Frameworks for Real-Time Feedback-Based Optimization

(with Applications in Energy Systems)

John W. Simpson-Porco
https://ece.uwaterloo.ca/~jwsimpso/

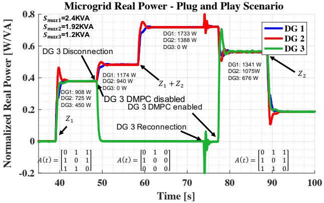
UNIVERSITY OF WATERLOO

Energy-Open, Groningen, Netherlands

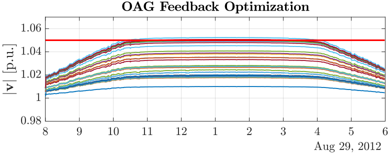
November 10, 2019

Research in control and optimization of energy systems

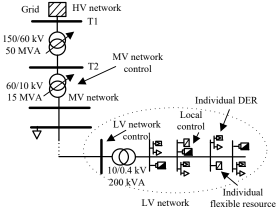
Predictive Control (MPC) of Microgrids



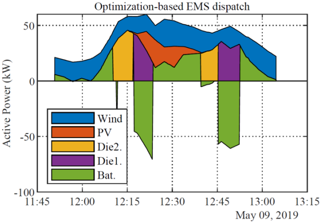
Real-Time Optimization of Feeders with High PV Penetration



MPC for Active Distribution Systems

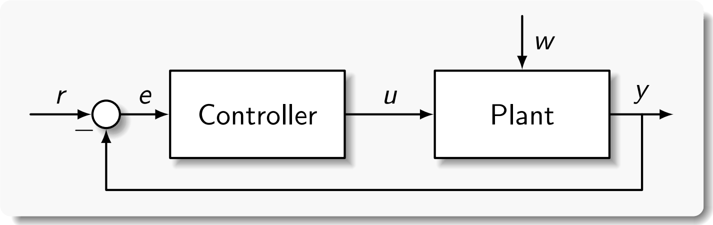


Microgrid Energy Management System (w/ Canadian Solar, Guelph, ON)



Control Systems 101

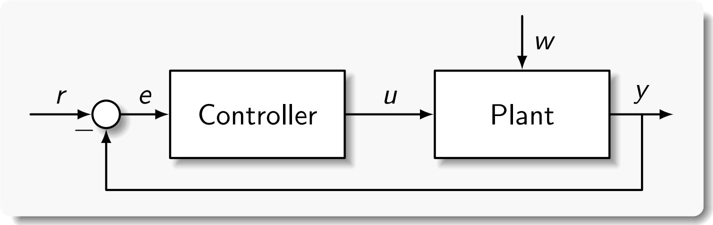
- Prototypical feedback control problem is **tracking** and **disturbance rejection** in the presence of **plant uncertainty**



Where does the reference r come from?

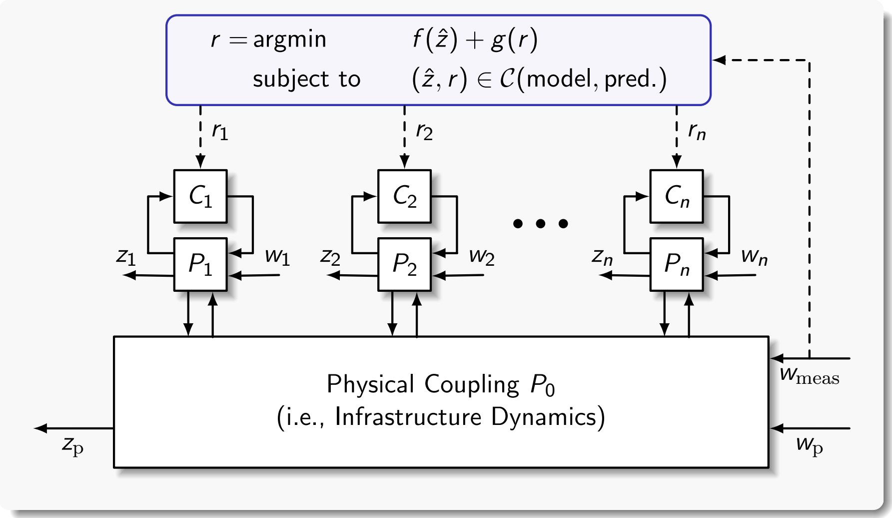
Control Systems 101

- Prototypical feedback control problem is **tracking** and **disturbance rejection** in the presence of **plant uncertainty**

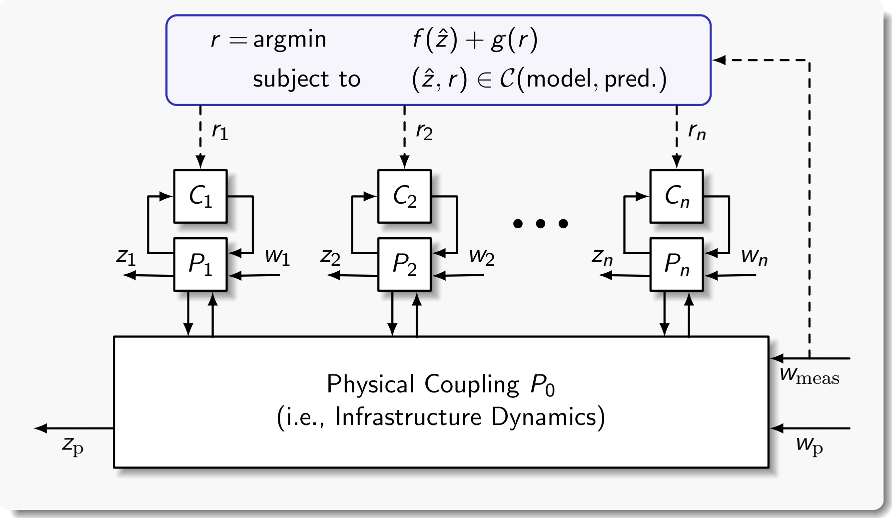


Where does the reference r come from?

Feedforward Optimization for Complex Control Systems

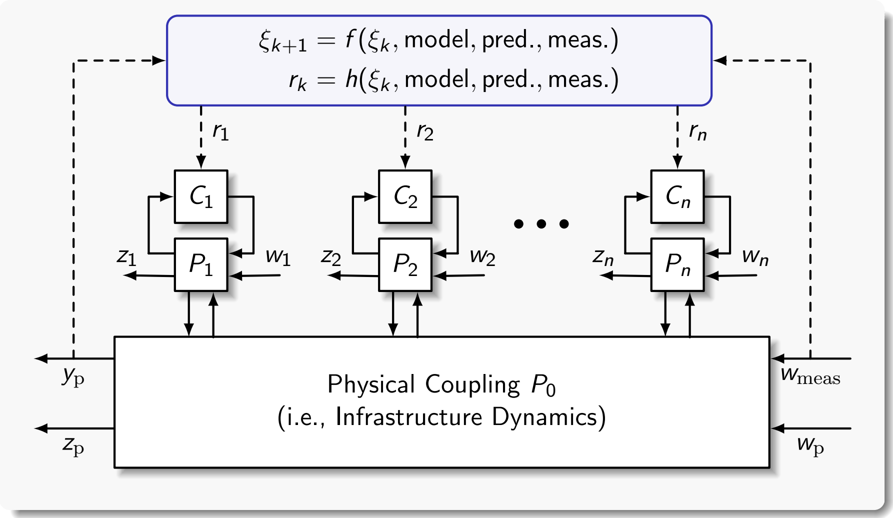


Feedforward Optimization for Complex Control Systems

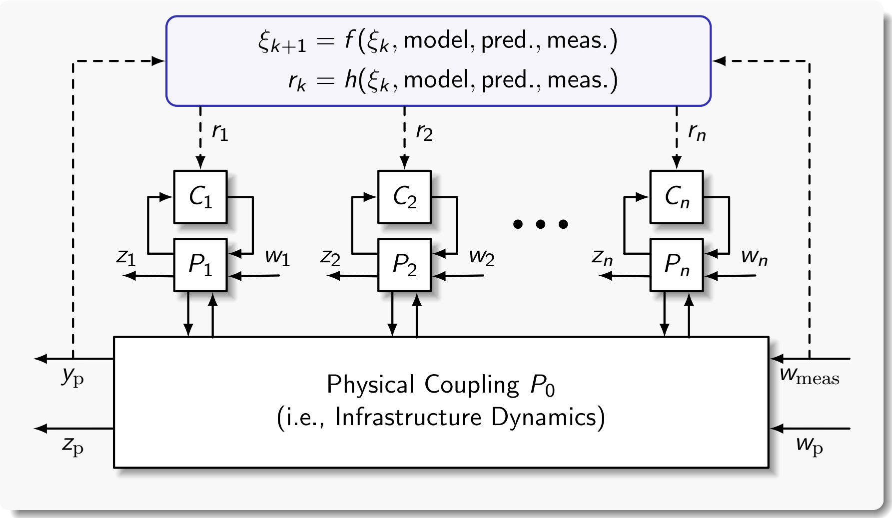


Feedforward: simple, but **sensitive to uncertainty**

Feedback Optimization for Complex Control Systems

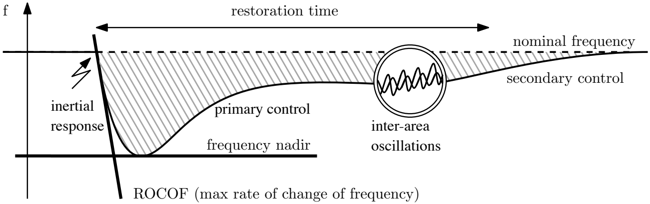


Feedback Optimization for Complex Control Systems



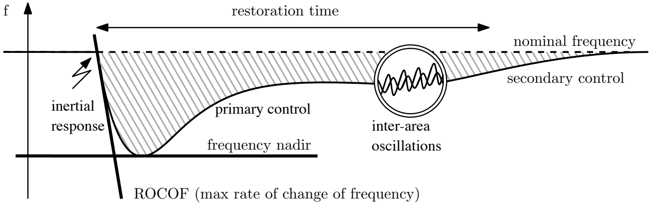
Feedback: improved **robustness / disturbance attenuation**

