mathbb{0}, \quad \forall\, u \in \mathcal{U}.$$

This didn't seem to help. **Robust control** to the rescue!

Idea: **Overbound** the set $\partial F_w(\mathcal{U})$ by a *simpler set* $\mathcal{J}$!

Convergence of Approx. Gradient Descent

$$\begin{aligned} \text{Mapping:} \quad & F_w(u) && = \nabla f(u) + \mathbf{\Pi}^\mathsf{T} \nabla g(\pi(u,w)) \\ \text{Jacobian:} \quad & \partial F_w(u) && = \nabla^2 f(u) + \mathbf{\Pi}^\mathsf{T} \nabla^2 g(\pi(u,w)) \partial \pi(u,w) \end{aligned}$$

Proposition: Given $P \succ 0$, equivalent statements:

1. F_w is ρ-strongly mono. and L-Lipschitz w.r.t $\langle \cdot, \cdot \rangle_P$ on $\mathcal{U}$
2. the following matrix inequality holds:

$$\begin{bmatrix} \partial F_w(u) \\ I \end{bmatrix}^\mathsf{T} (X_{\rho,L} \otimes P) \begin{bmatrix} \partial F_w(u) \\ I \end{bmatrix} \preceq \mathbb{0}, \quad \forall\, u \in \mathcal{U}.$$

This didn't seem to help. **Robust control** to the rescue!

Idea: **Overbound** the set $\partial F_w(\mathcal{U})$ by a *simpler set* $\mathcal{J}$!

Convergence of Approx. Gradient Descent

$$\begin{aligned} \text{Mapping:} \quad & F_w(u) && = \nabla f(u) + \boldsymbol{\Pi}^\mathsf{T} \nabla g(\pi(u,w)) \\ \text{Jacobian:} \quad & \partial F_w(u) && = \nabla^2 f(u) + \boldsymbol{\Pi}^\mathsf{T} \nabla^2 g(\pi(u,w)) \partial \pi(u,w) \end{aligned}$$

Proposition: Given $P \succ 0$, equivalent statements:

1. F_w is ρ-strongly mono. and L-Lipschitz w.r.t $\langle \cdot, \cdot \rangle_P$ on $\mathcal{U}$
2. the following matrix inequality holds:

$$\begin{bmatrix} \partial F_w(u) \\ I \end{bmatrix}^\mathsf{T} (X_{\rho,L} \otimes P) \begin{bmatrix} \partial F_w(u) \\ I \end{bmatrix} \preceq \mathbb{0}, \quad \forall\, u \in \mathcal{U}.$$

This didn't seem to help. **Robust control** to the rescue!

Idea: Overbound the set $\partial F_w(\mathcal{U})$ by a *simpler set* $\mathcal{J}$!

Robust Analysis for Approx. Gradient Descent

Robust Strong Monotonicity and Lipschitzness: Suppose we have a set $\mathcal{J}$ of matrices such that $\partial F_w(u) \subseteq \mathcal{J}$ for all $u \in \mathcal{U}$. Then $F_w(u)$ is ρ-strongly monotone and L-Lipschitz if

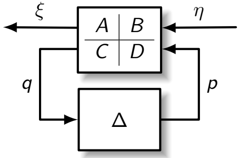
$$\begin{bmatrix} J \\ I \end{bmatrix}^\mathsf{T} (X_{\rho,L} \otimes P) \begin{bmatrix} J \\ I \end{bmatrix} \preceq \mathbb{0}, \quad \forall\, J \in \mathcal{J}.$$

For some (very practical) types of sets $\mathcal{J}$, this is tractable.

Linear Fractional Uncertainty

$$\mathcal{J} = \{A + B\Delta(I - D\Delta)^{-1}C \;:\; \Delta \in \boldsymbol{\Delta}\}$$

$$\begin{bmatrix} q \\ p \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} q \\ p \end{bmatrix} \geq 0 \qquad \forall \Theta \in \boldsymbol{\Theta}.$$



Robust Analysis for Approx. Gradient Descent

Robust Strong Monotonicity and Lipschitzness: Suppose we have a set $\mathcal{J}$ of matrices such that $\partial F_w(u) \subseteq \mathcal{J}$ for all $u \in \mathcal{U}$. Then $F_w(u)$ is ρ-strongly monotone and L-Lipschitz if

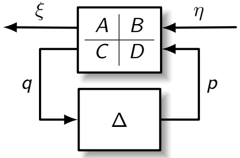
$$\begin{bmatrix} J \\ I \end{bmatrix}^\mathsf{T} (X_{\rho,L} \otimes P) \begin{bmatrix} J \\ I \end{bmatrix} \preceq \mathbb{0}, \quad \forall J \in \mathcal{J}.$$

For some (very practical) types of sets $\mathcal{J}$, this is tractable.

Linear Fractional Uncertainty

$$\mathcal{J} = \{A + B\Delta(I - D\Delta)^{-1}C \ : \ \Delta \in \boldsymbol{\Delta}\}$$

$$\begin{bmatrix} q \\ p \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} q \\ p \end{bmatrix} \geq 0 \qquad \forall \Theta \in \boldsymbol{\Theta}.$$



Robust Analysis for Approx. Gradient Descent

Robust Strong Monotonicity and Lipschitzness: Suppose we have a set $\mathcal{J}$ of matrices such that $\partial F_w(u) \subseteq \mathcal{J}$ for all $u \in \mathcal{U}$. Then $F_w(u)$ is ρ-strongly monotone and L-Lipschitz if

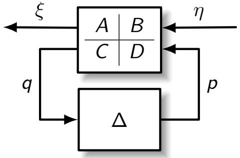
$$\begin{bmatrix} J \\ I \end{bmatrix}^{\mathsf{T}} (X_{\rho,L} \otimes P) \begin{bmatrix} J \\ I \end{bmatrix} \preceq \mathbb{0}, \quad \forall\, J \in \mathcal{J}.$$

For some (very practical) types of sets $\mathcal{J}$, this is tractable.

Linear Fractional Uncertainty

$$\mathcal{J} = \{A + B\Delta(I - D\Delta)^{-1}C \ : \ \Delta \in \boldsymbol{\Delta}\}$$

$$\begin{bmatrix} q \\ p \end{bmatrix}^{\mathsf{T}} \Theta \begin{bmatrix} q \\ p \end{bmatrix} \geq 0 \qquad \forall \Theta \in \boldsymbol{\Theta}.$$



Robust Analysis of Approx. Grad Descent

$$\partial F_w(u) = \nabla^2 f(u) + \mathbf{\Pi}^\mathsf{T} \nabla^2 g(\pi(u, w)) \partial \pi(u, w)$$

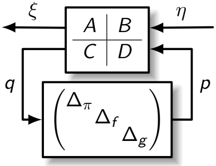
Robust Analysis of Approx. Grad Descent

$$\partial F_w(u) = \nabla^2 f(u) + \boldsymbol{\Pi}^\mathsf{T} \nabla^2 g(\pi(u,w)) \partial \pi(u,w)$$

Example: Suppose we had

- $\partial \pi(u,w) \in \{\boldsymbol{\Pi} + \Delta_\pi \ : \ \|\Delta_\pi\|_2 \leq \gamma\}$

$$\partial F_w(u) \subseteq \mathcal{J} = \Big\{ \Delta_f + \boldsymbol{\Pi}^\mathsf{T} \Delta_g (\boldsymbol{\Pi} + \Delta_\pi) \\ \|\Delta_\pi\| \leq \gamma, m_f \preceq \Delta_f \preceq L_f, 0 \preceq \Delta_g \preceq L_g \Big\}$$



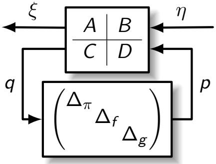
Robust Analysis of Approx. Grad Descent

$$\partial F_w(u) = \nabla^2 f(u) + \mathbf{\Pi}^\mathsf{T} \nabla^2 g(\pi(u,w)) \partial \pi(u,w)$$

Example: Suppose we had

- $\partial \pi(u,w) \in \{\mathbf{\Pi} + \Delta_\pi \;:\; \|\Delta_\pi\|_2 \leq \gamma\}$

$$\partial F_w(u) \subseteq \mathcal{J} = \Big\{ \Delta_f + \mathbf{\Pi}^\mathsf{T} \Delta_g (\mathbf{\Pi} + \Delta_\pi) \\ \|\Delta_\pi\| \leq \gamma, m_f \preceq \Delta_f \preceq L_f, 0 \preceq \Delta_g \preceq L_g \Big\}$$



Main Analysis SDP: Any map F_w with $\partial F_w(u) \subseteq \mathcal{J}$ is ρ-strongly monotone and L-Lipschitz if $\exists P \succ 0, \Theta \in \mathbf{\Theta}$ s.t.

$$\begin{bmatrix} A & B(\mathbf{\Pi}) \\ I & 0 \end{bmatrix}^\mathsf{T} (X_{\rho,L} \otimes P) \begin{bmatrix} A & B(\mathbf{\Pi}) \\ I & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \preceq 0.$$

Example #1: Voltage Regulation in PV-Heavy Feeders

$$\underset{(p_i,q_i)\in\mathcal{C}_i}{\text{minimize}} \quad \underbrace{\left\| \begin{pmatrix} p \\ q \end{pmatrix} - \begin{pmatrix} p^\star \\ 0 \end{pmatrix} \right\|_2^2}_{\text{curtailment}} + \underbrace{\gamma \sum_{i=1}^{m} \max(0, \underline{v}_i - v_i, v_i - \overline{v}_i)^2}_{\text{Soft voltage constraint}}$$

$$\text{subject to} \quad v = \pi(p, q, w) = \mathrm{PowerFlow}(p, q, w)$$

- Replace $\partial\pi$ with any **linearization** Π^{nom} of power flow equations
- Model uncertainty via **norm-bound** from nominal Jacobian

$$\partial\pi(u, w) \in \{\Pi^{\mathrm{nom}} + \Delta \ : \ \|\Delta\|_2 \leq \gamma\}.$$

