Optimal Steady-State Control

(with Application to Secondary Frequency Control of Power Systems)

John W. Simpson-Porco

UNIVERSITY OF WATERLOO

NREL Workshop on Opt./ Control

Golden, CO, USA

April 22, 2019

Collaborators

Liam S. P. Lawrence
MASc 2019, Waterloo

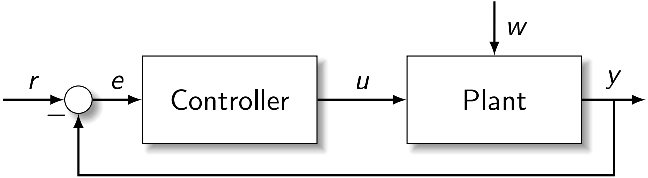
Enrique Mallada
John's Hopkins Univ.

Talk based on:

1. *Lawrence, JWSP, Mallada*: The optimal steady-state control problem (Arxiv preprint, revision pending . . .)

Control Systems 101

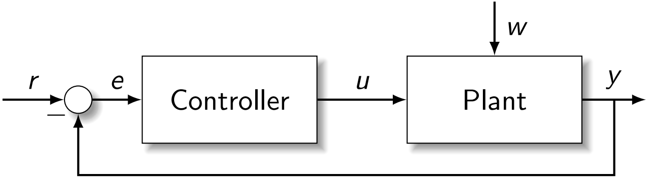
- Prototypical feedback control problem is **tracking** and **disturbance rejection** in the presence of **model uncertainty**



How is the reference r being determined?

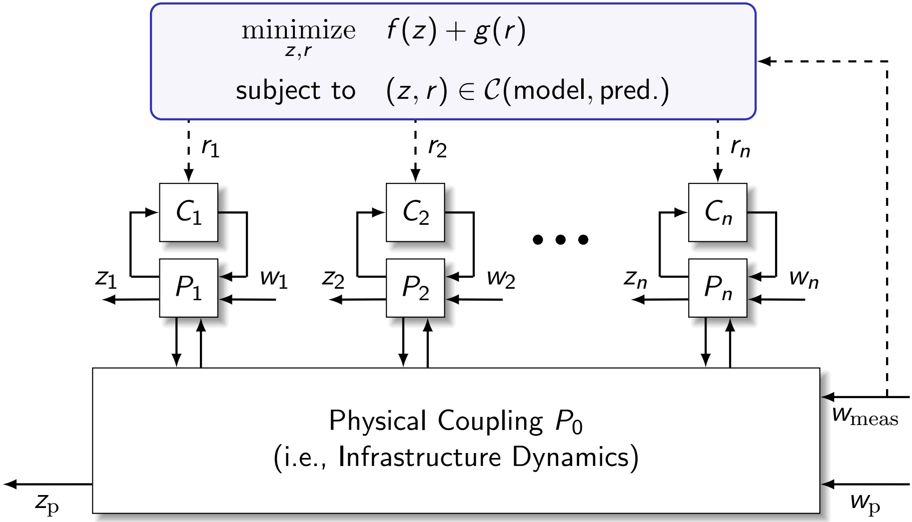
Control Systems 101

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

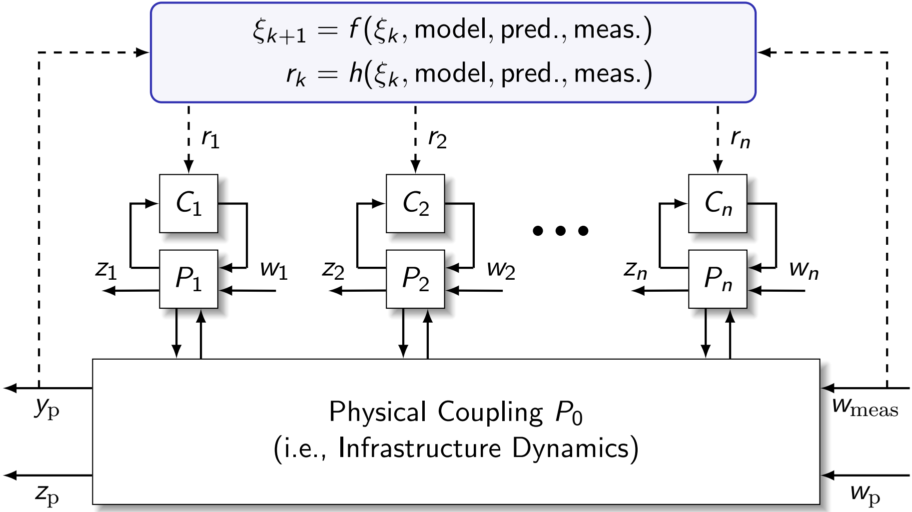


How is the reference r being determined?

Feedforward Optimization of Large-Scale Systems



Feedback Optimization of Large-Scale Systems



Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - a model uncertainty specification (e.g., parametric)
2. a vector of outputs $y \in \mathbb{R}^p$ of system to be optimized
3. an optimization problem in y

Design, *if possible*, a feedback controller such that

1. closed-loop is (robustly) well-posed and internally stable
2. the regulated output tracks its optimal value

$$\lim_{t\to\infty} y(t) - y^\star(t) = 0\,, \qquad \forall \textbf{disturb}, \ \forall \ \textbf{uncertainties}$$

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - a model uncertainty specification (e.g., parametric)
2. a vector of outputs $y \in \mathbb{R}^p$ of system to be optimized
3. an optimization problem in y

Design, *if possible*, a feedback controller such that

1. closed-loop is (robustly) well-posed and internally stable
2. the regulated output tracks its optimal value

$$\lim_{t \to \infty} y(t) - y^{\star}(t) = 0 \,, \qquad \forall \textbf{disturb}, \ \forall \ \textbf{uncertainties}$$

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - a model uncertainty specification (e.g., parametric)
2. a vector of outputs $y \in \mathbb{R}^p$ of system to be optimized
3. an optimization problem in y

Design, *if possible*, a feedback controller such that

1. closed-loop is (robustly) well-posed and internally stable
2. the regulated output tracks its optimal value

$$\lim_{t\to\infty} y(t) - y^\star(t) = 0\,, \qquad \forall \textbf{disturb}, \ \forall \ \textbf{uncertainties}$$

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - a model uncertainty specification (e.g., parametric)
2. a vector of outputs $y \in \mathbb{R}^p$ of system to be optimized
3. an optimization problem in y