Example #1: Secondary Frequency Control in Bulk Grid

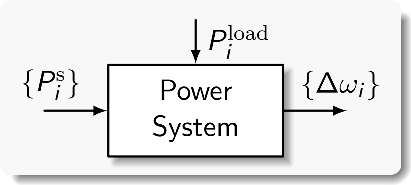


- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$

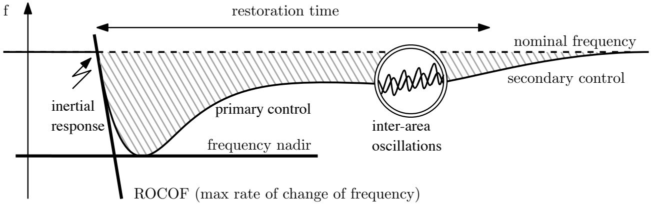
Example #1: Secondary Frequency Control in Bulk Grid



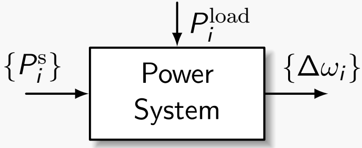
- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$



Example #1: Secondary Frequency Control in Bulk Grid

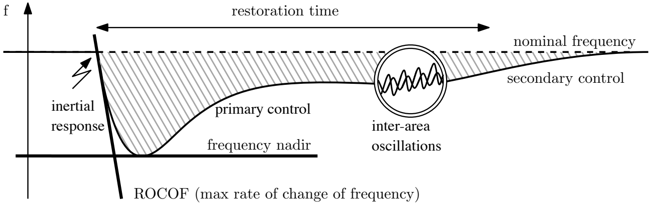


- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$

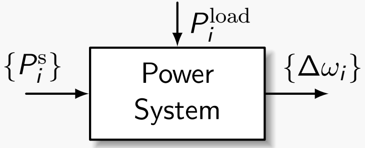


$$
\begin{aligned}
\underset{P_i^{\text{s}} \in \{\text{limits}\}}{\text{minimize}} \quad & \sum\nolimits_{i=1}^{n} C_i(P_i^{\text{s}}) \\
\text{subject to} \quad & \Delta\omega_i = 0 \\
& \text{(System dynamics)}
\end{aligned}
$$

Example #1: Secondary Frequency Control in Bulk Grid



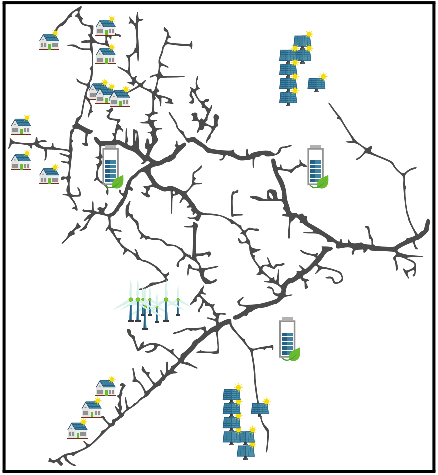
- Centralized **secondary** (integral) control drives $\Delta\omega \to 0$



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

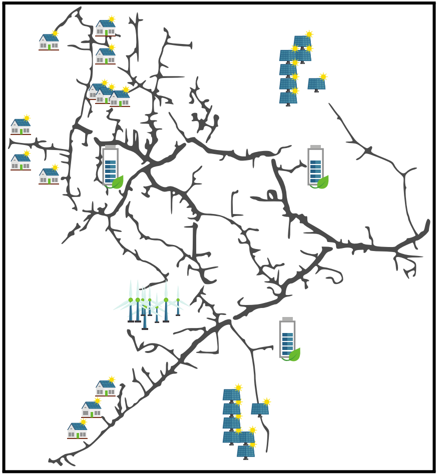
Want: Fast hierarchical resource-allocating control loops

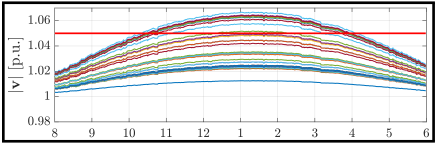
Example #2: Voltage Regulation in PV-Heavy Feeders



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

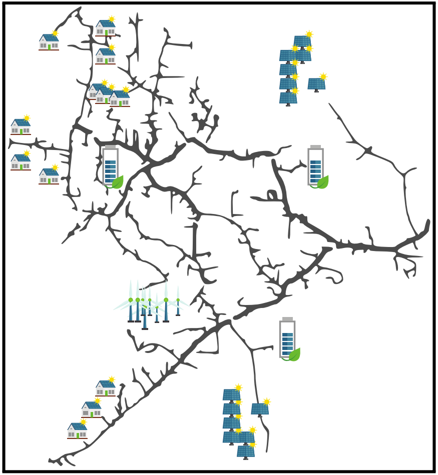
Example #2: Voltage Regulation in PV-Heavy Feeders

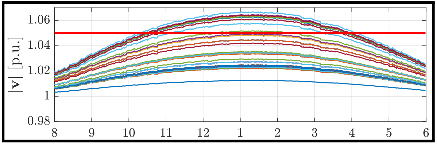




Want: Online control, robust w.r.t. grid model π

Example #2: Voltage Regulation in PV-Heavy Feeders





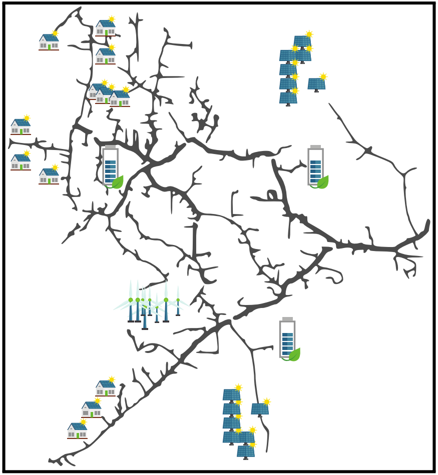
Grid Model: $\vec{v} = \pi(\vec{u}, \vec{w})$

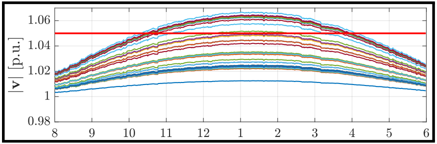
- $\vec{u}$ = controllable power
- $\vec{w}$ = uncontrollable power

$$\begin{aligned} \underset{u \in \mathcal{U}}{\text{minimize}} \quad & \|u - u^{\text{nom}}\|_2^2 \\ \text{subject to} \quad & v \in [v_{\min}, v_{\max}] \\ & v = \pi(u, w) \end{aligned}$$

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

Example #2: Voltage Regulation in PV-Heavy Feeders





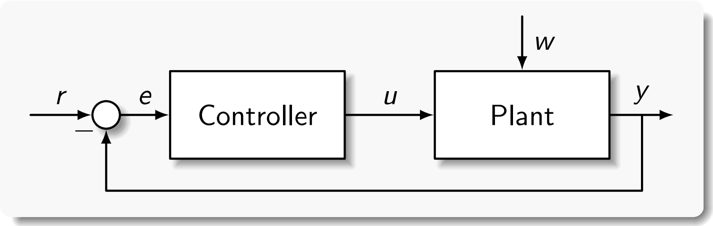
Grid Model: $\vec{v} = \pi(\vec{u}, \vec{w})$

- $\vec{u}$ = controllable power
- $\vec{w}$ = uncontrollable power

$$\begin{aligned} \underset{u \in \mathcal{U}}{\text{minimize}} \quad & \|u - u^{\mathrm{nom}}\|_2^2 \\ \text{subject to} \quad & v \in [v_{\min}, v_{\max}] \\ & v = \pi(u, w) \end{aligned}$$

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

Example #3: A Non-Traditional Servo Problem



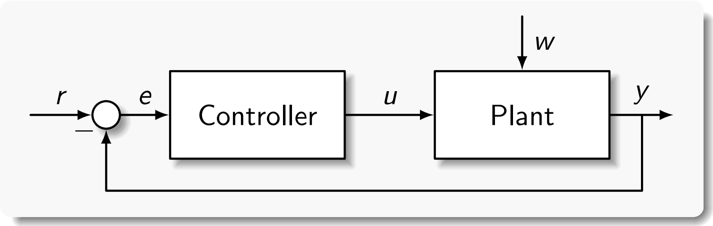
$$\underset{u}{\text{minimize}} \quad \underbrace{\|y - r\|_\infty}_{\text{Worst Error}} + \underbrace{c_1 \sum_{i=1}^{m} \max(0, \underline{u}_i - u_i, u_i - \overline{u}_i)^2}_{\text{Penalty on } u} + \underbrace{c_2 \|u\|_1}_{\text{Sparse action}}$$

subject to (System Dynamics)

- **Note:** feasible reference $\implies$ exact tracking

Want: Constructive solution to this class of problems

Example #3: A Non-Traditional Servo Problem



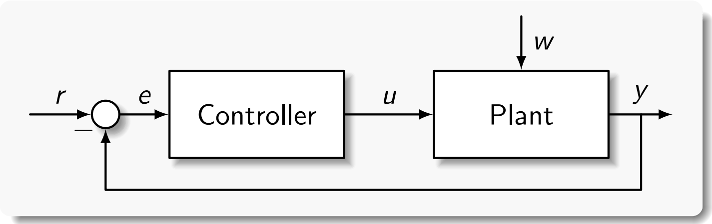
$$\underset{u}{\text{minimize}} \quad \underbrace{\|y - r\|_\infty}_{\text{Worst Error}} + \underbrace{c_1 \sum_{i=1}^{m} \max(0, \underline{u}_i - u_i, u_i - \overline{u}_i)^2}_{\text{Penalty on } u} + \underbrace{c_2 \|u\|_1}_{\text{Sparse action}}$$

subject to (System Dynamics)