Example #1: Voltage Regulation in PV-Heavy Feeders

$$\underset{(p_i,q_i)\in\mathcal{C}_i}{\text{minimize}} \quad \underbrace{\left\| \begin{pmatrix} p \\ q \end{pmatrix} - \begin{pmatrix} p^\star \\ 0 \end{pmatrix} \right\|_2^2}_{\text{curtailment}} + \underbrace{\gamma \sum_{i=1}^{m} \max(0, \underline{v}_i - v_i, v_i - \overline{v}_i)^2}_{\text{Soft voltage constraint}}$$

$$\text{subject to} \quad v = \pi(p, q, w) = \mathrm{PowerFlow}(p, q, w)$$

- Replace $\partial\pi$ with any **linearization** Π^{nom} of power flow equations
- Model uncertainty via **norm-bound** from nominal Jacobian

$$\partial\pi(u, w) \in \{\Pi^{\mathrm{nom}} + \Delta \ : \ \|\Delta\|_2 \leq \gamma\}.$$

Example #1: Voltage Regulation in PV-Heavy Feeders

$$\underset{(p_i,q_i)\in\mathcal{C}_i}{\text{minimize}} \quad \underbrace{\left\| \begin{pmatrix} p \\ q \end{pmatrix} - \begin{pmatrix} p^\star \\ 0 \end{pmatrix} \right\|_2^2}_{\text{curtailment}} + \underbrace{\gamma \sum_{i=1}^{m} \max(0, \underline{v}_i - v_i, v_i - \overline{v}_i)^2}_{\text{Soft voltage constraint}}$$

$$\text{subject to} \quad v = \pi(p,q,w) = \mathrm{PowerFlow}(p,q,w)$$

- Replace $\partial\pi$ with any **linearization** Π^{nom} of power flow equations
- Model uncertainty via **norm-bound** from nominal Jacobian

$$\partial\pi(u,w) \in \{\Pi^{\mathrm{nom}} + \Delta \ : \ \|\Delta\|_2 \leq \gamma\}.$$

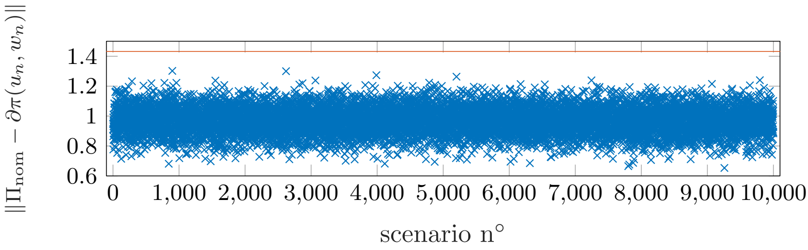
Example #1: Voltage Regulation in PV-Heavy Feeders

$$\underset{(p_i,q_i)\in\mathcal{C}_i}{\text{minimize}} \quad \underbrace{\left\| \begin{pmatrix} p \\ q \end{pmatrix} - \begin{pmatrix} p^\star \\ 0 \end{pmatrix} \right\|_2^2}_{\text{curtailment}} + \underbrace{\gamma \sum_{i=1}^{m} \max(0, \underline{v}_i - v_i, v_i - \overline{v}_i)^2}_{\text{Soft voltage constraint}}$$

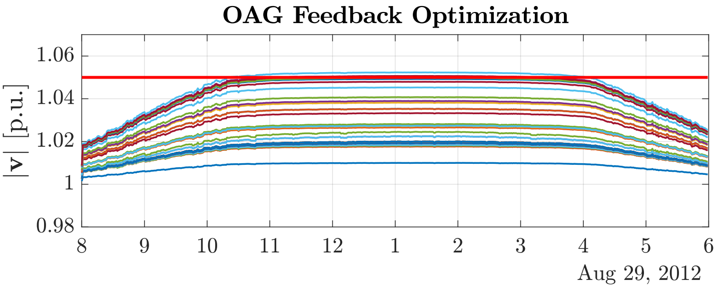
$$\text{subject to} \quad v = \pi(p,q,w) = \mathrm{PowerFlow}(p,q,w)$$

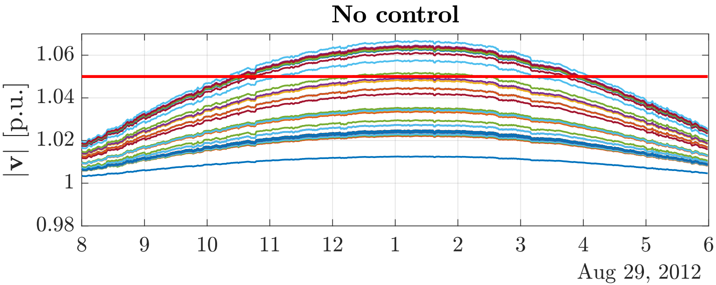
- Replace $\partial\pi$ with any **linearization** Π^{nom} of power flow equations
- Model uncertainty via **norm-bound** from nominal Jacobian

$$\partial\pi(u,w) \in \{\Pi^{\mathrm{nom}} + \Delta \ : \ \|\Delta\|_2 \le \gamma\}.$$



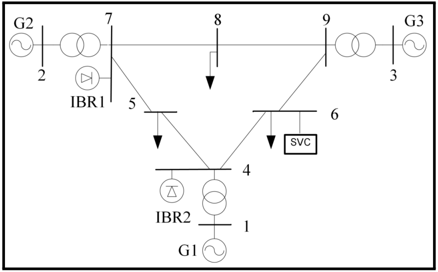
Provably Stable Coordinated Voltage Control





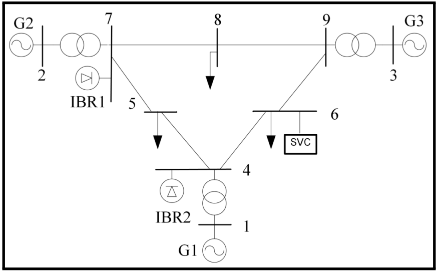
Example #2: Coordinated Voltage Control in Bulk Grid

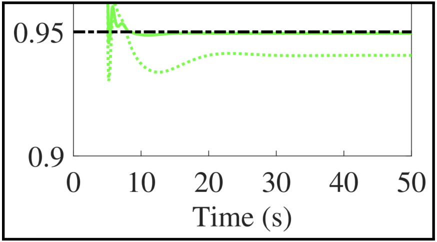
$$\begin{aligned} \underset{v_g^{\text{ref}}, v_s^{\text{ref}}, q_i^{\text{ref}}}{\text{minimize}} \quad & \mathsf{Cost}(q_g, q_s, q_i) + \mathsf{Penalty}(v) + \mathsf{Penalty}(q_g, q_s) \\ \text{subject to} \quad & (q_g, q_s, v) = \pi(v_g^{\text{ref}}, v_s^{\text{ref}}, q_i^{\text{ref}}, \mathsf{Load}) \\ & (v_g^{\text{ref}}, v_s^{\text{ref}}, q_i^{\text{ref}}) \in \{\mathsf{Limits}\} \end{aligned}$$



Example #2: Coordinated Voltage Control in Bulk Grid

$$\begin{aligned} \underset{v_g^{\text{ref}}, v_s^{\text{ref}}, q_i^{\text{ref}}}{\text{minimize}} \quad & \mathsf{Cost}(q_g, q_s, q_i) + \mathsf{Penalty}(v) + \mathsf{Penalty}(q_g, q_s) \\ \text{subject to} \quad & (q_g, q_s, v) = \pi(v_g^{\text{ref}}, v_s^{\text{ref}}, q_i^{\text{ref}}, \mathsf{Load}) \\ & (v_g^{\text{ref}}, v_s^{\text{ref}}, q_i^{\text{ref}}) \in \{\mathsf{Limits}\} \end{aligned}$$





Outlook for Theory of Feedback-Based Optimization

1. Enhanced **transient performance** via, e.g., *loop shaping*:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \Big(\nabla f(u_k) + \Pi^{\mathsf{T}} \nabla g(y_k) \Big) - \Gamma^{\mathsf{T}} (y_k - y_{k-1}) \right\}$$

2. **Synthesis** of Π for given **information structure**

$$\begin{aligned} \underset{\Pi,\Theta}{\text{minimize}} \quad & \|\Pi - \Pi_{\text{nom}}\| \\ \text{subject to} \quad & \Pi \in \boldsymbol{\Pi} \\ & \begin{bmatrix} A & B(\boldsymbol{\Pi}) \\ I & 0 \end{bmatrix}^{\mathsf{T}} (X_{\rho,L} \otimes P) \begin{bmatrix} A & B(\boldsymbol{\Pi}) \\ I & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^{\mathsf{T}} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \preceq 0. \end{aligned}$$

3. Gain-scheduling synthesis of Π

4. Hard output constraints via **primal-dual**

5. **Learn** gains Π online subject to **robust stability constraints**

Outlook for Theory of Feedback-Based Optimization

1. Enhanced **transient performance** via, e.g., *loop shaping*:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \Pi^\mathsf{T} \nabla g(y_k) \right) - \Gamma^\mathsf{T} (y_k - y_{k-1}) \right\}$$