Design, *if possible*, a feedback controller such that

1. closed-loop is (robustly) well-posed and internally stable
2. the regulated output tracks its optimal value

$$\lim_{t \to \infty} y(t) - y^\star(t) = 0\,, \qquad \forall \text{disturb}, \ \forall \text{ uncertainties}$$

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - a model uncertainty specification (e.g., parametric)
2. a vector of outputs $y \in \mathbb{R}^p$ of system to be optimized
3. an optimization problem in y

Design, *if possible*, a feedback controller such that

1. closed-loop is (robustly) well-posed and internally stable
2. the regulated output tracks its optimal value

$$\lim_{t \to \infty} y(t) - y^\star(t) = 0\,, \qquad \forall \textbf{disturb}, \ \forall \ \textbf{uncertainties}$$

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - a model uncertainty specification (e.g., parametric)
2. a vector of outputs $y \in \mathbb{R}^p$ of system to be optimized
3. an optimization problem in y

Design, *if possible*, a feedback controller such that

1. closed-loop is (robustly) well-posed and internally stable
2. the regulated output tracks its optimal value

$$\lim_{t \to \infty} y(t) - y^\star(t) = 0\,, \qquad \forall \text{disturb}, \; \forall \text{ uncertainties}$$

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - a model uncertainty specification (e.g., parametric)
2. a vector of outputs $y \in \mathbb{R}^p$ of system to be optimized
3. an optimization problem in y

Design, *if possible*, a feedback controller such that

1. closed-loop is (robustly) well-posed and internally stable
2. the regulated output tracks its optimal value

$$\lim_{t \to \infty} y(t) - y^\star(t) = 0\,, \qquad \forall \textbf{disturb},\ \forall\ \textbf{uncertainties}$$

LTI-Convex OSS Control: Setup Overview

1. **Uncertain (possibly unstable) LTI dynamics**

$$
\begin{aligned}
\dot{x} &= A(\delta)x + B(\delta)u + B_w(\delta)w \\
y_{\mathrm{m}} &= C_{\mathrm{m}}(\delta)x + D_{\mathrm{m}}(\delta) + Q_{\mathrm{m}}(\delta)w \\
y &= C(\delta)x + D(\delta)u + Q(\delta)w
\end{aligned}
$$

- δ = parametric **uncertainty**, w = const. **disturbances**
- y_{m} = system measurements available for **feedback**
- y = system states/inputs to be **optimized**

2. a steady-state **convex optimization problem**

$$
y^{\star}(w,\delta) = \underset{y \in \mathbb{R}^p}{\mathrm{argmin}} \{ f(y,w) \ : \ y \in \mathcal{C}(w,\delta) \}
$$

LTI-Convex OSS Control: Setup Overview

1. **Uncertain (possibly unstable) LTI dynamics**

$$\begin{aligned} \dot{x} &= A(\delta)x + B(\delta)u + B_w(\delta)w \\ y_{\rm m} &= C_{\rm m}(\delta)x + D_{\rm m}(\delta) + Q_{\rm m}(\delta)w \\ y &= C(\delta)x + D(\delta)u + Q(\delta)w \end{aligned}$$

- δ = parametric **uncertainty**, w = const. **disturbances**
- $y_{\rm m}$ = system measurements available for **feedback**
- y = system states/inputs to be **optimized**

2. a steady-state **convex optimization problem**

$$y^\star(w,\delta) = \underset{y \in \mathbb{R}^p}{\operatorname{argmin}} \{ f(y,w) \;:\; y \in \mathcal{C}(w,\delta) \}$$

LTI-Convex OSS Control: Setup I

Forced equilibria $(\bar{x}, \bar{u}, \bar{y})$ satisfy

$$\begin{aligned} \mathbb{0} &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w(\delta)w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + Q(\delta)w \end{aligned}$$

This defines an **affine set** of *achievable* steady-state outputs

$$\overline{Y}(w, \delta) = (\text{offset vector}) + V(\delta)$$

Note: Due to

1. selection of variables $y \in \mathbb{R}^p$ to be optimized, and/or
2. structure of state-space matrices (A, B, C, D)

it may be that $\overline{Y}(w, \delta) \subset \mathbb{R}^p$

constraint $\bar{y} \in \overline{Y}(w, \delta)$ cannot be ignored!!

LTI-Convex OSS Control: Setup I

Forced equilibria $(\bar{x}, \bar{u}, \bar{y})$ satisfy
$$\begin{aligned} \mathbb{0} &= A(\delta)\bar{x} + B(\delta)\bar{u} + B_w(\delta)w \\ \bar{y} &= C(\delta)\bar{x} + D(\delta)\bar{u} + Q(\delta)w \end{aligned}$$

This defines an **affine set** of *achievable* steady-state outputs

$$\overline{Y}(w, \delta) = (\text{offset vector}) + V(\delta)$$

Note: Due to

1. selection of variables $y \in \mathbb{R}^p$ to be optimized, and/or
2. structure of state-space matrices (A, B, C, D)

it may be that $\overline{Y}(w, \delta) \subset \mathbb{R}^p$

constraint $\bar{y} \in \overline{Y}(w, \delta)$ cannot be ignored!!

LTI-Convex OSS Control: Setup II

Desired regulated output $y^\star(w, \delta)$ solution to

$$\begin{aligned} \underset{y \in \mathbb{R}^p}{\text{minimize}} \quad & f_0(y, w) && \text{(convex cost)} \\ \text{subject to} \quad & y \in \overline{Y}(w, \delta) && \text{(equilibrium)} \\ & Hy = Lw && \text{(engineering equality)} \\ & Jy \leq Mw && \text{(engineering inequality)} \end{aligned}$$

Equilibrium constraints ensure **compatibility** between the plant and the optimization problem

$\Longrightarrow$ guarantees a steady-state exists s.t. $y = y^\star(w, \delta)$.

We want to **track** optimal output $y^\star(w, \delta)$

LTI-Convex OSS Control: Setup II

Desired regulated output $y^\star(w, \delta)$ solution to

$$\begin{aligned} \underset{y \in \mathbb{R}^p}{\text{minimize}} \quad & f_0(y, w) && \text{(convex cost)} \\ \text{subject to} \quad & y \in \overline{Y}(w, \delta) && \text{(equilibrium)} \\ & Hy = Lw && \text{(engineering equality)} \\ & Jy \leq Mw && \text{(engineering inequality)} \end{aligned}$$

Equilibrium constraints ensure **compatibility** between the plant and the optimization problem

$\implies$ guarantees a steady-state exists s.t. $y = y^\star(w, \delta)$.