- **Note:** feasible reference $\implies$ exact tracking

Want: Constructive solution to this class of problems

Example #3: A Non-Traditional Servo Problem



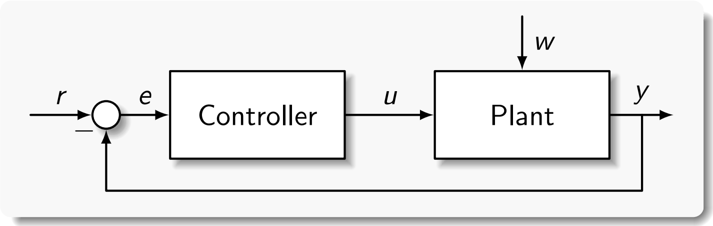
$$\underset{u}{\text{minimize}} \quad \underbrace{\|y - r\|_\infty}_{\text{Worst Error}} + \underbrace{c_1 \sum_{i=1}^{m} \max(0, \underline{u}_i - u_i, u_i - \overline{u}_i)^2}_{\text{Penalty on } u} + \underbrace{c_2 \|u\|_1}_{\text{Sparse action}}$$

subject to (System Dynamics)

- **Note:** feasible reference $\implies$ exact tracking

Want: Constructive solution to this class of problems

Example #3: A Non-Traditional Servo Problem



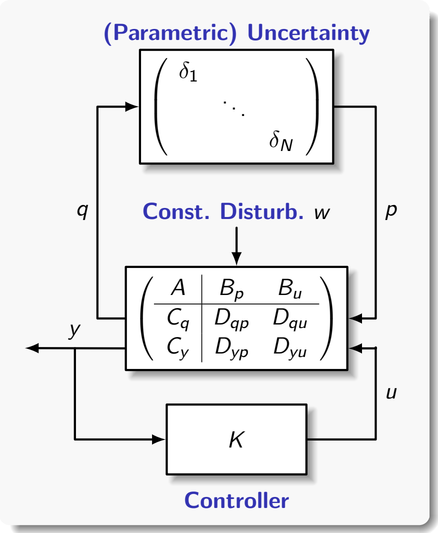
$$\underset{u}{\text{minimize}} \quad \underbrace{\|y - r\|_\infty}_{\text{Worst Error}} + \underbrace{c_1 \sum_{i=1}^{m} \max(0, \underline{u}_i - u_i, u_i - \overline{u}_i)^2}_{\text{Penalty on } u} + \underbrace{c_2 \|u\|_1}_{\text{Sparse action}}$$

subject to (System Dynamics)

- **Note:** feasible reference $\implies$ exact tracking

Want: Constructive solution to this class of problems

Optimal Steady-State Control Problem Statement



LTI w/ **Structured(!) Uncert.**

$$\dot{x} = A(\delta)x + B(\delta)u + B_w(\delta)w$$
$$y = C(\delta)x + D(\delta)u + Q(\delta)w$$

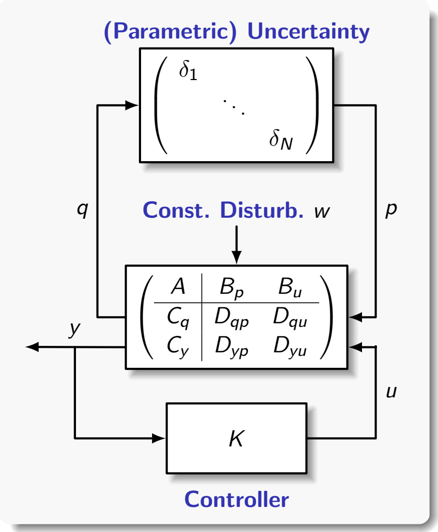
Control specification:

$$y^\star(w,\delta) = \operatorname*{argmin}_{\bar{y}\in\mathcal{C}(w,\delta)} f_0(\bar{y})$$

Wish list:

1. Closed-loop stability
2. $y(t) \to y^\star \quad \underline{\forall w \ \forall \delta \in \boldsymbol{\delta}}$
3. $\|y_T - y^\star\|_{\mathscr{L}_2} \leq \gamma \|w_T\|_{\mathscr{L}_2}$

Optimal Steady-State Control Problem Statement



LTI w/ **Structured(!) Uncert.**

$$\dot{x} = A(\delta)x + B(\delta)u + B_w(\delta)w$$

$$y = C(\delta)x + D(\delta)u + Q(\delta)w$$

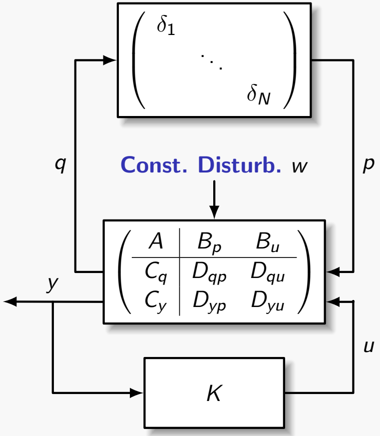
Control specification:

$$y^\star(w,\delta) = \underset{\bar{y}\in\mathcal{C}(w,\delta)}{\operatorname{argmin}}\; f_0(\bar{y})$$

Wish list:

1. Closed-loop stability
2. $y(t) \to y^\star \quad \underline{\forall w\ \forall\delta \in \boldsymbol{\delta}}$
3. $\|y_T - y^\star\|_{\mathscr{L}_2} \leq \gamma \|w_T\|_{\mathscr{L}_2}$

Optimal Steady-State Control Problem Statement



LTI w/ **Structured(!) Uncert.**

$$\dot{x} = A(\delta)x + B(\delta)u + B_w(\delta)w$$
$$y = C(\delta)x + D(\delta)u + Q(\delta)w$$

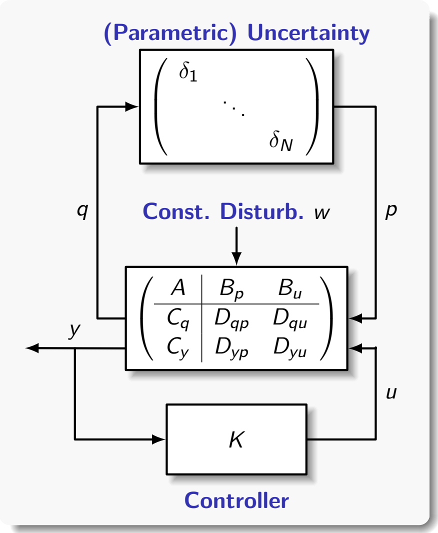
Control specification:

$$y^\star(w,\delta) = \underset{\bar{y}\in\mathcal{C}(w,\delta)}{\operatorname{argmin}} \; f_0(\bar{y})$$

Wish list:

1. Closed-loop stability
2. $y(t) \to y^\star \quad \underline{\forall w \; \forall \delta \in \boldsymbol{\delta}}$
3. $\|y_T - y^\star\|_{\mathscr{L}_2} \le \gamma \|w_T\|_{\mathscr{L}_2}$

Optimal Steady-State Control Problem Statement



LTI w/ **Structured(!) Uncert.**

$$\dot{x} = A(\delta)x + B(\delta)u + B_w(\delta)w$$
$$y = C(\delta)x + D(\delta)u + Q(\delta)w$$

Control specification:

$$y^\star(w,\delta) = \operatorname*{argmin}_{\bar{y}\in\mathcal{C}(w,\delta)} f_0(\bar{y})$$

Wish list:

1. Closed-loop stability
2. $y(t) \to y^\star \quad \underline{\forall w \ \forall \delta \in \boldsymbol{\delta}}$
3. $\|y_T - y^\star\|_{\mathscr{L}_2} \le \gamma \|w_T\|_{\mathscr{L}_2}$

Optimal Steady-State Control Specification

$$\begin{aligned} &\underset{\bar{y}\in\mathbb{R}^p}{\text{minimize}} \quad f_0(\bar{y}) && \text{(steady-state objective)} \\ &\text{subject to} && (\star\star) \end{aligned}$$

$(\star\star)$ ensures plant and optimization are **compatible**

Achievable Equilibria

$$\begin{aligned} 0 &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + D_w w \end{aligned} \quad \Longrightarrow \quad \begin{aligned} G(\delta) &:= [C \;\; D]\,\mathrm{null}([A \;\; B]) \\ G_\perp(\delta) &:= \text{ s.t. } G_\perp(\delta)G(\delta) = 0 \end{aligned}$$

Classic result: Robust tracking $\Longrightarrow$ $\mathrm{range}(G(\delta)) = \mathbb{R}^p \quad \forall\delta.$

This is **not necessary** in our formulation.

Optimal Steady-State Control Specification

$$\begin{array}{lll} \underset{\bar{y} \in \mathbb{R}^p}{\text{minimize}} & f_0(\bar{y}) & \text{(steady-state objective)} \\ \text{subject to} & \bar{y} \in \{\text{achievable equil.}\} & (\star\star) \end{array}$$

$(\star\star)$ ensures plant and optimization are **compatible**

Achievable Equilibria

$$\begin{aligned} 0 &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + D_w w \end{aligned} \implies \begin{aligned} G(\delta) &:= [C \;\; D]\, \mathrm{null}([A \;\; B]) \\ G_\perp(\delta) &:= \;\text{ s.t. } G_\perp(\delta)G(\delta) = 0 \end{aligned}$$

Classic result: Robust tracking $\implies$ $\mathrm{range}(G(\delta)) = \mathbb{R}^p \quad \forall\delta.$

This is **not necessary** in our formulation.