2. **Synthesis** of Π for given **information structure**

$$\begin{aligned} \underset{\Pi,\Theta}{\text{minimize}} \quad & \|\Pi - \Pi_{\mathrm{nom}}\| \\ \text{subject to} \quad & \Pi \in \boldsymbol{\Pi} \\ & \begin{bmatrix} A & B(\boldsymbol{\Pi}) \\ I & 0 \end{bmatrix}^\mathsf{T} (X_{\rho,L} \otimes P) \begin{bmatrix} A & B(\boldsymbol{\Pi}) \\ I & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \preceq 0. \end{aligned}$$

3. Gain-scheduling synthesis of Π

4. Hard output constraints via **primal-dual**

5. **Learn** gains Π online subject to **robust stability constraints**

Outlook for Theory of Feedback-Based Optimization

1. Enhanced **transient performance** via, e.g., *loop shaping*:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \Pi^{\mathsf{T}} \nabla g(y_k) \right) - \Gamma^{\mathsf{T}} (y_k - y_{k-1}) \right\}$$

2. **Synthesis** of Π for given **information structure**

$$\begin{aligned} \underset{\Pi,\Theta}{\text{minimize}} \quad & \|\Pi - \Pi_{\text{nom}}\| \\ \text{subject to} \quad & \Pi \in \mathbf{\Pi} \\ & \begin{bmatrix} A & B(\mathbf{\Pi}) \\ I & 0 \end{bmatrix}^{\mathsf{T}} (X_{\rho,L} \otimes P) \begin{bmatrix} A & B(\mathbf{\Pi}) \\ I & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^{\mathsf{T}} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \preceq 0. \end{aligned}$$

3. Gain-scheduling synthesis of Π

4. Hard output constraints via **primal-dual**

5. **Learn** gains Π online subject to **robust stability constraints**

Outlook for Theory of Feedback-Based Optimization

1. Enhanced **transient performance** via, e.g., *loop shaping*:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \Pi^{\mathsf{T}} \nabla g(y_k) \right) - \Gamma^{\mathsf{T}} (y_k - y_{k-1}) \right\}$$

2. **Synthesis** of Π for given **information structure**

$$\begin{aligned} \underset{\Pi,\Theta}{\text{minimize}} \quad & \|\Pi - \Pi_{\mathrm{nom}}\| \\ \text{subject to} \quad & \Pi \in \boldsymbol{\Pi} \\ & \begin{bmatrix} A & B(\boldsymbol{\Pi}) \\ I & 0 \end{bmatrix}^{\mathsf{T}} (X_{\rho,L} \otimes P) \begin{bmatrix} A & B(\boldsymbol{\Pi}) \\ I & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^{\mathsf{T}} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \preceq 0. \end{aligned}$$

3. Gain-scheduling synthesis of Π

4. Hard output constraints via **primal-dual**

5. **Learn** gains Π online subject to **robust stability constraints**

Outlook for Theory of Feedback-Based Optimization

1. Enhanced **transient performance** via, e.g., *loop shaping*:

$$u_{k+1} = \mathrm{Proj}_{\mathcal{U}} \left\{ u_k - \alpha \left(\nabla f(u_k) + \Pi^\mathsf{T} \nabla g(y_k) \right) - \Gamma^\mathsf{T}(y_k - y_{k-1}) \right\}$$

2. **Synthesis** of Π for given **information structure**

$$\begin{aligned} \underset{\Pi,\Theta}{\text{minimize}} \quad & \|\Pi - \Pi_{\text{nom}}\| \\ \text{subject to} \quad & \Pi \in \mathbf{\Pi} \\ & \begin{bmatrix} A & B(\mathbf{\Pi}) \\ I & 0 \end{bmatrix}^\mathsf{T} (X_{\rho,L} \otimes P) \begin{bmatrix} A & B(\mathbf{\Pi}) \\ I & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \preceq 0. \end{aligned}$$

3. Gain-scheduling synthesis of Π

4. Hard output constraints via **primal-dual**

5. **Learn** gains Π online subject to **robust stability constraints**

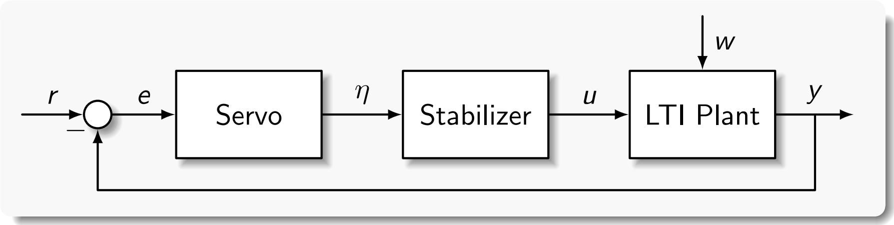
Framework #2

“The control-theoretic approach”

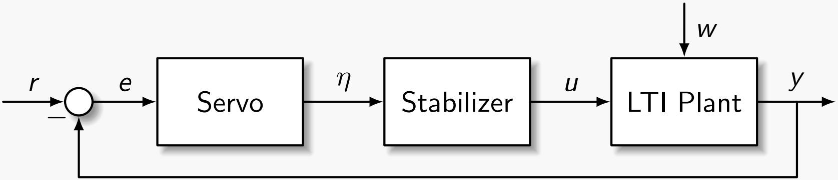
Key Ingredients: Tracking and regulation, convex opt.

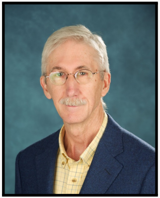
1. IEEE TAC: “*Linear-Convex Optimal Steady-State Control*”
2. IEEE TAC: “*Analysis and Synthesis of Low-Gain Integral Controllers for Nonlinear Systems*”
3. IEEE CDC: “*Low-Gain Stabilizers for Linear-Convex Optimal Steady-State Control*”

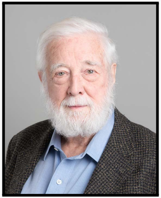
Theory of Tracking and Regulation for LTI Systems



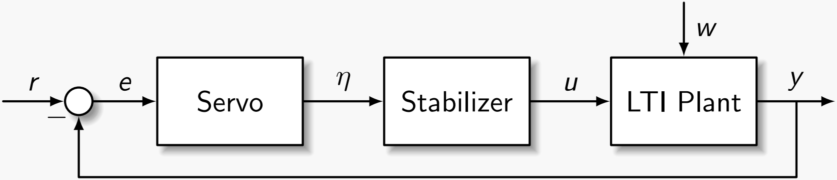
Theory of Tracking and Regulation for LTI Systems

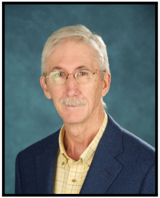


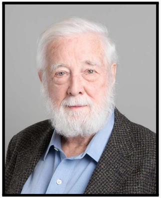




Theory of Tracking and Regulation for LTI Systems







Our goal: incorporate **optimality** and **constraints**.

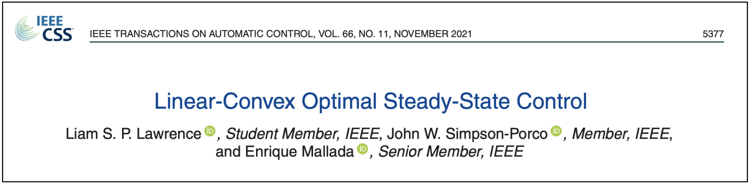
Optimal Steady-State Control

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 66, NO. 11, NOVEMBER 2021 5377

Linear-Convex Optimal Steady-State Control

Liam S. P. Lawrence, *Student Member, IEEE*, John W. Simpson-Porco, *Member, IEEE*, and Enrique Mallada, *Senior Member, IEEE*

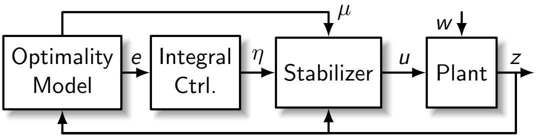
Optimal Steady-State Control



IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 66, NO. 11, NOVEMBER 2021 5377

Linear-Convex Optimal Steady-State Control

Liam S. P. Lawrence, *Student Member, IEEE*, John W. Simpson-Porco, *Member, IEEE*, and Enrique Mallada, *Senior Member, IEEE*



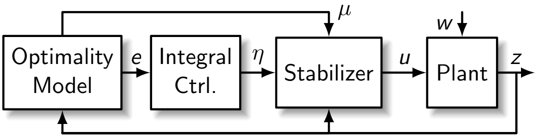
Classical LTI output regulation fused with **optimization**.

Optimal Steady-State Control

IEEE CSS IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 66, NO. 11, NOVEMBER 2021 5377

Linear-Convex Optimal Steady-State Control

Liam S. P. Lawrence, *Student Member, IEEE*, John W. Simpson-Porco, *Member, IEEE*, and Enrique Mallada, *Senior Member, IEEE*



Classical LTI output regulation fused with **optimization**.

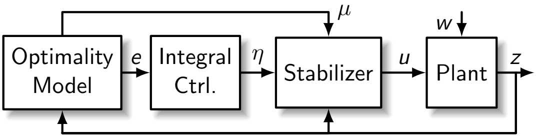
- Dynamic **optimality model** encodes **KKT conditions**
- Integral control regulates KKT error to zero; stabilizer stabilizes

Optimal Steady-State Control

IEEE CSS IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 66, NO. 11, NOVEMBER 2021 5377

Linear-Convex Optimal Steady-State Control

Liam S. P. Lawrence, *Student Member, IEEE*, John W. Simpson-Porco, *Member, IEEE*, and Enrique Mallada, *Senior Member, IEEE*

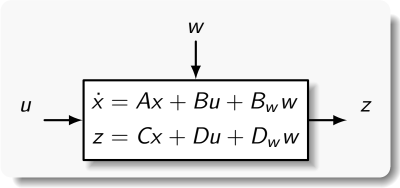


Classical LTI output regulation fused with **optimization**.