We want to **track** optimal output $y^\star(w, \delta)$

LTI-Convex OSS Control: Setup II

Desired regulated output $y^\star(w, \delta)$ solution to

$$
\begin{aligned}
\underset{y \in \mathbb{R}^p}{\text{minimize}} \quad & f_0(y, w) && \text{(convex cost)} \\
\text{subject to} \quad & y \in \overline{Y}(w, \delta) && \text{(equilibrium)} \\
& Hy = Lw && \text{(engineering equality)} \\
& Jy \leq Mw && \text{(engineering inequality)}
\end{aligned}
$$

Equilibrium constraints ensure **compatibility** between the plant and the optimization problem

$\implies$ guarantees a steady-state exists s.t. $y = y^\star(w, \delta)$.

We want to **track** optimal output $y^\star(w, \delta)$

LTI-Convex OSS Control: Setup II

Desired regulated output $y^\star(w, \delta)$ solution to

$$\begin{aligned} \underset{y \in \mathbb{R}^p}{\text{minimize}} \quad & f_0(y, w) && \text{(convex cost)} \\ \text{subject to} \quad & y \in \overline{Y}(w, \delta) && \text{(equilibrium)} \\ & Hy = Lw && \text{(engineering equality)} \\ & Jy \leq Mw && \text{(engineering inequality)} \end{aligned}$$

Equilibrium constraints ensure **compatibility** between the plant and the optimization problem

$\Longrightarrow$ guarantees a steady-state exists s.t. $y = y^\star(w, \delta)$.

We want to **track** optimal output $y^\star(w, \delta)$

LTI-Convex OSS Control: Setup II

Desired regulated output $y^\star(w,\delta)$ solution to

$$
\begin{aligned}
\underset{y\in\mathbb{R}^p}{\text{minimize}} \quad & f_0(y,w) && \text{(convex cost)}\\
\text{subject to} \quad & y \in \overline{Y}(w,\delta) && \text{(equilibrium)}\\
& Hy = Lw && \text{(engineering equality)}\\
& Jy \le Mw && \text{(engineering inequality)}
\end{aligned}
$$

Equilibrium constraints ensure **compatibility** between the plant and the optimization problem

$\Longrightarrow$ guarantees a steady-state exists s.t. $y = y^\star(w,\delta)$.

We want to **track** optimal output $y^\star(w,\delta)$

Optimality Models for OSS Control

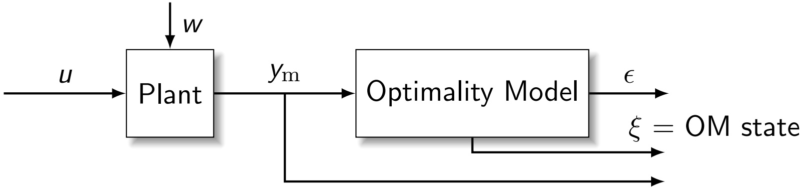
An **optimality model** filters the available measurements to robustly produce a proxy error ϵ for the true tracking error $e = y^\star(w, \delta) - y$

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

Driving ϵ to zero (+ internal stability) drives y to $y^\star$

Optimality Models for OSS Control

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

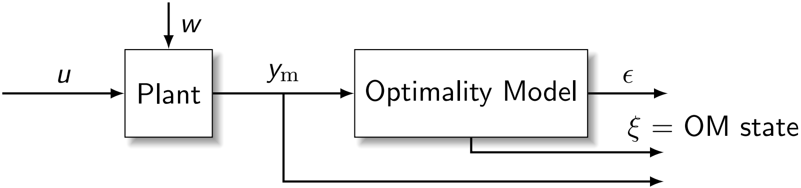


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^\star(w, \delta)$.

Driving ϵ to zero (+ internal stability) drives y to y*

Optimality Models for OSS Control

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

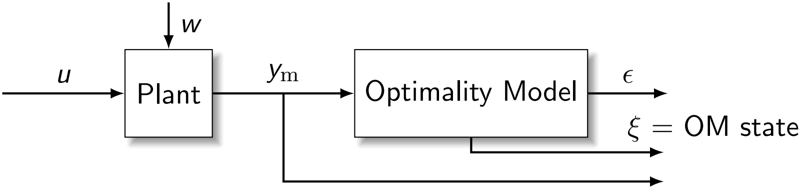


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^\star(w, \delta)$.

Driving ϵ to zero (+ internal stability) drives y to $y^\star$

Optimality Models for OSS Control

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

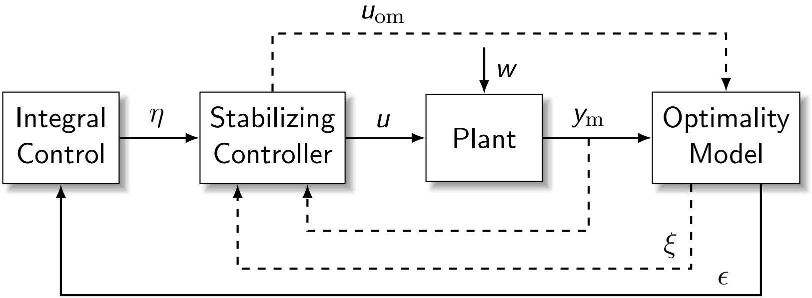


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^{\star}(w, \delta)$.

Driving ϵ to zero (+ internal stability) drives y to $y^{\star}$

Big Picture for OSS Control

Optimality model reduces OSS control to output regulation



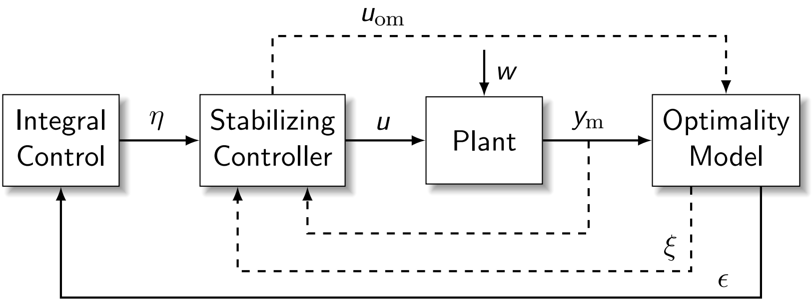
Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