Optimal Steady-State Control Specification

$$\begin{array}{lll} \underset{\bar{y}\in\mathbb{R}^p}{\text{minimize}} & f_0(\bar{y}) & \text{(steady-state objective)} \\ \text{subject to} & \bar{y} \in \{\text{achievable equil.}\} & (\star\star) \end{array}$$

$(\star\star)$ ensures plant and optimization are **compatible**

Achievable Equilibria

$$\begin{aligned} 0 &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + D_w w \end{aligned} \quad \Longrightarrow \quad \begin{aligned} G(\delta) &:= [C \;\; D]\,\mathrm{null}([A \;\; B]) \\ G_\perp(\delta) &:= \text{ s.t. } G_\perp(\delta)G(\delta) = 0 \end{aligned}$$

Classic result: Robust tracking $\Longrightarrow$ $\mathrm{range}(G(\delta)) = \mathbb{R}^p \quad \forall\delta.$

This is **not necessary** in our formulation.

Optimal Steady-State Control Specification

$$\begin{array}{lll} \underset{\bar{y}\in\mathbb{R}^p}{\text{minimize}} & f_0(\bar{y}) & \text{(steady-state objective)} \\ \text{subject to} & \bar{y} \in \{\text{achievable equil.}\} & (\star\star) \end{array}$$

$(\star\star)$ ensures plant and optimization are **compatible**

Achievable Equilibria

$$\begin{aligned} 0 &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + D_w w \end{aligned} \quad \Longrightarrow \quad \begin{aligned} G(\delta) &:= [C \;\; D]\,\mathrm{null}([A \;\; B]) \\ G_\perp(\delta) &:= \text{ s.t. } G_\perp(\delta)G(\delta) = 0 \end{aligned}$$

Classic result: Robust tracking $\Longrightarrow$ $\mathrm{range}(G(\delta)) = \mathbb{R}^p \quad \forall\delta.$

This is **not necessary** in our formulation.

Optimal Steady-State Control Specification

$$\begin{aligned} &\underset{\bar{y}\in\mathbb{R}^p}{\text{minimize}} && f_0(\bar{y}) && \text{(steady-state objective)} \\ &\text{subject to} && \boldsymbol{G(\delta)_\perp \bar{y} = b(d_1, d_2)} && (\star\star) \end{aligned}$$

$(\star\star)$ ensures plant and optimization are **compatible**

Achievable Equilibria

$$\begin{aligned} 0 &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + D_w w \end{aligned} \implies \begin{aligned} G(\delta) &:= [C \;\; D]\,\mathrm{null}([A \;\; B]) \\ G_\perp(\delta) &:= \text{ s.t. } G_\perp(\delta)G(\delta) = 0 \end{aligned}$$

Classic result: Robust tracking $\implies$ $\mathrm{range}(G(\delta)) = \mathbb{R}^p \quad \forall\delta.$

This is **not necessary** in our formulation.

Optimal Steady-State Control Specification

$$\begin{array}{lll} \underset{\bar{y} \in \mathbb{R}^p}{\text{minimize}} & f_0(\bar{y}) & \text{(steady-state objective)} \\ \text{subject to} & \boldsymbol{G(\delta)_\perp \bar{y} = b(d_1, d_2)} & (\star\star) \\ & H\bar{y} = Lw & \text{(engineering equality)} \\ & J\bar{y} \le Mw & \text{(engineering inequality)} \end{array}$$

$(\star\star)$ ensures plant and optimization are **compatible**

Achievable Equilibria

$$\begin{aligned} 0 &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + D_w w \end{aligned} \quad \Longrightarrow \quad \begin{aligned} G(\delta) &:= [C \;\; D]\,\mathrm{null}([A \;\; B]) \\ G_\perp(\delta) &:= \text{ s.t. } G_\perp(\delta)G(\delta) = 0 \end{aligned}$$

Classic result: Robust tracking $\Longrightarrow$ $\mathrm{range}(G(\delta)) = \mathbb{R}^p \quad \forall \delta.$

This is **not necessary** in our formulation.

Optimal Steady-State Control Specification

$$
\begin{array}{lll}
\underset{\bar{y}\in\mathbb{R}^p}{\text{minimize}} & f_0(\bar{y}) & \text{(objective + ineq. constraint penalty)} \\
\text{subject to} & G(\delta)_\perp \bar{y} = b & (\star\star) \\
 & H\bar{y} = Lw & \text{(engineering equality)}
\end{array}
$$

Optimal Steady-State Control Specification

$$
\begin{array}{lll}
\underset{\bar{y}\in\mathbb{R}^p}{\text{minimize}} & f_0(\bar{y}) & \text{(objective + ineq. constraint penalty)} \\
\text{subject to} & G(\delta)_\perp \bar{y} = b & (\star\star) \\
 & H\bar{y} = Lw & \text{(engineering equality)}
\end{array}
$$

Equivalent ways of writing stationarity:

$$
\begin{aligned}
& 0 = \nabla f_0(y^\star) + G(\delta)_\perp^\mathsf{T} \lambda^\star + H^\mathsf{T} \mu^\star \\
\iff \quad & 0 = G(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) + H^\mathsf{T} \mu^\star \right) \\
\iff \quad & 0 = T(\delta)^\mathsf{T} \nabla f_0(y^\star)
\end{aligned}
$$

where

$$
\operatorname{range} T(\delta) = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}
$$

Optimality Models for OSS Control

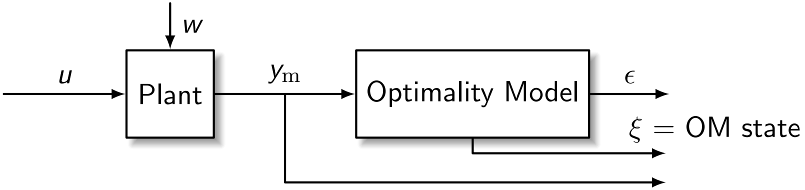
An **optimality model** filters the available measurements to *robustly* produce a **proxy error** ϵ for the unknown tracking error $e = y^\star(w, \delta) - y$

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

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

Optimality Models for OSS Control

An **optimality model** filters the available measurements to *robustly* produce a **proxy error** ϵ for the unknown tracking error $e = y^\star(w,\delta) - y$

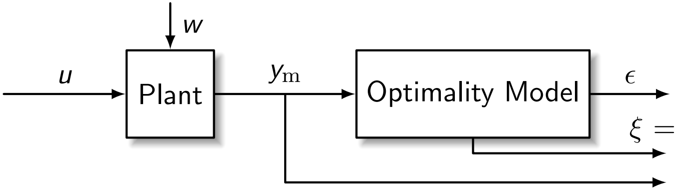


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^\star(w,\delta)$.

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

Optimality Models for OSS Control

An **optimality model** filters the available measurements to *robustly* produce a **proxy error** ϵ for the unknown tracking error $e = y^\star(w, \delta) - y$

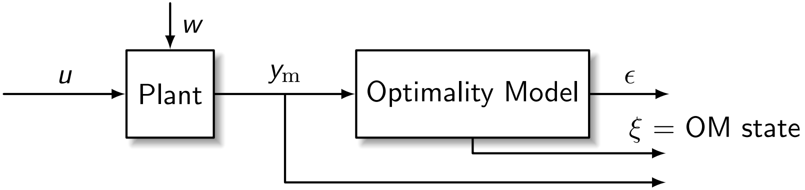


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^\star(w, \delta)$.

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

Optimality Models for OSS Control

An **optimality model** filters the available measurements to *robustly* produce a **proxy error** ϵ for the unknown tracking error $e = y^\star(w, \delta) - y$

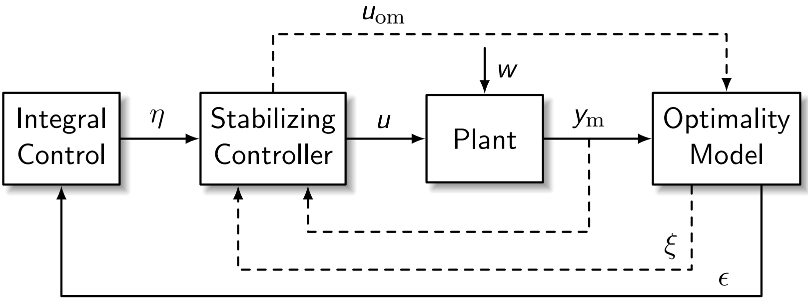


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^\star(w, \delta)$.