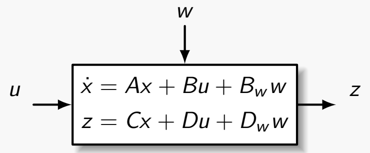
- Dynamic **optimality model** encodes **KKT conditions**
- Integral control regulates KKT error to zero; stabilizer stabilizes

Today: The important case of **exponentially stable LTI** plants.

OSS Control Problem Setup

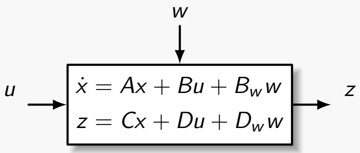


OSS Control Problem Setup



- A Hurwitz, w constant
- $G_u \triangleq -CA^{-1}B + D$
- $G_w \triangleq -CA^{-1}B_w + D_w$

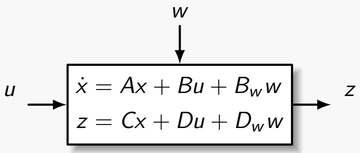
OSS Control Problem Setup



- A Hurwitz, w constant
- $G_u \triangleq -CA^{-1}B + D$
- $G_w \triangleq -CA^{-1}B_w + D_w$

Equilibrium I/O Map: $\bar{z} = G_u \bar{u} + G_w w$

OSS Control Problem Setup

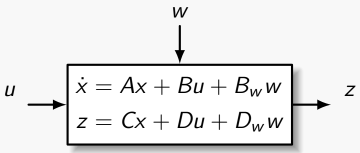


- A Hurwitz, w constant
- $G_u \triangleq -CA^{-1}B + D$
- $G_w \triangleq -CA^{-1}B_w + D_w$

Equilibrium I/O Map: $\bar{z} = G_u \bar{u} + G_w w$

$$
\begin{array}{lll}
\underset{\bar{u}}{\text{minimize}} & f_0(\bar{u}) + g_0(\bar{z}) & \text{(steady-state objective)} \\
\text{subject to} & \bar{z} = G_u \bar{u} + G_w \bar{w} & \text{(steady-state physics)} \\
 & 0 = H_z z + H_u u + H_w w & \text{(design constraints)}
\end{array}
$$

OSS Control Problem Setup



- A Hurwitz, w constant
- $G_u \triangleq -CA^{-1}B + D$
- $G_w \triangleq -CA^{-1}B_w + D_w$

Equilibrium I/O Map: $\bar{z} = G_u \bar{u} + G_w w$

$$
\begin{array}{lll}
\underset{\bar{u}}{\text{minimize}} & f_0(\bar{u}) + g_0(\bar{z}) & \text{(steady-state objective)} \\
\text{subject to} & \bar{z} = G_u \bar{u} + G_w \bar{w} & \text{(steady-state physics)} \\
 & 0 = H_z z + H_u u + H_w w & \text{(design constraints)}
\end{array}
$$

- $f_0 : \mathcal{U} \to \mathbb{R}$ convex, diff.
- $g_0 : \mathcal{Z} \to \mathbb{R}$ convex, diff.
- strictly feasible, $\exists$ opt. solution
- z and $H_z z + H_u u + H_w w$ measured
- $H_z G_u + H_u$ full row rank

Optimality Models for OSS Control

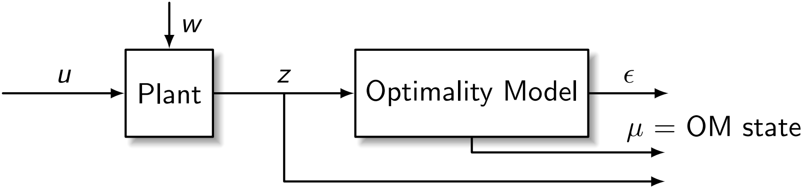
An **optimality model** filters measurements to produce a **proxy error ϵ** quantifying the KKT violation

Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $z = z^\star(w)$.

"Internal Model" Interpretation: The loop gain *incorporates a model of the optimal solution set*

Optimality Models for OSS Control

An **optimality model** filters measurements to produce a **proxy error** ϵ quantifying the KKT violation

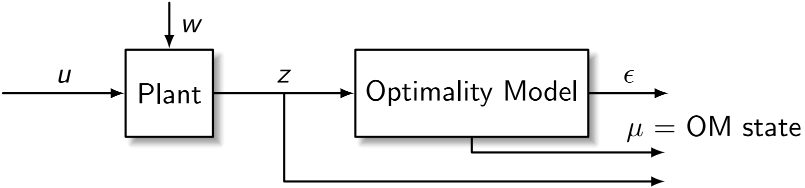


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $z = z^{\star}(w)$.

"Internal Model" Interpretation: The loop gain *incorporates a model of the optimal solution set*

Optimality Models for OSS Control

An **optimality model** filters measurements to produce a **proxy error** ϵ quantifying the KKT violation

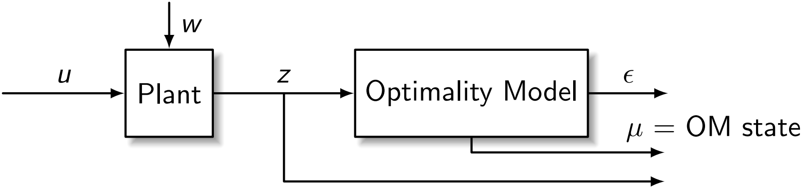


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $z = z^{\star}(w)$.

"Internal Model" Interpretation: The loop gain *incorporates a model of the optimal solution set*

Optimality Models for OSS Control

An **optimality model** filters measurements to produce a **proxy error** ϵ quantifying the KKT violation

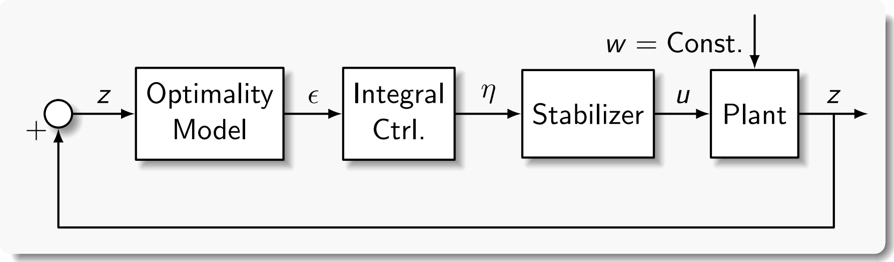


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $z = z^\star(w)$.

"Internal Model" Interpretation: The loop gain *incorporates a model of the optimal solution set*

From OSS Control to Regulator Problem

Optimality model reduces OSS control to regulator/servomechanism problem



Optimality Model: creates proxy error signal ϵ

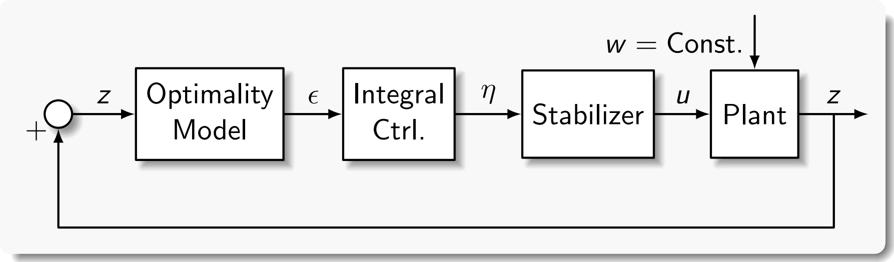
Integral Control: integrates ϵ

Stabilizing Controller: stabilizes the cascade

Theorem: Closed-Loop Stability $\implies$ $z(t) \to z^\star(w)$

From OSS Control to Regulator Problem

Optimality model reduces OSS control to regulator/servomechanism problem



Optimality Model: creates proxy error signal ϵ

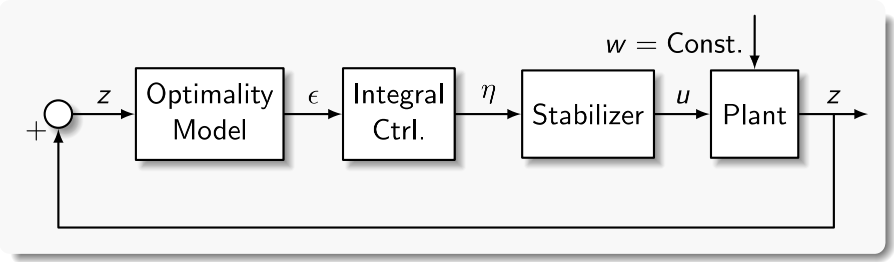
Integral Control: integrates ϵ

Stabilizing Controller: stabilizes the cascade

Theorem: Closed-Loop Stability $\implies$ $z(t) \to z^\star(w)$

From OSS Control to Regulator Problem

Optimality model reduces OSS control to regulator/servomechanism problem



Optimality Model: creates proxy error signal ϵ

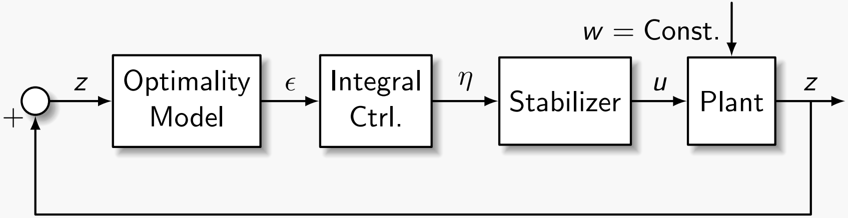
Integral Control: integrates ϵ

Stabilizing Controller: stabilizes the cascade

Theorem: Closed-Loop Stability $\implies$ $z(t) \to z^{\star}(w)$

From OSS Control to Regulator Problem

Optimality model reduces OSS control to regulator/servomechanism problem



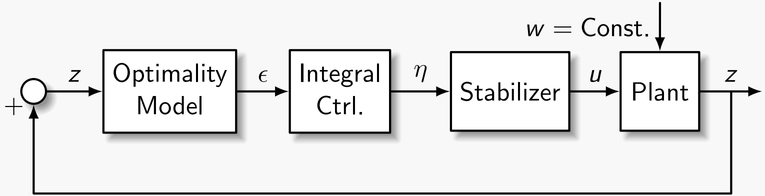
Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes the cascade

Theorem: Closed-Loop Stability $\implies$ $\boldsymbol{z}(t) \to \boldsymbol{z}^{\star}(\boldsymbol{w})$

Sanity Check: Existence of Stabilizer for Quadratic Obj.