Big Picture for OSS Control

Optimality model reduces OSS control to output regulation



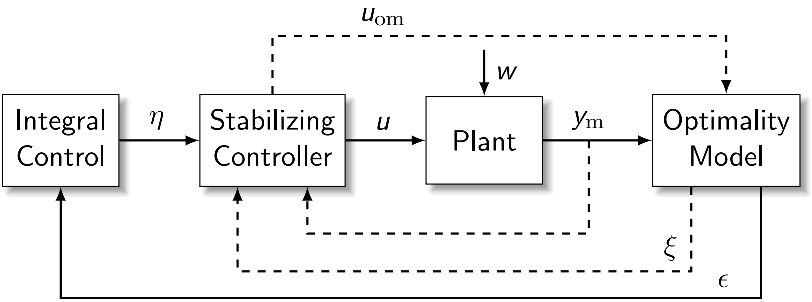
Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

Big Picture for OSS Control

Optimality model reduces OSS control to output regulation



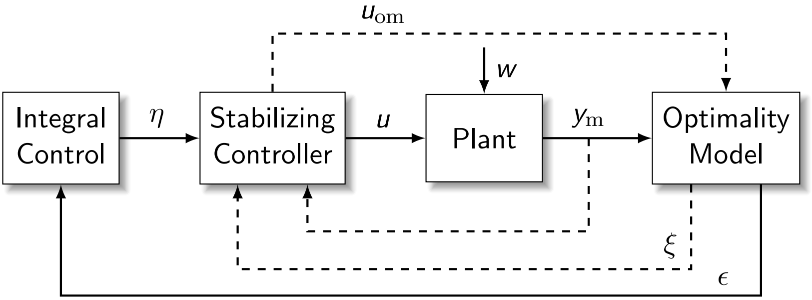
Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

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

Optimality Model Details I

Can we implement an optimality model that is *robust* against δ?

$$\begin{aligned} \underset{y \in \mathbb{R}^p}{\text{minimize}} \quad & f_0(y, w) \\ \text{subject to} \quad & y \in \overline{Y}(w, \delta) = (\text{offset}) + V(\delta) \\ & Hy = Lw \\ & Jy \leq Mw \end{aligned}$$

Optimality condition:

$$\nabla f_0(y^\star, w) + J^\mathsf{T} \nu^\star \perp (V(\delta) \cap \mathrm{null}(H))$$

possibly depends on uncertain parameter δ.

Optimality Model Details I

Can we implement an optimality model that is *robust* against δ?

$$\begin{aligned} \underset{y\in\mathbb{R}^p}{\text{minimize}} \quad & f_0(y,w) \\ \text{subject to} \quad & y \in \overline{Y}(w,\delta) = (\text{offset}) + V(\delta) \\ & Hy = Lw \\ & Jy \le Mw \end{aligned}$$

Optimality condition:

$$\nabla f_0(y^\star, w) + J^\mathsf{T}\nu^\star \perp (V(\delta) \cap \mathrm{null}(H))$$

possibly depends on uncertain parameter δ.

Optimality Model Details I

Can we implement an optimality model that is *robust* against δ?

$$\begin{aligned} \underset{y \in \mathbb{R}^p}{\text{minimize}} \quad & f_0(y, w) \\ \text{subject to} \quad & y \in \overline{Y}(w, \delta) = (\text{offset}) + V(\delta) \\ & Hy = Lw \\ & Jy \leq Mw \end{aligned}$$

Optimality condition:

$$\nabla f_0(y^\star, w) + J^\mathsf{T} \nu^\star \perp (V(\delta) \cap \mathrm{null}(H))$$

possibly depends on uncertain parameter δ.

Optimality Model Details II

When can an optimality model encode the gradient KKT condition?

$$\nabla f_0(y^\star, w) + J^\mathsf{T}\nu^\star \perp (V(\delta) \cap \mathrm{null}(H))$$

Robust Feasible Subspace Property

$V(\delta) \cap \mathrm{null}(H)$ is independent of δ

Optimality Model Details II

When can an optimality model encode the gradient KKT condition?

$$\nabla f_0(y^\star, w) + J^\mathsf{T} \nu^\star \perp (V(\delta) \cap \mathrm{null}(H))$$

Robust Feasible Subspace Property

$V(\delta) \cap \mathrm{null}(H)$ is independent of δ

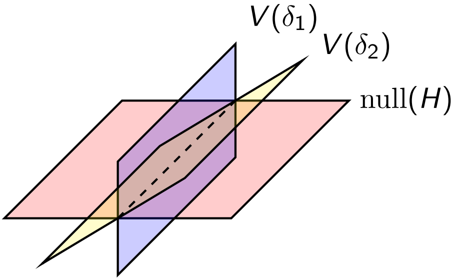
Optimality Model Details II

When can an optimality model encode the gradient KKT condition?

$$\nabla f_0(y^\star, w) + J^\mathsf{T} \nu^\star \perp (V(\delta) \cap \mathrm{null}(H))$$

Robust Feasible Subspace Property

$V(\delta) \cap \mathrm{null}(H)$ is independent of δ



Optimality Model Details III

If Robust Feasible Subspace property holds, then

$$\dot{\nu} = \varphi(\nu, Jy - Mw)$$
$$\epsilon = \begin{bmatrix} Hy - Lw \\ \boldsymbol{T_0}^\mathsf{T}(\nabla f_0(y, w) + J^\mathsf{T}\nu) \end{bmatrix}$$

$$\mathrm{range}(T_0) = \mathsf{V}(\delta) \cap \mathrm{null}(H)$$

(Design freedom!)

is an optimality model for the LTI-Convex OSS Control Problem.

Comments:

1. $T_0^\mathsf{T} z$ extracts component of z in subspace $\mathsf{V}(\delta) \cap \mathrm{null}(H)$:

$$\epsilon_2 = \mathbb{0} \quad \Longleftrightarrow \quad \nabla f_0(y, w) + J^\mathsf{T}\nu \perp V(\delta) \cap \mathrm{null}(H)$$

2. **Flexibility:** different equivalent formulations of optimization problem yield different optimality models

Optimality Model Details III

If Robust Feasible Subspace property holds, then

$$\dot{\nu} = \varphi(\nu, Jy - Mw)$$
$$\epsilon = \begin{bmatrix} Hy - Lw \\ \boldsymbol{T_0}^{\mathsf{T}}(\nabla f_0(y,w) + J^{\mathsf{T}}\nu) \end{bmatrix}$$

$$\mathrm{range}(T_0) = \mathsf{V}(\delta) \cap \mathrm{null}(H)$$

(Design freedom!)

is an optimality model for the LTI-Convex OSS Control Problem.