"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

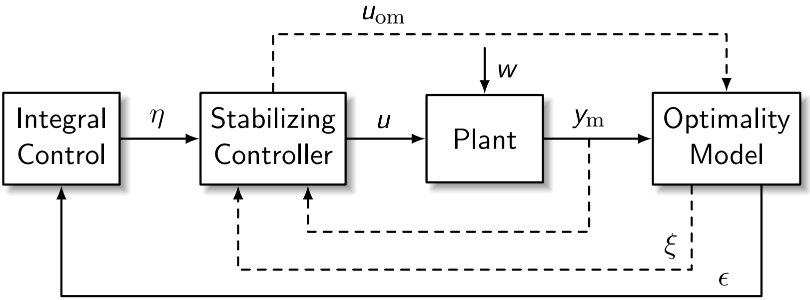
Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

Theorem: **(Stability)** + $(\epsilon \to 0) \quad \Longrightarrow \quad (\boldsymbol{y}(t) \to \boldsymbol{y}^\star)$

From OSS Control to Regulator Problem

Optimality model reduces OSS control to regulator/servomechanism problem



Optimality Model: creates proxy error signal ϵ

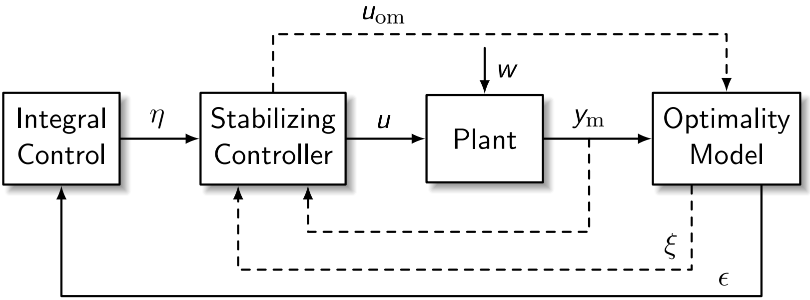
Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

Theorem: (Stability) + ($\epsilon \to 0$) $\implies$ $(y(t) \to y^{\star})$

From OSS Control to Regulator Problem

Optimality model reduces OSS control to regulator/servomechanism problem



Optimality Model: creates proxy error signal ϵ

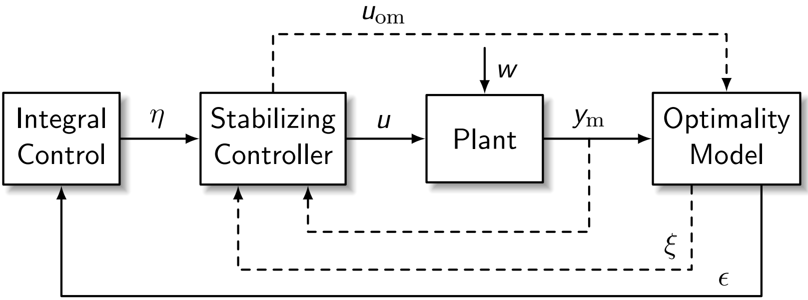
Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

Theorem: (Stability) + ($\epsilon \to 0$) $\implies$ $(\boldsymbol{y}(t) \to \boldsymbol{y}^\star)$

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 closed-loop system

Theorem: (Stability) + ($\epsilon \to 0$) $\implies$ ($\boldsymbol{y}(t) \to \boldsymbol{y}^\star$)

Robustness: When is the uncertainty important?

Recall: With $\operatorname{range} T(\delta) = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$, optimality conditions

$$
\begin{aligned}
& 0 = G(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) + H^\mathsf{T} \mu^\star \right) \\
\iff \quad & 0 = T(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) \right)
\end{aligned}
$$

Idea: Use these to (robustly?) construct proxy error signals

1. Robust Full Rank: $\operatorname{rank} \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = n + p$ $\quad \forall \delta \in \boldsymbol{\delta}$
2. Robust Output Space: $\exists G_0$ s.t. $\operatorname{range} G_0 = \operatorname{range} G(\delta)$ $\quad \forall \delta \in \boldsymbol{\delta}$
3. Robust Feasible Space: $\exists T_0$ s.t. $\operatorname{range} T_0 = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$ $\quad \forall \delta \in \boldsymbol{\delta}$

Easy to show: (1) $\implies$ (2) $\implies$ (3). Conditions **checkable via SDPs**.

Robustness: When is the uncertainty important?

Recall: With $\operatorname{range} T(\delta) = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$, optimality conditions

$$
\begin{aligned}
& 0 = G(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) + H^\mathsf{T} \mu^\star \right) \\
\iff \quad & 0 = T(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) \right)
\end{aligned}
$$

Idea: Use these to (robustly?) construct proxy error signals

1. Robust Full Rank: $\operatorname{rank} \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = n + p$ $\quad \forall \delta \in \boldsymbol{\delta}$
2. Robust Output Space: $\exists G_0$ s.t. $\operatorname{range} G_0 = \operatorname{range} G(\delta)$ $\quad \forall \delta \in \boldsymbol{\delta}$
3. Robust Feasible Space: $\exists T_0$ s.t. $\operatorname{range} T_0 = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$ $\quad \forall \delta \in \boldsymbol{\delta}$

Easy to show: (1) $\implies$ (2) $\implies$ (3). Conditions **checkable via SDPs**.

Robustness: When is the uncertainty important?

Recall: With $\operatorname{range} T(\delta) = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$, optimality conditions

$$\begin{aligned} 0 &= G(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) + H^\mathsf{T} \mu^\star\right) \\ \iff \qquad 0 &= T(\delta)^\mathsf{T} \left(\nabla f_0(y^\star)\right) \end{aligned}$$

Idea: Use these to (robustly?) construct proxy error signals

1. Robust Full Rank: $\operatorname{rank} \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = n + p \qquad \forall \delta \in \boldsymbol{\delta}$
2. Robust Output Space: $\exists G_0$ s.t. $\operatorname{range} G_0 = \operatorname{range} G(\delta) \qquad \forall \delta \in \boldsymbol{\delta}$
3. Robust Feasible Space: $\exists T_0$ s.t. $\operatorname{range} T_0 = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix} \qquad \forall \delta \in \boldsymbol{\delta}$

Easy to show: (1) $\implies$ (2) $\implies$ (3). Conditions **checkable via SDPs**.

Robustness: When is the uncertainty important?

Recall: With $\operatorname{range} T(\delta) = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$, optimality conditions

$$\begin{aligned} 0 &= G(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) + H^\mathsf{T} \mu^\star \right) \\ \iff \qquad 0 &= T(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) \right) \end{aligned}$$

Idea: Use these to (robustly?) construct proxy error signals

1. Robust Full Rank: $\operatorname{rank} \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = n + p \qquad \forall \delta \in \boldsymbol{\delta}$
2. Robust Output Space: $\exists G_0$ s.t. $\operatorname{range} G_0 = \operatorname{range} G(\delta) \qquad \forall \delta \in \boldsymbol{\delta}$
3. Robust Feasible Space: $\exists T_0$ s.t. $\operatorname{range} T_0 = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix} \qquad \forall \delta \in \boldsymbol{\delta}$

Easy to show: (1) $\implies$ (2) $\implies$ (3). Conditions **checkable via SDPs**.

Robustness: When is the uncertainty important?

Recall: With $\operatorname{range} T(\delta) = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$, optimality conditions

$$\begin{aligned} & 0 = G(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) + H^\mathsf{T} \mu^\star \right) \\ \iff \quad & 0 = T(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) \right) \end{aligned}$$

Idea: Use these to (robustly?) construct proxy error signals

1. Robust Full Rank: $\operatorname{rank} \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = n + p \qquad \forall \delta \in \boldsymbol{\delta}$
2. Robust Output Space: $\exists G_0$ s.t. $\operatorname{range} G_0 = \operatorname{range} G(\delta) \qquad \forall \delta \in \boldsymbol{\delta}$
3. Robust Feasible Space: $\exists T_0$ s.t. $\operatorname{range} T_0 = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix} \qquad \forall \delta \in \boldsymbol{\delta}$

Easy to show: (1) $\implies$ (2) $\implies$ (3). Conditions **checkable via SDPs**.

Robustness: When is the uncertainty important?

Recall: With $\operatorname{range} T(\delta) = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix}$, optimality conditions

$$\begin{aligned} & 0 = G(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) + H^\mathsf{T} \mu^\star \right) \\ \iff \quad & 0 = T(\delta)^\mathsf{T} \left(\nabla f_0(y^\star) \right) \end{aligned}$$

Idea: Use these to (robustly?) construct proxy error signals

1. Robust Full Rank: $\operatorname{rank} \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = n + p \qquad \forall \delta \in \boldsymbol{\delta}$
2. Robust Output Space: $\exists G_0$ s.t. $\operatorname{range} G_0 = \operatorname{range} G(\delta) \qquad \forall \delta \in \boldsymbol{\delta}$
3. Robust Feasible Space: $\exists T_0$ s.t. $\operatorname{range} T_0 = \operatorname{null} \begin{bmatrix} G_\perp(\delta) \\ H \end{bmatrix} \qquad \forall \delta \in \boldsymbol{\delta}$

Easy to show: (1) $\implies$ (2) $\implies$ (3). Conditions **checkable via SDPs**.

Construction of Optimality Models

1. Robust Full Rank Optimality Model (akin to classic tracking)

$$\begin{aligned} \dot{\mu} &= Hy - Lw \\ \epsilon &= \nabla f_0(y) + H^{\mathsf{T}}\mu \end{aligned}$$

2. Robust Output Subspace Optimality Model

3. Robust Feasible Subspace Optimality Model

. . . and many more! Ask me if you're curious.

Construction of Optimality Models

1. Robust Full Rank Optimality Model (akin to classic tracking)

$$\dot{\mu} = Hy - Lw$$
$$\epsilon = \nabla f_0(y) + H^\mathsf{T}\mu$$

2. Robust Output Subspace Optimality Model

$$\dot{\mu} = Hy - Lw$$
$$\epsilon = \boldsymbol{G_0}^\mathsf{T}(\nabla f_0(y) + H^\mathsf{T}\mu)$$

3. Robust Feasible Subspace Optimality Model

. . . and many more! Ask me if you're curious.