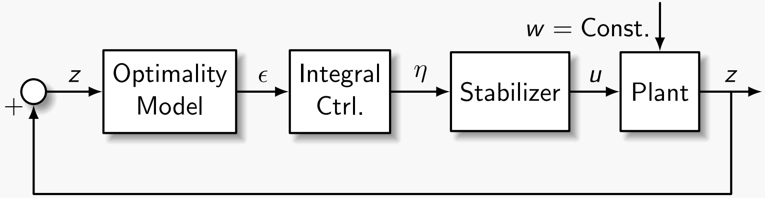


- **Convex quadratic objective** $\begin{bmatrix} z \\ u \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} z \\ u \end{bmatrix} + c^{\mathsf{T}} \begin{bmatrix} z \\ u \end{bmatrix}$

Plant→OM→Integrator **cascade is stabilizable/detectable** $\Longleftrightarrow$

1. plant stabilizable/detectable;
2. optimization problem has a unique solution;
3. $H_z G_u + H_u$ full row rank;
4. full rank conditions on optimality model.

Sanity Check: Existence of Stabilizer for Quadratic Obj.

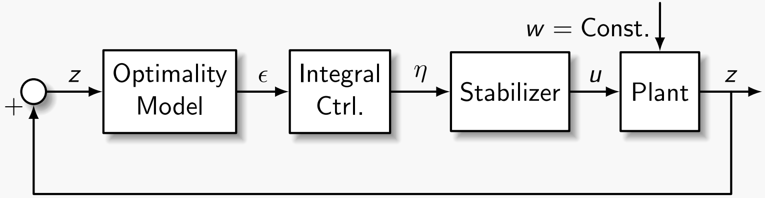


- **Convex quadratic objective** $\begin{bmatrix} z \\ u \end{bmatrix}^\mathsf{T} Q \begin{bmatrix} z \\ u \end{bmatrix} + c^\mathsf{T} \begin{bmatrix} z \\ u \end{bmatrix}$

Plant→OM→Integrator **cascade is stabilizable/detectable** $\iff$

1. plant stabilizable/detectable;
2. optimization problem has a unique solution;
3. $H_z G_u + H_u$ full row rank;
4. full rank conditions on optimality model.

Sanity Check: Existence of Stabilizer for Quadratic Obj.

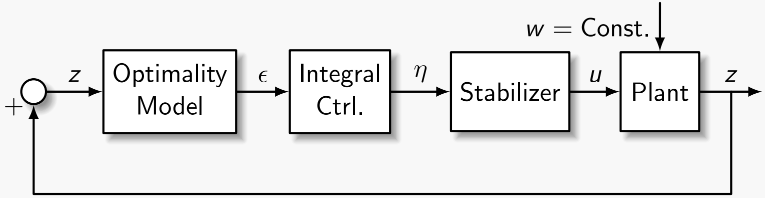


- **Convex quadratic objective** $\begin{bmatrix} z \\ u \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} z \\ u \end{bmatrix} + c^{\mathsf{T}} \begin{bmatrix} z \\ u \end{bmatrix}$

Plant→OM→Integrator **cascade is stabilizable/detectable** $\iff$

1. plant stabilizable/detectable;
2. optimization problem has a unique solution;
3. $H_z G_u + H_u$ full row rank;
4. full rank conditions on optimality model.

Sanity Check: Existence of Stabilizer for Quadratic Obj.

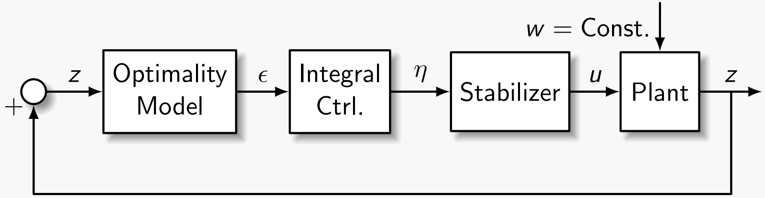


- **Convex quadratic objective** $\begin{bmatrix} z \\ u \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} z \\ u \end{bmatrix} + c^{\mathsf{T}} \begin{bmatrix} z \\ u \end{bmatrix}$

Plant→OM→Integrator **cascade is stabilizable/detectable** $\iff$

1. plant stabilizable/detectable;
2. optimization problem has a unique solution;
3. $H_z G_u + H_u$ full row rank;
4. full rank conditions on optimality model.

Sanity Check: Existence of Stabilizer for Quadratic Obj.



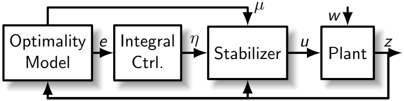
- **Convex quadratic objective** $\begin{bmatrix} z \\ u \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} z \\ u \end{bmatrix} + c^{\mathsf{T}} \begin{bmatrix} z \\ u \end{bmatrix}$

Plant$\to$OM$\to$Integrator **cascade is stabilizable/detectable** $\iff$

1. plant stabilizable/detectable;
2. optimization problem has a unique solution;
3. $H_z G_u + H_u$ full row rank;
4. full rank conditions on optimality model.

Examples of Optimality Models

$$\begin{aligned} \underset{\bar{u}}{\text{minimize}} \quad & f_0(\bar{u}) + g_0(\bar{z}) \\ \text{subject to} \quad & \bar{z} = G_u \bar{u} + G_w \bar{w} \\ & 0 = H_z z + H_u u + H_w w \end{aligned}$$

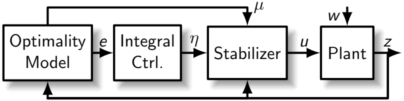


Optimality Model #1

Optimality Model #2

Examples of Optimality Models

$$
\begin{aligned}
\underset{\bar{u}}{\text{minimize}} \quad & f_0(\bar{u}) + g_0(\bar{z}) \\
\text{subject to} \quad & \bar{z} = G_u \bar{u} + G_w \bar{w} \\
& 0 = H_z z + H_u u + H_w w
\end{aligned}
$$



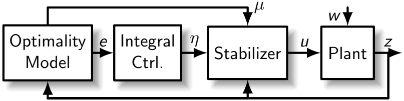
Optimality Models ≈ KKT conditions driven by measurements.

Optimality Model #1

Optimality Model #2

Examples of Optimality Models

$$\begin{aligned} \underset{\bar{u}}{\text{minimize}} \quad & f_0(\bar{u}) + g_0(\bar{z}) \\ \text{subject to} \quad & \bar{z} = G_u \bar{u} + G_w \bar{w} \\ & 0 = H_z z + H_u u + H_w w \end{aligned}$$



Optimality Models ≈ KKT conditions driven by measurements.

Optimality Model #1

- Dualize engineering constraint

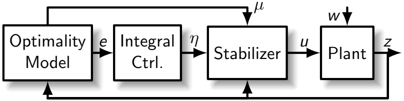
$$\tau \dot{\mu} = H_z z + H_u u + H_w w$$

$$e = \nabla f_0(u) + G_u^{\mathsf{T}} \nabla g_0(z) + (H_z G_u + H_u)^{\mathsf{T}} \mu$$

Optimality Model #2

Examples of Optimality Models

$$\begin{aligned} \underset{\bar{u}}{\text{minimize}} \quad & f_0(\bar{u}) + g_0(\bar{z}) \\ \text{subject to} \quad & \bar{z} = G_u \bar{u} + G_w \bar{w} \\ & 0 = H_z z + H_u u + H_w w \end{aligned}$$



Optimality Models $\approx$ KKT conditions driven by measurements.