Comments:

1. $T_0^{\mathsf{T}} z$ extracts component of z in subspace $\mathsf{V}(\delta) \cap \mathrm{null}(H)$:

$$\epsilon_2 = \mathbb{0} \qquad \Longleftrightarrow \qquad \nabla f_0(y,w) + J^{\mathsf{T}}\nu \perp V(\delta) \cap \mathrm{null}(H)$$

2. **Flexibility:** different equivalent formulations of optimization problem yield different optimality models

Optimality Model Details III

If Robust Feasible Subspace property holds, then

$$\dot{\nu} = \varphi(\nu, Jy - Mw)$$
$$\epsilon = \begin{bmatrix} Hy - Lw \\ \mathbf{T_0}^{\mathsf{T}}(\nabla f_0(y, w) + J^{\mathsf{T}}\nu) \end{bmatrix}$$

$$\mathrm{range}(T_0) = \mathsf{V}(\delta) \cap \mathrm{null}(H)$$

(Design freedom!)

is an optimality model for the LTI-Convex OSS Control Problem.

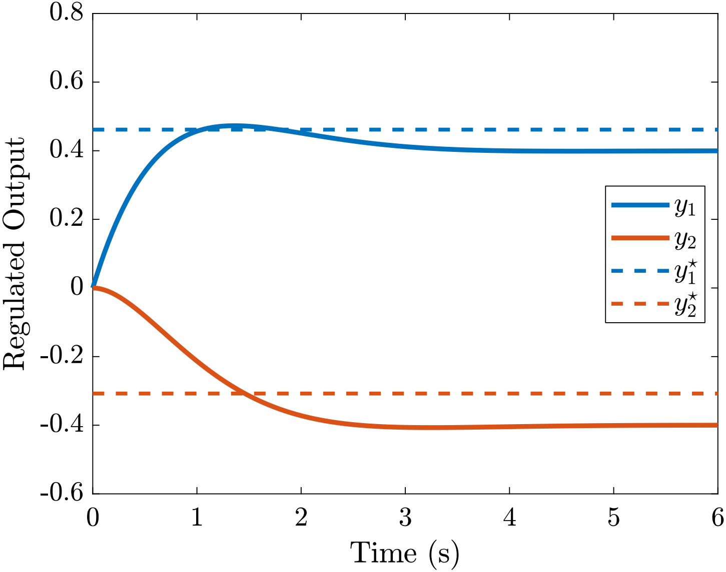
Comments:

1. $T_0^{\mathsf{T}} z$ extracts component of z in subspace $\mathsf{V}(\delta) \cap \mathrm{null}(H)$:

$$\epsilon_2 = \mathbb{0} \quad \Longleftrightarrow \quad \nabla f_0(y, w) + J^{\mathsf{T}}\nu \perp V(\delta) \cap \mathrm{null}(H)$$

2. **Flexibility:** different equivalent formulations of optimization problem yield different optimality models

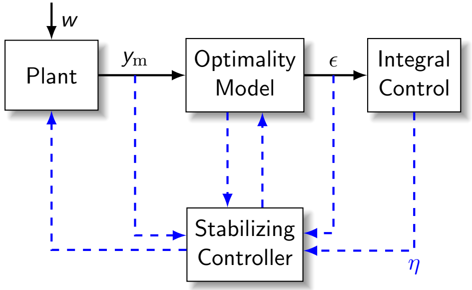
What happens if RFS does not hold?



Stabilizer Details I

Can we actually stabilize this thing?

1. **Goal**: stabilize *cascade* of plant → optimality model → integrators

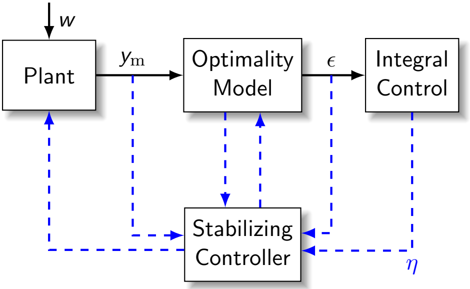


2. Can prove closed-loop stable $\implies$ OSS control problem solved
3. For QP OSS control, can prove cascade is stabilizable iff
    - plant stabilizable/detectable
    - optimization problem has a unique solution
    - engineering constraints not *redundant* with equilibrium constraints
    - T_0 has full column rank

Stabilizer Details I

Can we actually stabilize this thing?

1. **Goal**: stabilize *cascade* of plant $\rightarrow$ optimality model $\rightarrow$ integrators

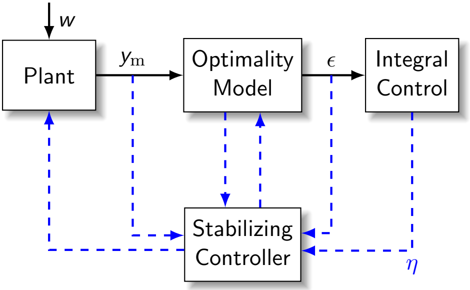


2. Can prove closed-loop stable $\implies$ OSS control problem solved
3. For QP OSS control, can prove cascade is stabilizable iff
    - plant stabilizable/detectable
    - optimization problem has a unique solution
    - engineering constraints not *redundant* with equilibrium constraints
    - T_0 has full column rank

Stabilizer Details I

Can we actually stabilize this thing?

1. **Goal**: stabilize *cascade* of plant → optimality model → integrators

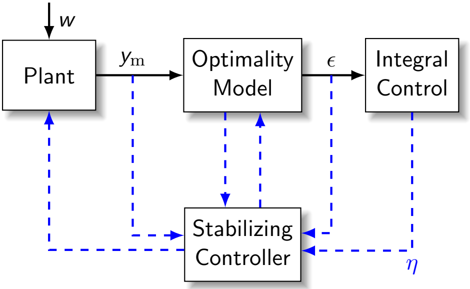


2. Can prove closed-loop stable $\implies$ OSS control problem solved
3. For QP OSS control, can prove cascade is stabilizable iff
   - plant stabilizable/detectable
   - optimization problem has a unique solution
   - engineering constraints not *redundant* with equilibrium constraints
   - T_0 has full column rank

Stabilizer Details I

Can we actually stabilize this thing?