Construction of Optimality Models

1. Robust Full Rank Optimality Model (akin to classic tracking)

$$\begin{aligned} \dot{\mu} &= Hy - Lw \\ \epsilon &= \nabla f_0(y) + H^\mathsf{T} \mu \end{aligned}$$

2. Robust Output Subspace Optimality Model

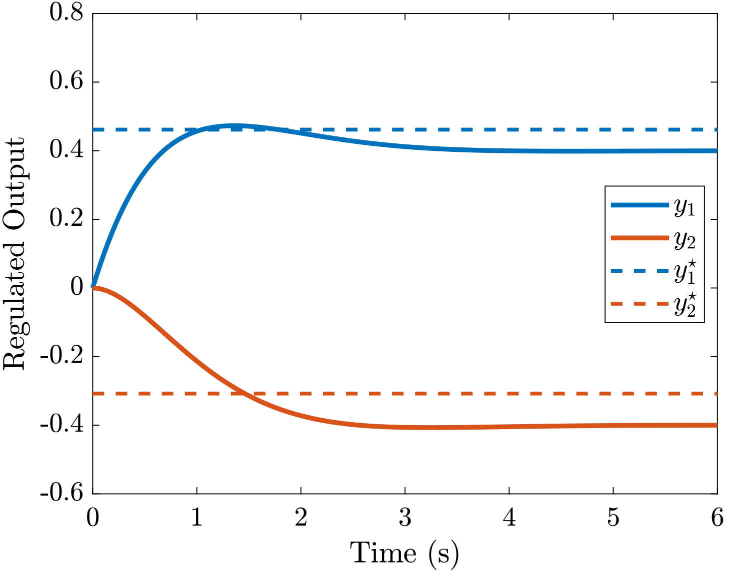
$$\begin{aligned} \dot{\mu} &= Hy - Lw \\ \epsilon &= \boldsymbol{G_0}^\mathsf{T} (\nabla f_0(y) + H^\mathsf{T} \mu) \end{aligned}$$

3. Robust Feasible Subspace Optimality Model

$$\epsilon = \begin{bmatrix} Hy - Lw \\ \boldsymbol{T_0}^\mathsf{T} \nabla f_0(y) \end{bmatrix}$$

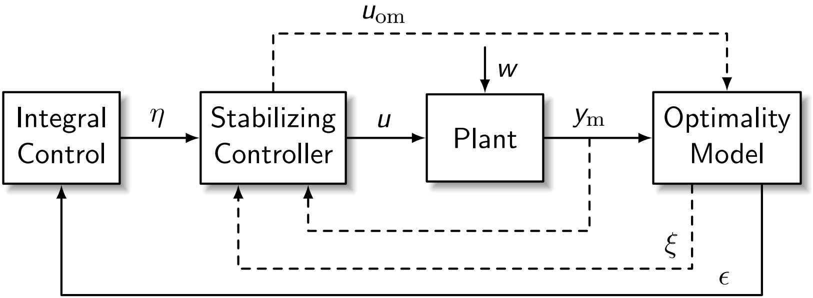
. . . and many more! Ask me if you're curious.

What if robustness conditions fail?



Big Picture for OSS Control

Optimality model reduces OSS control to regulator/servomechanism problem

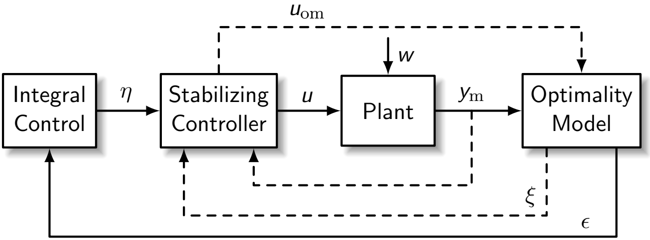


Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

Existence of stabilizing output feedback controller

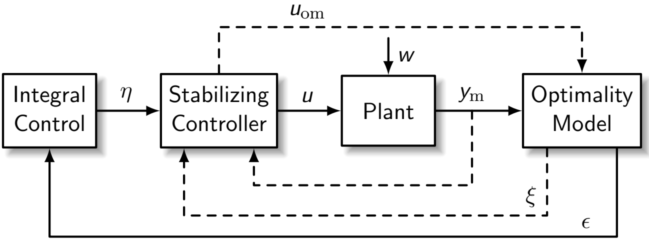


- To use LTI theory, study case $\delta = \mathbb{0}$ with **convex quadratic objective** $f_0(y) = y^{\mathsf{T}}Qy + c^{\mathsf{T}}y$, $Q \succeq \mathbb{0}$

Cascade is stabilizable/detectable **if and only if**

1. plant stabilizable/detectable
2. optimization problem has a unique solution
3. equality constraints are not redundant ($\left[\begin{smallmatrix} G_{\perp} \\ H \end{smallmatrix}\right]$ full row rank)
4. T_0 or G_0 full column rank

Existence of stabilizing output feedback controller

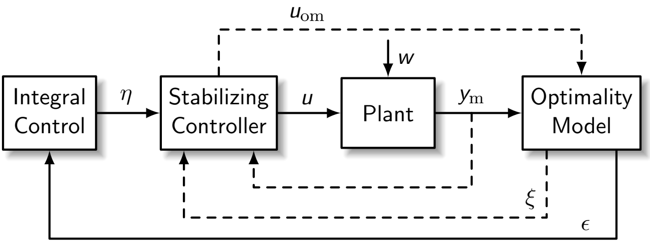


- To use LTI theory, study case $\delta = \mathbb{0}$ with **convex quadratic objective** $f_0(y) = y^\mathsf{T} Q y + c^\mathsf{T} y$, $Q \succeq \mathbb{0}$

Cascade is stabilizable/detectable **if and only if**

1. plant stabilizable/detectable
2. optimization problem has a unique solution
3. equality constraints are not redundant ($\left[\begin{smallmatrix} G_\perp \\ H \end{smallmatrix}\right]$ full row rank)
4. T_0 or G_0 full column rank

Existence of stabilizing output feedback controller

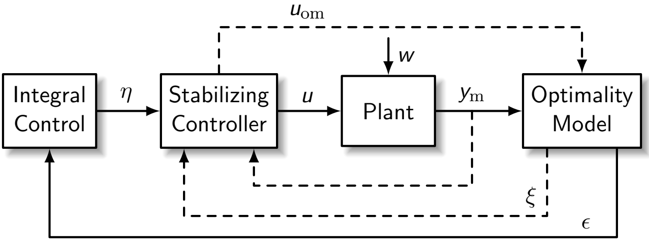


- To use LTI theory, study case $\delta = \mathbb{0}$ with **convex quadratic objective** $f_0(y) = y^{\mathsf{T}} Q y + c^{\mathsf{T}} y$, $Q \succeq \mathbb{0}$

Cascade is stabilizable/detectable **if and only if**

1. plant stabilizable/detectable
2. optimization problem has a unique solution
3. equality constraints are not redundant ($\left[\begin{smallmatrix} G_\perp \\ H \end{smallmatrix}\right]$ full row rank)
4. T_0 or G_0 full column rank

Existence of stabilizing output feedback controller

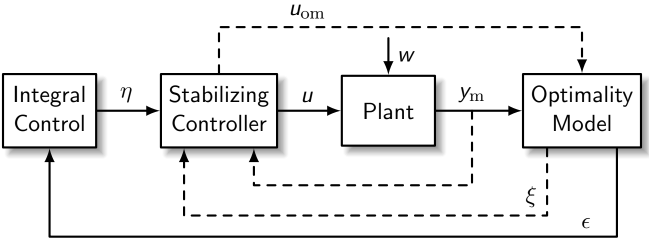


- To use LTI theory, study case $\delta = \mathbb{0}$ with **convex quadratic objective** $f_0(y) = y^\mathsf{T} Q y + c^\mathsf{T} y$, $Q \succeq \mathbb{0}$

Cascade is stabilizable/detectable **if and only if**

1. plant stabilizable/detectable
2. optimization problem has a unique solution
3. equality constraints are not redundant ($\left[\begin{smallmatrix} G_\perp \\ H \end{smallmatrix}\right]$ full row rank)
4. T_0 or G_0 full column rank

Existence of stabilizing output feedback controller



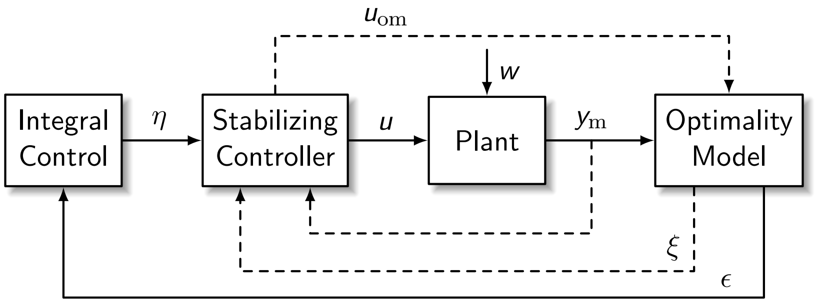
- To use LTI theory, study case $\delta = \mathbb{0}$ with **convex quadratic objective** $f_0(y) = y^{\mathsf{T}}Qy + c^{\mathsf{T}}y$, $Q \succeq \mathbb{0}$

Cascade is stabilizable/detectable **if and only if**

1. plant stabilizable/detectable
2. optimization problem has a unique solution
3. equality constraints are not redundant ($\left[\begin{smallmatrix} G_\perp \\ H \end{smallmatrix}\right]$ full row rank)
4. T_0 or G_0 full column rank

Big Picture for OSS Control

Optimality model reduces OSS control to output regulation

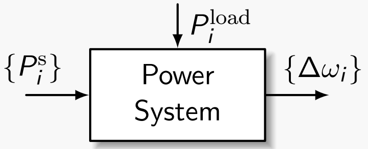


Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

Example #1: Secondary Frequency Control in Bulk Grid



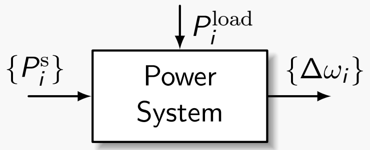
$$\begin{aligned} &\underset{P_i^{\text{s}} \in \{\text{limits}\}}{\text{minimize}} && \sum_{i=1}^{n} C_i(P_i^{\text{s}}) \\ &\text{subject to} && \Delta\omega_i = 0 \\ & && \text{(System dynamics)} \end{aligned}$$

Structured uncertain dynamic model:

$$\begin{aligned} \Delta\dot{\theta}_i &= \Delta\omega_i\,, \\ \boldsymbol{M_i}\Delta\dot{\omega}_i &= -\sum_{j=1}^{n} \boldsymbol{T_{ij}}(\Delta\theta_i - \Delta\theta_j) - \boldsymbol{D_i}\Delta\omega_i + \Delta P_{\mathrm{m},i} + \Delta P_{\mathrm{u},i} \\ \boldsymbol{T_i}\Delta\dot{P}_{\mathrm{m},i} &= -\Delta P_{\mathrm{m},i} - \boldsymbol{R_{\mathrm{d},i}^{-1}}\Delta\omega_i + P_i^{\mathrm{s}}\,. \end{aligned}$$

Can construct many different optimality models for this problem
$\Longrightarrow$ Reveals **many** possible control architectures!