Optimality Model #1

- Dualize engineering constraint

$$\tau \dot{\mu} = H_z z + H_u u + H_w w$$

$$e = \nabla f_0(u) + G_u^{\mathsf{T}} \nabla g_0(z) + (H_z G_u + H_u)^{\mathsf{T}} \mu$$

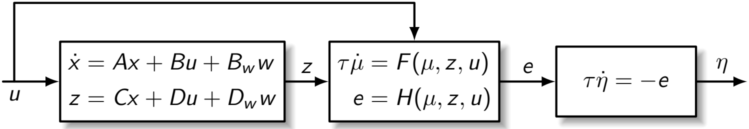
Optimality Model #2

- Parameterize intersection of equality constraints in (z, u):

$$\text{range} \begin{bmatrix} T_z \\ T_u \end{bmatrix} = \text{null} \begin{bmatrix} I_r & -G_u \\ H_z & H_u \end{bmatrix}$$

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^{\mathsf{T}} \nabla f_0(u) + T_z^{\mathsf{T}} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}.$$

Stabilizers via Low-Gain Integral Control

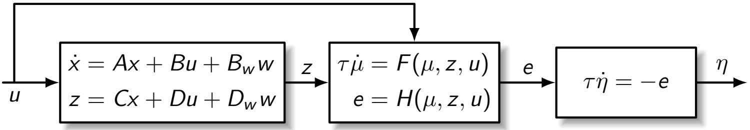


- We now aim to **stabilize** the cascade.
- We look for simple stabilizers of the form $u = k(\eta, z, \mu)$.
- **Low-gain feedback:** When $\tau \gg 1$, plant is fast compared to controller; replace plant with equilibrium I/O map
- Closed-loop stability determined by **reduced dynamics**

$$\dot{\mu} = F(\mu, G_u u + G_w w, u)$$
$$\dot{\eta} = -H(\mu, G_u u + G_w w, u)$$

JWSP. "*Analysis and Synthesis of Low-Gain Integral Controllers for Nonlinear Systems*". TAC 2021

Stabilizers via Low-Gain Integral Control

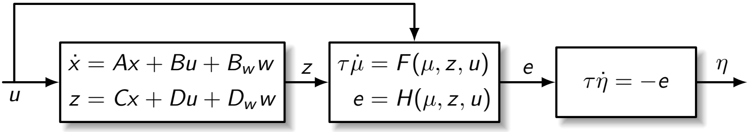


- We now aim to **stabilize** the cascade.
- We look for simple stabilizers of the form $u = k(\eta, z, \mu)$.
- **Low-gain feedback:** When $\tau \gg 1$, plant is fast compared to controller; replace plant with equilibrium I/O map
- Closed-loop stability determined by **reduced dynamics**

$$\dot{\mu} = F(\mu, G_u u + G_w w, u)$$
$$\dot{\eta} = -H(\mu, G_u u + G_w w, u)$$

JWSP. "*Analysis and Synthesis of Low-Gain Integral Controllers for Nonlinear Systems*". TAC 2021

Stabilizers via Low-Gain Integral Control

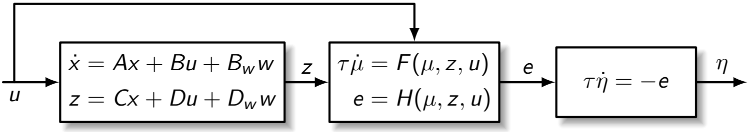


- We now aim to **stabilize** the cascade.
- We look for simple stabilizers of the form $u = k(\eta, z, \mu)$.
- **Low-gain feedback:** When $\tau \gg 1$, plant is fast compared to controller; replace plant with equilibrium I/O map
- Closed-loop stability determined by **reduced dynamics**

$$\dot{\mu} = F(\mu, G_u u + G_w w, u)$$
$$\dot{\eta} = -H(\mu, G_u u + G_w w, u)$$

JWSP. "*Analysis and Synthesis of Low-Gain Integral Controllers for Nonlinear Systems*". TAC 2021

Stabilizers via Low-Gain Integral Control

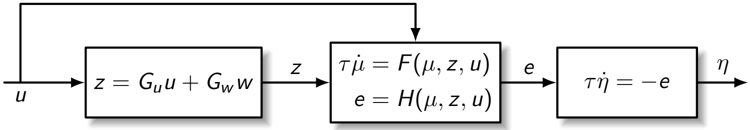


- We now aim to **stabilize** the cascade.
- We look for simple stabilizers of the form $u = k(\eta, z, \mu)$.
- **Low-gain feedback:** When $\tau \gg 1$, plant is fast compared to controller; replace plant with equilibrium I/O map
- Closed-loop stability determined by **reduced dynamics**

$$\dot{\mu} = F(\mu, G_u u + G_w w, u)$$
$$\dot{\eta} = -H(\mu, G_u u + G_w w, u)$$

JWSP. "*Analysis and Synthesis of Low-Gain Integral Controllers for Nonlinear Systems*". TAC 2021

Stabilizers via Low-Gain Integral Control

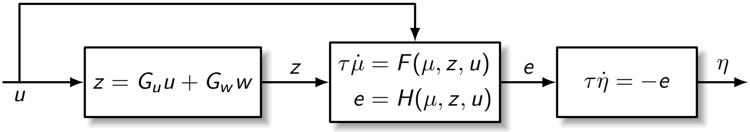


- We now aim to **stabilize** the cascade.
- We look for simple stabilizers of the form $u = k(\eta, z, \mu)$.
- **Low-gain feedback:** When $\tau \gg 1$, plant is fast compared to controller; replace plant with equilibrium I/O map
- Closed-loop stability determined by **reduced dynamics**

$$\dot{\mu} = F(\mu, G_u u + G_w w, u)$$
$$\dot{\eta} = -H(\mu, G_u u + G_w w, u)$$

JWSP. "*Analysis and Synthesis of Low-Gain Integral Controllers for Nonlinear Systems*". TAC 2021

Stabilizers via Low-Gain Integral Control



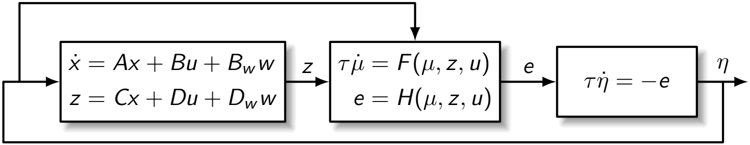
- We now aim to **stabilize** the cascade.
- We look for simple stabilizers of the form $u = k(\eta, z, \mu)$.
- **Low-gain feedback:** When $\tau \gg 1$, plant is fast compared to controller; replace plant with equilibrium I/O map
- Closed-loop stability determined by **reduced dynamics**

$$\begin{aligned}\dot{\mu} &= F(\mu, G_u u + G_w w, u) \\ \dot{\eta} &= -H(\mu, G_u u + G_w w, u)\end{aligned}$$

JWSP. "*Analysis and Synthesis of Low-Gain Integral Controllers for Nonlinear Systems*". TAC 2021

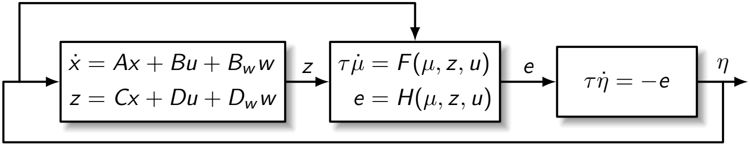
Optimality Model #1: Primal-Dual Stabilizer

$$(PD): \quad \begin{aligned} \tau\dot{\mu} &= H_z z + H_u u + H_w w \\ e &= \nabla f_0(u) + G_u^\mathsf{T} \nabla g_0(z) + (H_z G_u + H_u)^\mathsf{T} \mu \\ \tau\dot{\eta} &= -e \\ u &= \eta \end{aligned}$$



Optimality Model #1: Primal-Dual Stabilizer

$$
(PD): \quad
\begin{aligned}
\tau\dot{\mu} &= H_z z + H_u u + H_w w \\
e &= \nabla f_0(u) + G_u^{\mathsf{T}} \nabla g_0(z) + (H_z G_u + H_u)^{\mathsf{T}} \mu \\
\tau\dot{\eta} &= -e \\
u &= \eta
\end{aligned}
$$



Theorem: If $u \mapsto f_0(u) + g_0(G_u u)$ is strongly convex on $\mathcal{U}$, then there exists $\tau^\star > 0$ such that for all $\tau > \tau^\star$, the controller (PD) solves the OSS control problem.

Optimality Model #1: Inversion-Based Stabilizer

- What if we **directly solve for error-zeroing** u:

$$0 = e = \nabla f_0(u) + G_u^\mathsf{T} \nabla g_0(z) + (H_z G_u + H_u)^\mathsf{T} \mu$$

- When is ∇f_0 invertible? When f_0 is strongly convex and *essentially smooth*, i.e., it blows up at the boundary of $\mathcal{U}$.

Optimality Model #1: Inversion-Based Stabilizer

- What if we **directly solve for error-zeroing** u:

$$0 = e = \nabla f_0(u) + G_u^{\mathsf{T}} \nabla g_0(z) + (H_z G_u + H_u)^{\mathsf{T}} \mu$$

- When is ∇f_0 invertible? When f_0 is strongly convex and *essentially smooth*, i.e., it blows up at the boundary of $\mathcal{U}$.

Optimality Model #1: Inversion-Based Stabilizer

- What if we **directly solve for error-zeroing** u:

$$0 = e = \nabla f_0(u) + G_u^\mathsf{T} \nabla g_0(z) + (H_z G_u + H_u)^\mathsf{T} \mu$$

- When is ∇f_0 invertible? When f_0 is strongly convex and *essentially smooth*, i.e., it blows up at the boundary of $\mathcal{U}$.

$$(Inv): \qquad \begin{aligned} \tau \dot{\mu} &= H_z z + H_u u + H_w w \\ u &= (\nabla f_0)^{-1}(-G_u^\mathsf{T} \nabla g_0(z) - (H_z G_u + H_u)^\mathsf{T} \mu) \end{aligned}$$

Optimality Model #1: Inversion-Based Stabilizer

- What if we **directly solve for error-zeroing** u:

$$0 = e = \nabla f_0(u) + G_u^{\mathsf{T}} \nabla g_0(z) + (H_z G_u + H_u)^{\mathsf{T}} \mu$$

- When is ∇f_0 invertible? When f_0 is strongly convex and *essentially smooth*, i.e., it blows up at the boundary of $\mathcal{U}$.

$$(Inv): \quad \begin{aligned} \tau \dot{\mu} &= H_z z + H_u u + H_w w \\ u &= (\nabla f_0)^{-1}(-G_u^{\mathsf{T}} \nabla g_0(z) - (H_z G_u + H_u)^{\mathsf{T}} \mu) \end{aligned}$$

Theorem: If f_0 is strongly convex and essentially smooth on $\mathcal{U}$, then there exists $\tau^\star > 0$ such that for all $\tau > \tau^\star$, the controller (Inv) solves the OSS control problem.