1. **Goal**: stabilize *cascade* of plant → optimality model → integrators

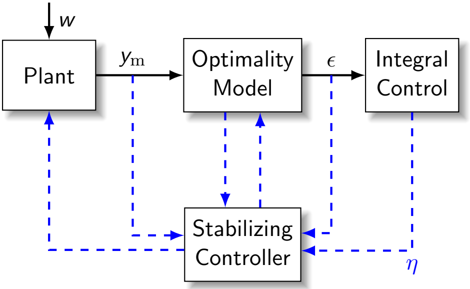


2. Can prove closed-loop stable $\Longrightarrow$ OSS control problem solved
3. For QP OSS control, can prove cascade is stabilizable iff
   - plant stabilizable/detectable
   - optimization problem has a unique solution
   - engineering constraints not *redundant* with equilibrium constraints
   - T_0 has full column rank

Stabilizer Details I

Can we actually stabilize this thing?

1. **Goal**: stabilize *cascade* of plant → optimality model → integrators

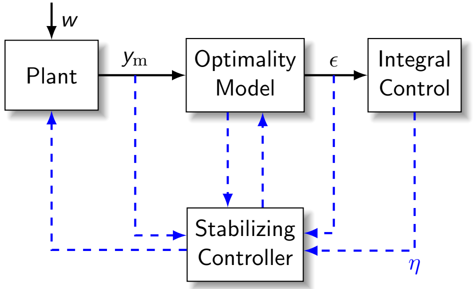


2. Can prove closed-loop stable $\Longrightarrow$ OSS control problem solved
3. For QP OSS control, can prove cascade is stabilizable iff
   - plant stabilizable/detectable
   - optimization problem has a unique solution
   - engineering constraints not *redundant* with equilibrium constraints
   - T_0 has full column rank

Stabilizer Details I

Can we actually stabilize this thing?

1. **Goal**: stabilize *cascade* of plant → optimality model → integrators

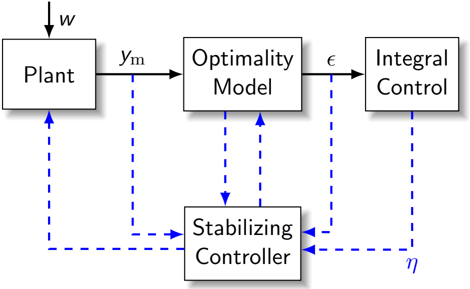


2. Can prove closed-loop stable $\Longrightarrow$ OSS control problem solved
3. For QP OSS control, can prove cascade is stabilizable iff
   - plant stabilizable/detectable
   - optimization problem has a unique solution
   - engineering constraints not *redundant* with equilibrium constraints
   - T_0 has full column rank

Stabilizer Details I

Can we actually stabilize this thing?

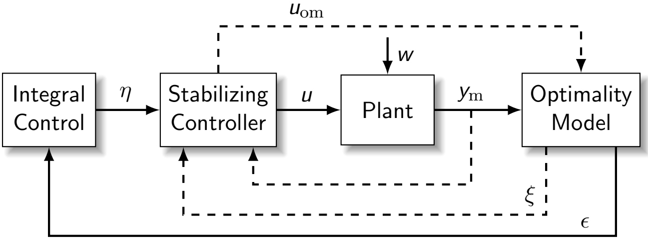
1. **Goal**: stabilize *cascade* of plant → optimality model → integrators



2. Can prove closed-loop stable $\Longrightarrow$ OSS control problem solved
3. For QP OSS control, can prove cascade is stabilizable iff
   - plant stabilizable/detectable
   - optimization problem has a unique solution
   - engineering constraints not *redundant* with equilibrium constraints
   - T_0 has full column rank

Stabilizer Details II

Optimality model contains monotone nonlinearity $\nabla f_0(y)$ . . .



Stabilizer design options:

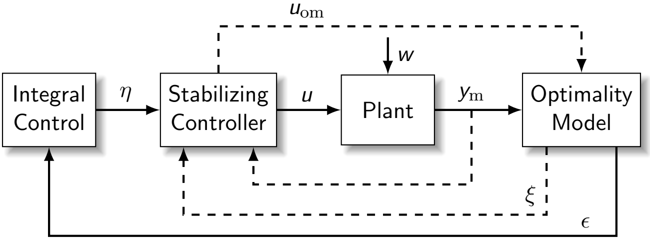
1. In *theory*: full-order robustly stabilizing controller design
2. In *practice*: low-gain integral control $u = -K\eta$ if open-loop stable, or any heuristic, e.g., linearize and do $\mathcal{H}_2$ design

Closed-loop stability analysis:

1. Robust stability (e.g., IQC-based) or time-scale separation

Stabilizer Details II

Optimality model contains monotone nonlinearity $\nabla f_0(y)$ ...



Stabilizer design options:

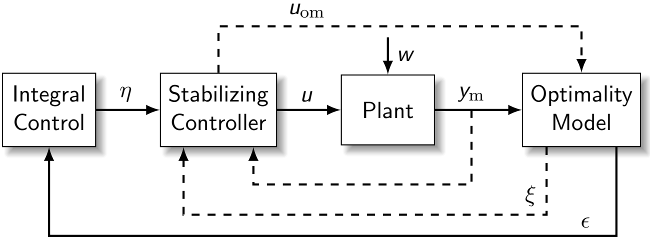
1. In *theory*: full-order robustly stabilizing controller design
2. In *practice*: low-gain integral control $u = -K\eta$ if open-loop stable, or any heuristic, e.g., linearize and do $\mathcal{H}_2$ design

Closed-loop stability analysis:

1. Robust stability (e.g., IQC-based) or time-scale separation

Stabilizer Details II

Optimality model contains monotone nonlinearity $\nabla f_0(y)$ ...



Stabilizer design options:

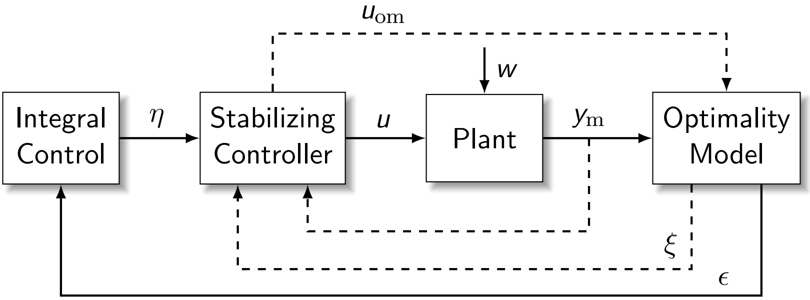
1. In *theory*: full-order robustly stabilizing controller design
2. In *practice*: low-gain integral control $u = -K\eta$ if open-loop stable, or any heuristic, e.g., linearize and do $\mathcal{H}_2$ design