Example #1: Secondary Frequency Control in Bulk Grid



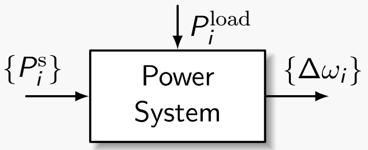
$$
\begin{aligned}
\underset{P_i^{\mathrm{s}} \in \{\text{limits}\}}{\text{minimize}} \quad & \sum_{i=1}^{n} C_i(P_i^{\mathrm{s}}) \\
\text{subject to} \quad & \Delta\omega_i = 0 \\
& \text{(System dynamics)}
\end{aligned}
$$

Structured uncertain dynamic model:

$$
\begin{aligned}
\Delta\dot{\theta}_i &= \Delta\omega_i \,, \\
\boldsymbol{M_i}\Delta\dot{\omega}_i &= -\sum_{j=1}^{n} \boldsymbol{T_{ij}}(\Delta\theta_i - \Delta\theta_j) - \boldsymbol{D_i}\Delta\omega_i + \Delta P_{\mathrm{m},i} + \Delta P_{\mathrm{u},i} \\
\boldsymbol{T_i}\Delta\dot{P}_{\mathrm{m},i} &= -\Delta P_{\mathrm{m},i} - \boldsymbol{R_{\mathrm{d},i}^{-1}}\Delta\omega_i + P_i^{\mathrm{s}} \,.
\end{aligned}
$$

Can construct many different optimality models for this problem
$\Longrightarrow$ Reveals **many** possible control architectures!

Example #1: Secondary Frequency Control in Bulk Grid



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

Structured uncertain dynamic model:

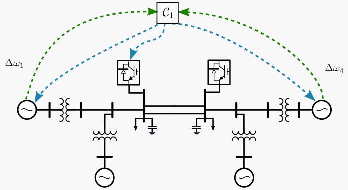
$$\begin{aligned} \Delta\dot{\theta}_i &= \Delta\omega_i \,, \\ \boldsymbol{M_i}\Delta\dot{\omega}_i &= -\sum_{j=1}^{n} \boldsymbol{T_{ij}}(\Delta\theta_i - \Delta\theta_j) - \boldsymbol{D_i}\Delta\omega_i + \Delta P_{\mathrm{m},i} + \Delta P_{\mathrm{u},i} \\ \boldsymbol{T_i}\Delta\dot{P}_{\mathrm{m},i} &= -\Delta P_{\mathrm{m},i} - \boldsymbol{R_{\mathrm{d},i}^{-1}}\Delta\omega_i + P_i^{\mathrm{s}} . \end{aligned}$$

Can construct many different optimality models for this problem
$\Longrightarrow$ Reveals **many** possible control architectures!

Example #1: Secondary Frequency Control in Bulk Grid

1. **Centralized** (generalized AGC) approach:

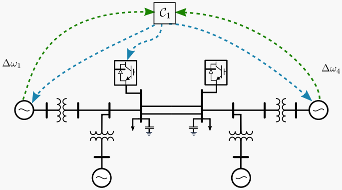
$$\tau \dot{\eta} = -\sum_{i=1}^{n} c_i \Delta\omega_i$$
$$P_i^{\mathrm{s}} = (\nabla C_i)^{-1}(\eta)$$



Example #1: Secondary Frequency Control in Bulk Grid

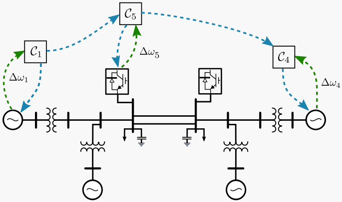
1. **Centralized** (generalized AGC) approach:

$$\tau \dot{\eta} = -\sum\nolimits_{i=1}^{n} c_i \Delta\omega_i$$
$$P_i^{\mathrm{s}} = (\nabla C_i)^{-1}(\eta)$$

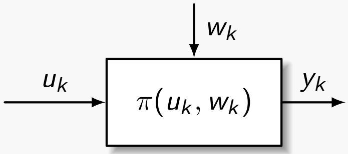


2. **Distributed** consensus-based approach:

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

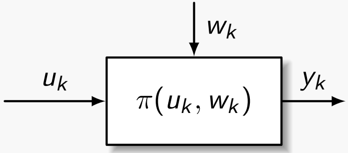


Feedback-Based Optimization of Maps

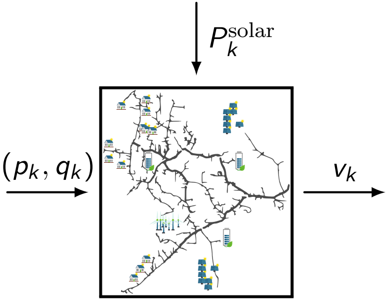


- Map π **unknown**
- Disturbance w **unknown**
- Output y **measurable**

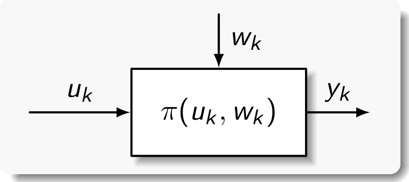
Feedback-Based Optimization of Maps



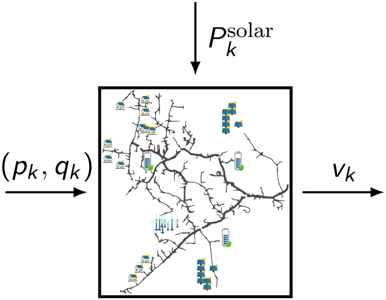
- Map π **unknown**
- Disturbance w **unknown**
- Output y **measurable**



Feedback-Based Optimization of Maps



- Map π **unknown**
- Disturbance w **unknown**
- Output y **measurable**



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

Assumptions:

- $\mathcal{U}$ is convex & compact
- $w \in \mathcal{W}$ compact
- f, g are cvx, $\mathcal{C}^2$, Lipschitz ∇
- π is $\mathcal{C}^1$ in u and $\mathcal{C}^0$ in w

Feedback-Based Optimization of Maps

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

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:

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

Feedback-Based Optimization of Maps

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

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 Maps

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

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 Maps

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

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:

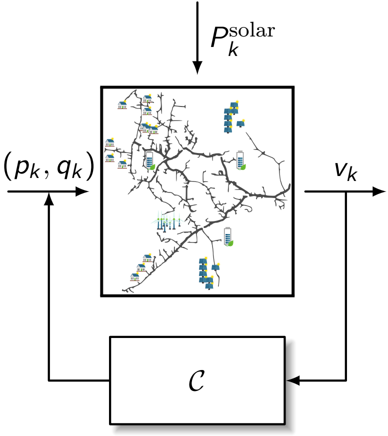
$$u_{k+1}=\mathrm{Proj}_{\mathcal{U}}\left\{u_k-\alpha\left(\nabla f(u_k)+\mathbf{\Pi}^{\mathsf{T}}\nabla g(\pi(u_k,w_k))\right)\right\}$$

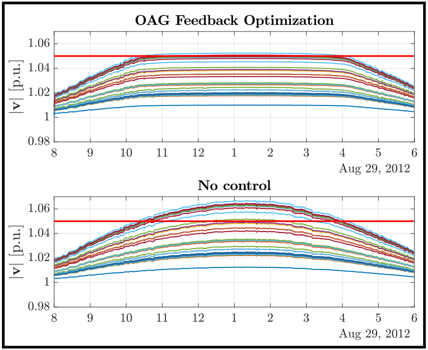
Approximate Online Projected Gradient Descent:

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

Example #2: Voltage Regulation in PV-Heavy Feeders

Short story: Outperforms volt-var control in cost and is provably robust to large model variations





Longer Story: Convergence of Approx. Gradient Descent

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

Theorem from VI Literature: If F_w is ρ-strongly monotone and L-Lipschitz continuous and $\alpha < \rho / L^2$ w.r.t. inner product $\langle x, y \rangle_P = x^{\mathsf{T}} P y'$ with $P \succ \mathbb{0}$, then iteration converges **exponentially** to a **unique** equilibrium.

(Put Lipschitz condition to the side, focus on monotone)

Problem: $F_w(u)$ is **uncertain**.

When is F_w **robustly** ρ-strongly monotone?

Longer Story: Convergence of Approx. Gradient Descent

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

Theorem from VI Literature: If F_w is ρ-strongly monotone and L-Lipschitz continuous and $\alpha < \rho/L^2$ w.r.t. inner product $\langle x, y\rangle_P = x^\mathsf{T} P y'$ with $P \succ \mathbb{0}$, then iteration converges **exponentially** to a **unique** equilibrium.

(Put Lipschitz condition to the side, focus on monotone)

Problem: $F_w(u)$ is **uncertain**.

When is F_w **robustly** ρ-strongly monotone?

Longer Story: Convergence of Approx. Gradient Descent

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

Theorem from VI Literature: If F_w is ρ-strongly monotone and L-Lipschitz continuous and $\alpha < \rho/L^2$ w.r.t. inner product $\langle x, y\rangle_P = x^{\mathsf{T}} P y'$ with $P \succ \mathbb{0}$, then iteration converges **exponentially** to a **unique** equilibrium.

(Put Lipschitz condition to the side, focus on monotone)

Problem: $F_w(u)$ is **uncertain**.

When is F_w **robustly** ρ-strongly monotone?