Optimality Model #2: Two-Loop Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^{\mathsf{T}} \nabla f_0(u) + T_z^{\mathsf{T}} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Optimality Model #2: Two-Loop Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^\mathsf{T} \nabla f_0(u) + T_z^\mathsf{T} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(TL): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$

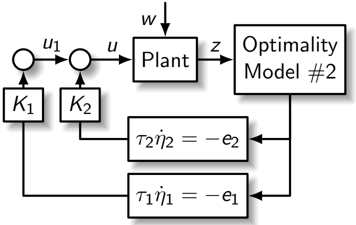
Optimality Model #2: Two-Loop Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^{\mathsf{T}} \nabla f_0(u) + T_z^{\mathsf{T}} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(TL): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$



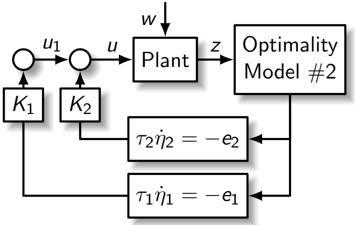
Optimality Model #2: Two-Loop Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^\mathsf{T} \nabla f_0(u) + T_z^\mathsf{T} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(TL): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$



Theorem:

(i) Let $N = H_z G_u + H_u$

(ii) Choose K_2 s.t. $-NK_2$ Hurwitz

(iii) Let $\Pi = I - K_2(NK_2)^{-1}N$

(iv) Choose K_1 s.t. $\Pi K_1 = T_u$

If

$$\xi \mapsto f_0(T_u\xi) + g_0(T_z\xi)$$

is strongly convex, then (TL) solves the OSS control problem for all

$$\tau_1 \gg \tau_2 \gg 0.$$

Optimality Model #2: Robust/Optimal Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^{\mathsf{T}} \nabla f_0(u) + T_z^{\mathsf{T}} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(RO): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$

The design of K_1, K_2 can be formulated as a **robust state-feedback design problem** for the **reduced dynamics**.

- ∇f_0 and ∇g_0 are monotone/slope-restricted nonlinearities
- minimize induced $\mathscr{L}_2$ norm from w to (e_1, e_2)
- convexified via "dualization lemma"

Optimality Model #2: Robust/Optimal Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^\mathsf{T} \nabla f_0(u) + T_z^\mathsf{T} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(RO): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$

The design of K_1, K_2 can be formulated as a **robust state-feedback design problem** for the **reduced dynamics**.

- ∇f_0 and ∇g_0 are monotone/slope-restricted nonlinearities
- minimize induced $\mathscr{L}_2$ norm from w to (e_1, e_2)
- convexified via "dualization lemma"

Optimality Model #2: Robust/Optimal Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^{\mathsf{T}} \nabla f_0(u) + T_z^{\mathsf{T}} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(RO): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$

The design of K_1, K_2 can be formulated as a **robust state-feedback design problem** for the **reduced dynamics**.

- ∇f_0 and ∇g_0 are monotone/slope-restricted nonlinearities
- minimize induced $\mathscr{L}_2$ norm from w to (e_1, e_2)
- convexified via "dualization lemma"

Optimality Model #2: Robust/Optimal Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^\mathsf{T} \nabla f_0(u) + T_z^\mathsf{T} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(RO): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$

The design of K_1, K_2 can be formulated as a **robust state-feedback design problem** for the **reduced dynamics**.

- ∇f_0 and ∇g_0 are monotone/slope-restricted nonlinearities
- minimize induced $\mathscr{L}_2$ norm from w to (e_1, e_2)
- convexified via "dualization lemma"

Optimality Model #2: Robust/Optimal Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^{\mathsf{T}} \nabla f_0(u) + T_z^{\mathsf{T}} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(RO): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$

The design of K_1, K_2 can be formulated as a **robust state-feedback design problem** for the **reduced dynamics**.

- ∇f_0 and ∇g_0 are monotone/slope-restricted nonlinearities
- minimize induced $\mathscr{L}_2$ norm from w to (e_1, e_2)
- convexified via "dualization lemma"

Optimality Model #2: Robust/Optimal Stabilizer

Optimality Model:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} T_u^{\mathsf{T}} \nabla f_0(u) + T_z^{\mathsf{T}} \nabla g_0(z) \\ H_z z + H_u u + H_w w \end{bmatrix}$$

Stabilizer:

$$(RO): \quad \begin{aligned} \tau_1 \dot{\eta}_1 &= -e_1 \\ \tau_2 \dot{\eta}_2 &= -e_2 \\ u &= K_1 \eta_1 + K_2 \eta_2 \end{aligned}$$

The design of K_1, K_2 can be formulated as a **robust state-feedback design problem** for the **reduced dynamics**.

- ∇f_0 and ∇g_0 are monotone/slope-restricted nonlinearities
- minimize induced $\mathscr{L}_2$ norm from w to (e_1, e_2)
- convexified via "dualization lemma"

Theorem: An optimal selection of K_1, K_2 can be found by solving an SDP; see paper. Under Lipschitz assumptions, the SDP is always feasible, since (TL) is a feasible point.

Simulation: 30 states, 7 inputs, 5 outputs

- **Objectives:** (z_1, z_2) step tracking, min. control, constraints

$$\begin{aligned} \text{minimize} \quad & \left[\sum_{k=1}^{m} \tfrac{1}{2}\bar{u}_k^2 + \gamma \mathrm{B}(\bar{u}_k)\right] + c\,\mathrm{P}(\bar{z}) \\ \text{subject to} \quad & \bar{z} = G_u \bar{u} + G_w w \\ & 0 = z_i - r_i, \qquad i \in \{1, 2\} \end{aligned}$$

$$\mathrm{B}(u_k) = \text{log barrier fcn.}$$

$$\mathrm{P}(\bar{z}_3, \bar{z}_4, \bar{z}_5) = \text{penalty fcn.}$$

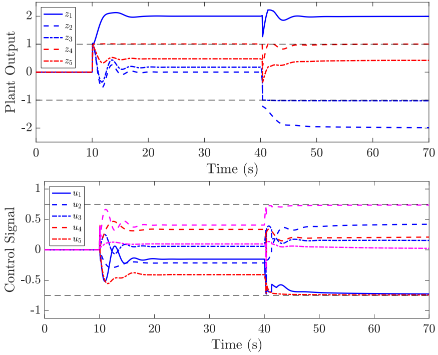
Simulation: 30 states, 7 inputs, 5 outputs

- **Objectives:** (z_1, z_2) step tracking, min. control, constraints

$$\begin{aligned} \text{minimize} \quad & \Big[\sum_{k=1}^{m} \tfrac{1}{2}\bar{u}_k^2 + \gamma \mathrm{B}(\bar{u}_k)\Big] + c\,\mathrm{P}(\bar{z}) \\ \text{subject to} \quad & \bar{z} = G_u \bar{u} + G_w w \\ & 0 = z_i - r_i, \qquad i \in \{1,2\} \end{aligned}$$

$$\mathrm{B}(u_k) = \text{log barrier fcn.}$$
$$\mathrm{P}(\bar{z}_3, \bar{z}_4, \bar{z}_5) = \text{penalty fcn.}$$

Inversion Stabilizer $\tau = 2$



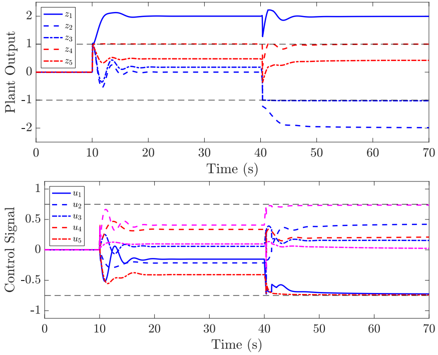
Simulation: 30 states, 7 inputs, 5 outputs

- **Objectives:** (z_1, z_2) step tracking, min. control, constraints

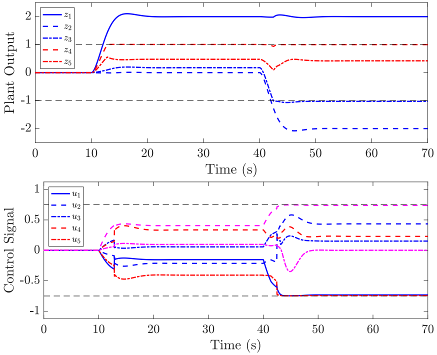
$$\begin{aligned} \text{minimize} \quad & \left[\sum_{k=1}^{m} \tfrac{1}{2}\bar{u}_k^2 + \gamma \mathrm{B}(\bar{u}_k)\right] + c\,\mathrm{P}(\bar{z}) \\ \text{subject to} \quad & \bar{z} = G_u \bar{u} + G_w w \\ & 0 = z_i - r_i, \qquad i \in \{1, 2\} \end{aligned}$$

$$\mathrm{B}(u_k) = \text{log barrier fcn.}$$
$$\mathrm{P}(\bar{z}_3, \bar{z}_4, \bar{z}_5) = \text{penalty fcn.}$$