Closed-loop stability analysis:

1. Robust stability (e.g., IQC-based) or time-scale separation

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

Application: Inexact Reference Tracking

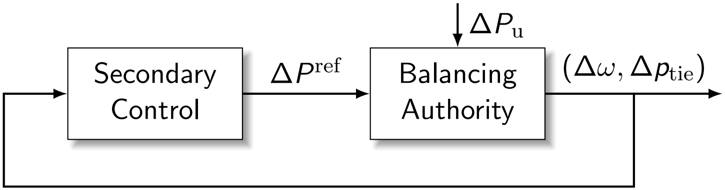
- Want **minimum error** asymptotic tracking of a (possibly infeasible) reference signal subject to actuator limits, e.g.

$$\begin{aligned} \underset{y_{\mathrm{m}},u}{\text{minimize}} \quad & \|y_{\mathrm{m}} - r\|_\infty \\ \text{subject to} \quad & (y_{\mathrm{m}}, u) \in \overline{Y}(w, \delta) \\ & \underline{u} \leq u \leq \overline{u} \end{aligned}$$

- If reference **feasible**, then exact tracking possible
- Could promote sparsity in steady-state control actions

Application: Frequency Control of Power Systems

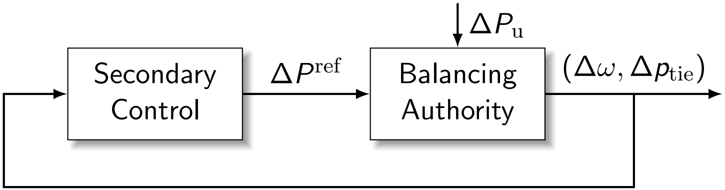
1. **Regulate** frequency of an interconnected AC power system in presence of unknown disturbances (locally balance supply and demand)



2. Modern challenges / opportunities:
   - variation due to RES $\Longrightarrow$ need fast control
   - inverter-based resources $\Longrightarrow$ fast actuation
   - new sensing, comm., comp. $\Longrightarrow$ new architectures

Application: Frequency Control of Power Systems

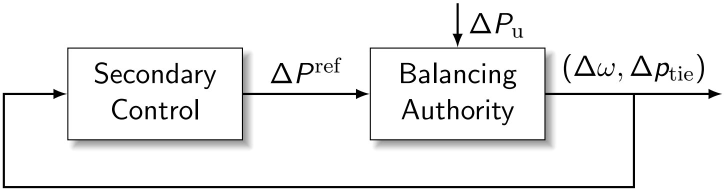
1. **Regulate** frequency of an interconnected AC power system in presence of unknown disturbances (locally balance supply and demand)



2. Modern challenges / opportunities:
   - variation due to RES $\Longrightarrow$ need fast control
   - inverter-based resources $\Longrightarrow$ fast actuation
   - new sensing, comm., comp. $\Longrightarrow$ new architectures

Application: Frequency Control of Power Systems

1. **Regulate** frequency of an interconnected AC power system in presence of unknown disturbances (locally balance supply and demand)



2. Modern challenges / opportunities:
   - variation due to RES $\implies$ need fast control
   - inverter-based resources $\implies$ fast actuation
   - new sensing, comm., comp. $\implies$ new architectures

Key insights into frequency control problem

1. For discussion, small-signal network of machines + turbine/gov

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

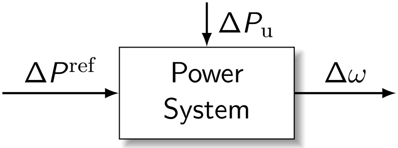
2. Model internally stable, DC gain analysis yields

$$
\Delta\omega_{\mathrm{ss}} = \frac{1}{\beta}\sum_i \left[\Delta P_i^{\mathrm{ref}} + \Delta P_{\mathrm{u},i}\right]
$$

where $\beta = \sum_i (D_i + R_{\mathrm{d},i}^{-1})$ is frequency stiffness.

3. Lots of **flexibility** in choice of ΔP^{ref}!

Optimal Allocation of Secondary Resources

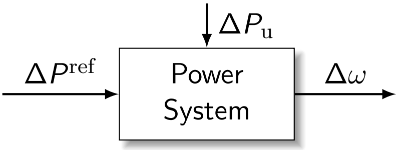


Allocate reserves $\Delta P_i^{\mathrm{ref}}$ subject to frequency regulation

$$\underset{\Delta P^{\mathrm{ref}} \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{i=1}^{n} C_i(\Delta P_i^{\mathrm{ref}})$$
$$\text{subject to } F\Delta\omega = \mathbb{0}$$

- This OSS problem satisfies the robust feasible subspace property
  $\Longrightarrow$ can construct (several) different optimality models!

Optimal Allocation of Secondary Resources



Allocate reserves $\Delta P_i^{\mathrm{ref}}$ subject to frequency regulation

$$\underset{\Delta P^{\mathrm{ref}} \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{i=1}^{n} C_i(\Delta P_i^{\mathrm{ref}})$$

$$\text{subject to } F \Delta\omega = \mathbb{0}$$

- This OSS problem satisfies the robust feasible subspace property
  $\Longrightarrow$ can construct (several) different optimality models!

OSS Framework Recovers Recent Controllers

1. Distributed Averaging PI Control

$$\epsilon_i = \Delta\omega_i - \sum_{j=1}^{n} a_{ij}(\nabla C_i(P_i^{\mathrm{ref}}) - \nabla C_j(P_j^{\mathrm{ref}}))$$
$$\dot{\eta}_i = \epsilon_i$$
$$P_i^{\mathrm{ref}} = \mathsf{Stabilizer}_i(\epsilon_i, \eta_i, \omega_i)$$

- Note: **many** architecture variations possible

2. AGC (stylized version)

$$\dot{\eta} = k \cdot \Delta\omega_{\mathrm{cc}}\,, \qquad P_i^{\mathrm{ref}} = (\nabla C_i)^{-1}(\eta)$$

3. Gather-and-broadacst (Dörfler & Grammatico)

$$\dot{\eta} = \frac{1}{n}\sum_{i=1}^{n} \Delta\omega_i\,, \qquad P_i^{\mathrm{ref}} = (\nabla C_i)^{-1}(\eta)$$