Longer Story: Monotonicity via the Jacobian

Strong Monotonicity: Given $P \succ 0$, the map F_w is $\rho-$strongly monotone w.r.t $\langle \cdot, \cdot \rangle_P$ if and only if

$$\partial F_w(u)^\mathsf{T} P + P \partial F_w(u) \succ 2\rho P, \qquad \forall\, u \in \mathcal{U}$$

where $\partial F_w(u_k) = \nabla^2 f(u_k) + \Pi^\mathsf{T} \nabla^2 g(\pi(u_k, w_k)) \partial \pi(u_k, w_k)$

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

Robust Strong Monotonicity: If we have a set $\mathcal{J}$ of matrices such that $\partial F_w(\mathcal{U}) \subseteq \mathcal{J}$, then $F_w(u)$ is ρ-strongly monotone if

$$J^\mathsf{T} P + PJ \succ 2\rho P \qquad \forall J \in \mathcal{J}.$$

When is this test tractable?

Longer Story: Monotonicity via the Jacobian

Strong Monotonicity: Given $P \succ 0$, the map F_w is $\rho-$strongly monotone w.r.t $\langle \cdot, \cdot \rangle_P$ if and only if

$$\partial F_w(u)^\mathsf{T} P + P \partial F_w(u) \succ 2\rho P, \qquad \forall\, u \in \mathcal{U}$$

where $\partial F_w(u_k) = \nabla^2 f(u_k) + \Pi^\mathsf{T} \nabla^2 g(\pi(u_k, w_k)) \partial \pi(u_k, w_k)$

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

Robust Strong Monotonicity: If we have a set $\mathcal{J}$ of matrices such that $\partial F_w(\mathcal{U}) \subseteq \mathcal{J}$, then $F_w(u)$ is ρ-strongly monotone if

$$J^\mathsf{T} P + PJ \succ 2\rho P \qquad \forall J \in \mathcal{J}.$$

When is this test tractable?

Longer Story: Monotonicity via the Jacobian

Strong Monotonicity: Given $P \succ 0$, the map F_w is ρ−strongly monotone w.r.t $\langle \cdot, \cdot \rangle_P$ if and only if

$$\partial F_w(u)^\top P + P \partial F_w(u) \succ 2\rho P, \qquad \forall\, u \in \mathcal{U}$$

where $\partial F_w(u_k) = \nabla^2 f(u_k) + \Pi^\top \nabla^2 g(\pi(u_k, w_k)) \partial \pi(u_k, w_k)$

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

Robust Strong Monotonicity: If we have a set $\mathcal{J}$ of matrices such that $\partial F_w(\mathcal{U}) \subseteq \mathcal{J}$, then $F_w(u)$ is ρ-strongly monotone if

$$J^\top P + PJ \succ 2\rho P \qquad \forall J \in \mathcal{J}.$$

When is this test tractable?

Longer Story: LFR Uncertainty Modelling

Linear Fractional Representation of Uncertainty

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

where $\boldsymbol{\Delta} \subset \mathbb{R}^{r\times s}$ is a set of matrices and we have a convex cone $\boldsymbol{\Theta}$ of matrices such that

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

Longer Story: LFR Uncertainty Modelling

Linear Fractional Representation of Uncertainty

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

where $\mathbf{\Delta} \subset \mathbb{R}^{r \times s}$ is a set of matrices and we have a convex cone $\mathbf{\Theta}$ of matrices such that

$$\begin{bmatrix} q \\ p \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} q \\ p \end{bmatrix} \geq 0 \qquad \forall p = \Delta q, \ \ \Theta \in \mathbf{\Theta}.$$

Robust Monotonicity via S-Procedure: The set of maps F_w with $\partial F_w(u) \subseteq \mathcal{J}$ is ρ-strongly monotone if $\exists P \succ 0, \Theta \in \mathbf{\Theta}$ s.t.

$$\begin{bmatrix} A^\mathsf{T} P + PA - 2\rho P & PB \\ B^\mathsf{T} P & 0 \end{bmatrix} - \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \succeq 0.$$

Example #2: Voltage Regulation in PV-Heavy Feeders

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

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

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

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

Example #2: Voltage Regulation in PV-Heavy Feeders

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

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

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

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

Example #2: Voltage Regulation in PV-Heavy Feeders

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

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

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

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

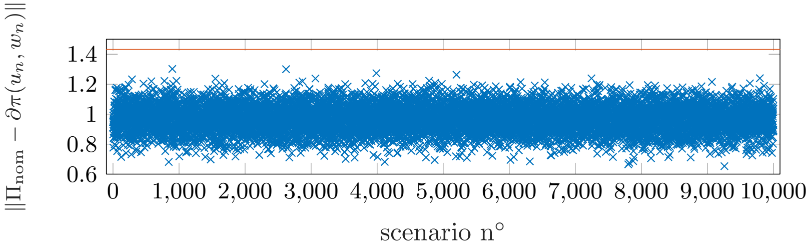
Example #2: Voltage Regulation in PV-Heavy Feeders

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

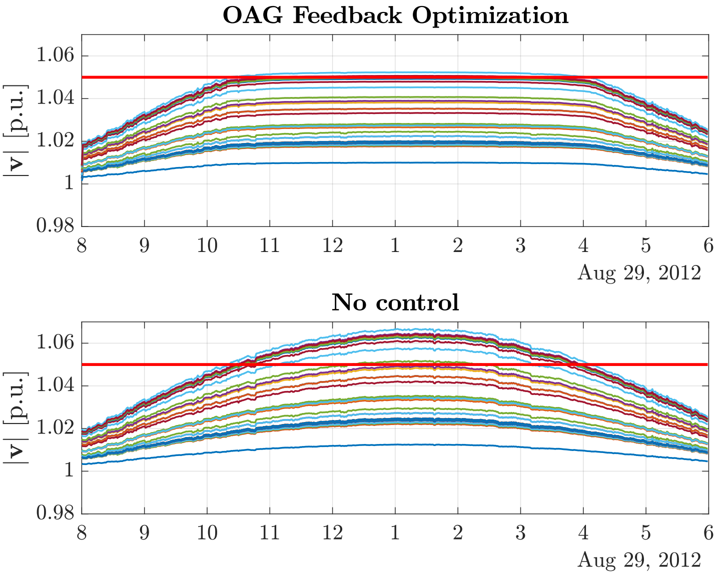
$$\text{subject to} \quad v = \pi(p, q, w) = \mathrm{PowerFlow}(p, q, w)$$

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

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



Results



OAG Feedback Optimization
|v| [p.u.]
1.06
1.04
1.02
1
0.98
8
9
10
11
12
1
2
3
4
5
6
Aug 29, 2012
No control
|v| [p.u.]
1.06
1.04
1.02
1
0.98
8
9
10
11
12
1
2
3
4
5
6
Aug 29, 2012

Recent Work: From Analysis to Design

1. **Synthesize** Π for **distributed** control

$$\begin{aligned} \underset{\Pi,\Theta}{\text{minimize}} \quad & \|\Pi - \hat{\Pi}\| \\ \text{subject to} \quad & \Pi \in \boldsymbol{\Pi} \\ & \begin{bmatrix} A^\mathsf{T} P + PA - 2\rho P & PB(\Pi) \\ B(\Pi)^\mathsf{T} P & 0 \end{bmatrix} - \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\mathsf{T} \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \succeq 0. \end{aligned}$$

2. Output constraints via dualization and **Moreau smoothing**

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

$$\mathscr{L}_\mu(u, \lambda) = f(u) + g(\pi(u, w)) + M_{\mu\mathbb{I}}(\pi(u, w) + \mu\lambda) - \frac{\mu}{2}\|\lambda\|_2^2$$

Primal-Dual Iteration on "Proximal Augmented Lagrangian"

Recent Work: From Analysis to Design

1. **Synthesize** Π for **distributed** control

$$\begin{aligned} \underset{\Pi,\Theta}{\text{minimize}} \quad & \|\Pi - \hat{\Pi}\| \\ \text{subject to} \quad & \Pi \in \boldsymbol{\Pi} \\ & \begin{bmatrix} A^\top P + PA - 2\rho P & PB(\Pi) \\ B(\Pi)^\top P & 0 \end{bmatrix} - \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\top \Theta \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \succeq 0. \end{aligned}$$

2. Output constraints via dualization and **Moreau smoothing**

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

$$\mathscr{L}_\mu(u,\lambda) = f(u) + g(\pi(u,w)) + M_{\mu\mathbb{I}}(\pi(u,w) + \mu\lambda) - \frac{\mu}{2}\|\lambda\|_2^2$$

Primal-Dual Iteration on "Proximal Augmented Lagrangian"

Conclusions

Two frameworks for feedback optimization

1. Optimal steady-state control (leverage regulator/servo theory)
2. Gradient-based feedback (leverage opt. theory + robust ctrl)

Many directions wide open . . .

1. Decentralized, hierarchical, competitive, . . .
2. Performance improvement (e.g., feedforward, anti-windup)
3. Intersection with latest in opt. for ML

Conclusions

Two frameworks for feedback optimization

1. Optimal steady-state control (leverage regulator/servo theory)
2. Gradient-based feedback (leverage opt. theory + robust ctrl)

Many directions wide open . . .

1. Decentralized, hierarchical, competitive, . . .
2. Performance improvement (e.g., feedforward, anti-windup)
3. Intersection with latest in opt. for ML

Conclusions

Two frameworks for feedback optimization

1. Optimal steady-state control (leverage regulator/servo theory)
2. Gradient-based feedback (leverage opt. theory + robust ctrl)

Many directions wide open . . .

1. Decentralized, hierarchical, competitive, . . .
2. Performance improvement (e.g., feedforward, anti-windup)
3. Intersection with latest in opt. for ML




Questions

https://ece.uwaterloo.ca/~jwsimpso/

jwsimpson@uwaterloo.ca

appendix