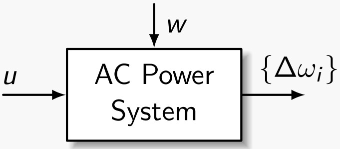
Inversion Stabilizer $\tau = 2$



Optimal/Robust Stabilizer



Application: Secondary Frequency Control in Bulk Grid



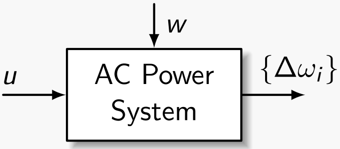
$$
\begin{array}{ll}
\underset{u}{\text{minimize}} & \sum_{i=1}^{n} C_i(\bar{u}_i) \\
\text{subject to} & \Delta\boldsymbol{\omega} = \frac{1}{\beta}\mathbb{1}_n(\mathbb{1}_n^{\mathsf{T}}\bar{u} + w) \\
 & \Delta\omega_n = 0
\end{array}
$$

- **Note:** the way one encodes the frequency regulation constraint can lead to different controller architectures; this is just one possible choice.
- $G_u = \frac{1}{\beta}\mathbb{1}_n\mathbb{1}_n^{\mathsf{T}}$, $G_w = \frac{1}{\beta}\mathbb{1}_n$

Optimality Model #1

Optimality Model #2

Application: Secondary Frequency Control in Bulk Grid



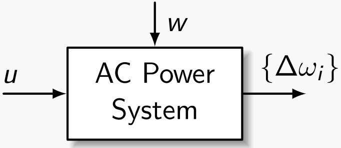
$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \sum_{i=1}^{n} C_i(\bar{u}_i) \\
\text{subject to} \quad & \Delta\boldsymbol{\omega} = \tfrac{1}{\beta}\mathbb{1}_n(\mathbb{1}_n^{\mathsf{T}}\bar{u} + w) \\
& \Delta\omega_n = 0
\end{aligned}
$$

- **Note:** the way one encodes the frequency regulation constraint can lead to different controller architectures; this is just one possible choice.
- $G_u = \frac{1}{\beta}\mathbb{1}_n\mathbb{1}_n^{\mathsf{T}}$, $G_w = \frac{1}{\beta}\mathbb{1}_n$

Optimality Model #1

Optimality Model #2

Application: Secondary Frequency Control in Bulk Grid



$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \sum_{i=1}^{n} C_i(\bar{u}_i) \\ \text{subject to} \quad & \Delta\boldsymbol{\omega} = \tfrac{1}{\beta}\mathbb{1}_n(\mathbb{1}_n^{\mathsf{T}}\bar{u} + w) \\ & \Delta\omega_n = 0 \end{aligned}$$

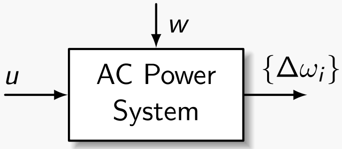
- **Note:** the way one encodes the frequency regulation constraint can lead to different controller architectures; this is just one possible choice.
- $G_u = \frac{1}{\beta}\mathbb{1}_n\mathbb{1}_n^{\mathsf{T}}$, $G_w = \frac{1}{\beta}\mathbb{1}_n$

Optimality Model #1

$$\begin{aligned} \tau\dot{\mu} &= -\Delta\omega_n \\ e_i &= \nabla C_i(u_i) - \mu \end{aligned}$$

Optimality Model #2

Application: Secondary Frequency Control in Bulk Grid



$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \sum\nolimits_{i=1}^{n} C_i(\bar{u}_i) \\ \text{subject to} \quad & \Delta\boldsymbol{\omega} = \tfrac{1}{\beta}\mathbb{1}_n(\mathbb{1}_n^{\mathsf{T}}\bar{u} + w) \\ & \Delta\omega_n = 0 \end{aligned}$$

- **Note:** the way one encodes the frequency regulation constraint can lead to different controller architectures; this is just one possible choice.
- $G_u = \frac{1}{\beta}\mathbb{1}_n\mathbb{1}_n^{\mathsf{T}}$, $G_w = \frac{1}{\beta}\mathbb{1}_n$

Optimality Model #1

$$\begin{aligned} \tau\dot{\mu} &= -\Delta\omega_n \\ e_i &= \nabla C_i(u_i) - \mu \end{aligned}$$

Optimality Model #2

$$\text{null}\begin{bmatrix} I_n & -\frac{1}{\beta}\mathbb{1}_n\mathbb{1}_n^{\mathsf{T}} \\ \mathrm{e}_n^{\mathsf{T}} & 0 \end{bmatrix} = \text{range}\begin{bmatrix} 0 \\ L^{\mathsf{T}} \end{bmatrix}$$

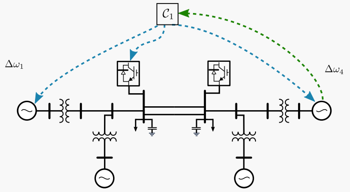
$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} L\nabla C(u) \\ \Delta\omega_n \end{bmatrix}.$$

Application: Secondary Frequency Control in Bulk Grid

Both designs provably stable for large τ

1. Inversion-based controller

$$\tau \dot{\mu} = -\Delta\omega_n$$
$$u_i = (\nabla C_i)^{-1}(\mu)$$

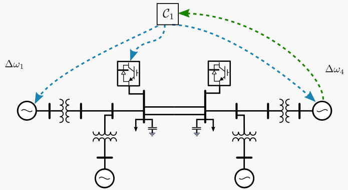


Application: Secondary Frequency Control in Bulk Grid

Both designs provably stable for large τ

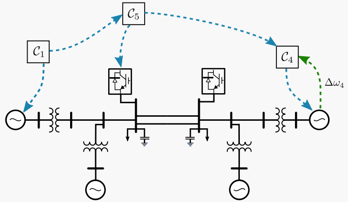
1. Inversion-based controller

$$\tau \dot{\mu} = -\Delta\omega_n$$
$$u_i = (\nabla C_i)^{-1}(\mu)$$



2. Two-loop controller

$$\tau_1 \dot{\eta}_i = -\sum_{j=1}^{n} a_{ij}(\nabla C_i(u_i) - \nabla C_j(u_j))$$
$$\tau_2 \dot{\eta}_n = -\Delta\omega_n$$
$$u_i = \eta_i$$



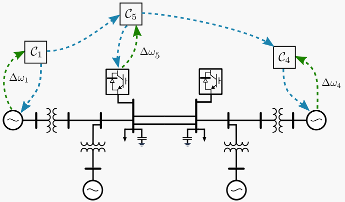
Example #3: Secondary Frequency Control in Bulk Grid

Both designs provably stable for large τ

1. **Distributed** consensus-based approach:

$$\tau_i \dot{\eta}_i = -\Delta\omega_i - \sum_{j=1}^{n} a_{ij}(\eta_i - \eta_j)$$

$$P_i^{\mathrm{s}} = (\nabla C_i)^{-1}(\eta_i)$$



Conclusions

Two frameworks for feedback-based optimization

1. Gradient-type algorithms (leverage opt. theory + robust ctrl)
2. Optimal steady-state control (leverage servomech. theory)

Opportunities at $\{$**control**$\} \cap \{$**optimization**$\} \cap \{$**power systems**$\} \cap \cdots$

1. High-performance optimizing designs (e.g., loop-shaping, feedforward, anti-windup)
2. Hierarchical, competitive multi-agent, learning-based, ...

Conclusions

Two frameworks for feedback-based optimization

1. Gradient-type algorithms (leverage opt. theory + robust ctrl)
2. Optimal steady-state control (leverage servomech. theory)

Opportunities at $\{$**control**$\} \cap \{$**optimization**$\} \cap \{$**power systems**$\} \cap \cdots$

1. High-performance optimizing designs (e.g., loop-shaping, feedforward, anti-windup)
2. Hierarchical, competitive multi-agent, learning-based, . . .

Conclusions

Two frameworks for feedback-based optimization

1. Gradient-type algorithms (leverage opt. theory + robust ctrl)
2. Optimal steady-state control (leverage servomech. theory)

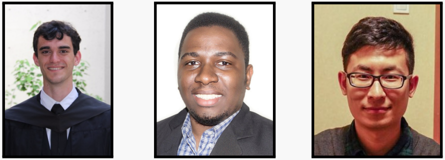
Opportunities at $\{\textbf{control}\} \cap \{\textbf{optimization}\} \cap \{\textbf{power systems}\} \cap \cdots$

1. High-performance optimizing designs (e.g., loop-shaping, feedforward, anti-windup)
2. Hierarchical, competitive multi-agent, learning-based, ...

Note: Open PhD position at University of Toronto for Fall 2023, focusing on data-driven control and estimation for energy systems!

Collaborators

Group: Liam Lawrence, Etinosa Ekomwenrenren, Zhiyuan Tang



EPRI: Evangelos Farantatos, Mahendra Patel, Hossein Hooshyar
NREL: Marcello Colombino, Andrey Bernstein **JHU:** Enrique Mallada

Questions

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering

UNIVERSITY OF TORONTO

https://www.control.utoronto.ca/~jwsimpson/

jwsimpson@ece.utoronto.ca

Temporary page!

LaTeX was unable to guess the total number of pages correctly. A
was some unprocessed data that should have been added to the
this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page
away, because LaTeX now knows how many pages to expect for t
document.