OSS Framework Recovers Recent Controllers

1. Distributed Averaging PI Control

$$\begin{aligned} \epsilon_i &= \Delta\omega_i - \sum\nolimits_{j=1}^{n} a_{ij}(\nabla C_i(P_i^{\mathrm{ref}}) - \nabla C_j(P_j^{\mathrm{ref}})) \\ \dot{\eta}_i &= \epsilon_i \\ P_i^{\mathrm{ref}} &= \mathsf{Stabilizer}_i(\epsilon_i, \eta_i, \omega_i) \end{aligned}$$

   - Note: **many** architecture variations possible

2. AGC (stylized version)

$$\dot{\eta} = k \cdot \Delta\omega_{\mathrm{cc}}\,, \qquad P_i^{\mathrm{ref}} = (\nabla C_i)^{-1}(\eta)$$

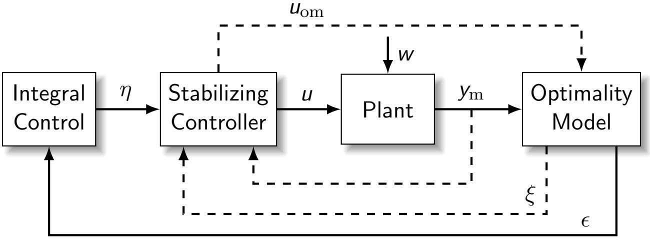
3. Gather-and-broadacst (Dörfler & Grammatico)

$$\dot{\eta} = \frac{1}{n}\sum_{i=1}^{n} \Delta\omega_i\,, \qquad P_i^{\mathrm{ref}} = (\nabla C_i)^{-1}(\eta)$$

Conclusions

Optimal Steady-State (OSS) Control framework

1. Robust feedback optimization of dynamic systems
2. Optimality model reduces OSS problem to output reg. problem



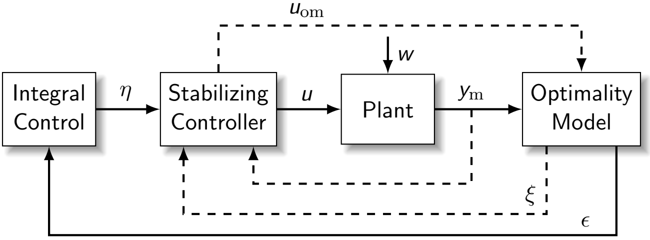
Many pieces of theory wide open . . .

1. Decentralized, hierarchical, competitive, . . .
2. Performance improvement (e.g., feedforward, anti-windup)

Conclusions

Optimal Steady-State (OSS) Control framework

1. Robust feedback optimization of dynamic systems
2. Optimality model reduces OSS problem to output reg. problem



Many pieces of theory wide open . . .

1. Decentralized, hierarchical, competitive, . . .
2. Performance improvement (e.g., feedforward, anti-windup)

Details in paper on arXiv

SUBMITTED TO IEEE TRANSACTIONS ON AUTOMATIC CONTROL. THIS VERSION: OCTOBER 15, 2018 1

The Optimal Steady-State Control Problem

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

https://arxiv.org/abs/1810.12892

Liam S. P. Lawrence
University of Waterloo

Enrique Mallada
John's Hopkins Univ.

Questions

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

appendix

Feedforward vs. Feedback Optimization

Property	**Feedforward**	**Feedback**
Setpoint Quality	≈ Optimal	≈ Optimal
High-Fidelity Model	Crucial	Not crucial
Robustness	No	Yes
Feedback Design/Analysis	Unchanged	More difficult
Computational Effort	Moderate	???

MPC: high computational effort, difficult analysis ⇒ Alternatives?

Compared to MPC, if we **give a bit** on *trajectory* optimality, can we can **gain a lot** on ease of design, analysis, and implementation?

Here is a first cut of such an approach.

Feedforward vs. Feedback Optimization

Property	**Feedforward**	**Feedback**
Setpoint Quality	≈ Optimal	≈ Optimal
High-Fidelity Model	Crucial	Not crucial
Robustness	No	Yes
Feedback Design/Analysis	Unchanged	More difficult
Computational Effort	Moderate	???

MPC: high computational effort, difficult analysis ⇒ Alternatives?

Compared to MPC, if we **give a bit** on *trajectory* optimality, can we can **gain a lot** on ease of design, analysis, and implementation?

Here is a first cut of such an approach.

Feedforward vs. Feedback Optimization

Property	Feedforward	Feedback
Setpoint Quality	$\approx$ Optimal	$\approx$ Optimal
High-Fidelity Model	Crucial	Not crucial
Robustness	No	Yes
Feedback Design/Analysis	Unchanged	More difficult
Computational Effort	Moderate	???

MPC: high computational effort, difficult analysis $\Rightarrow$ Alternatives?

Compared to MPC, if we **give a bit** on *trajectory* optimality, can we can **gain a lot** on ease of design, analysis, and implementation?

Here is a first cut of such an approach.

Feedforward vs. Feedback Optimization

Property	**Feedforward**	**Feedback**
Setpoint Quality	$\approx$ Optimal	$\approx$ Optimal
High-Fidelity Model	Crucial	Not crucial
Robustness	No	Yes
Feedback Design/Analysis	Unchanged	More difficult
Computational Effort	Moderate	???

MPC: high computational effort, difficult analysis $\Rightarrow$ Alternatives?

Compared to MPC, if we **give a bit** on *trajectory* optimality, can we can **gain a lot** on ease of design, analysis, and implementation?

Here is a first cut of such an approach.

Feedforward vs. Feedback Optimization

Property	**Feedforward**	**Feedback**
Setpoint Quality	$\approx$ Optimal	$\approx$ Optimal
High-Fidelity Model	Crucial	Not crucial
Robustness	No	Yes
Feedback Design/Analysis	Unchanged	More difficult
Computational Effort	Moderate	???

MPC: high computational effort, difficult analysis $\Rightarrow$ Alternatives?

Compared to MPC, if we **give a bit** on *trajectory* optimality, can we can **gain a lot** on ease of design, analysis, and implementation?

Here is a first cut of such an approach.

Feedforward vs. Feedback Optimization

Property	**Feedforward**	**Feedback**
Setpoint Quality	$\approx$ Optimal	$\approx$ Optimal
High-Fidelity Model	Crucial	Not crucial
Robustness	No	Yes
Feedback Design/Analysis	Unchanged	More difficult
Computational Effort	Moderate	???

MPC: high computational effort, difficult analysis $\Rightarrow$ Alternatives?

Compared to MPC, if we **give a bit** on *trajectory* optimality, can we can **gain a lot** on ease of design, analysis, and implementation?

Here is a first cut of such an approach.

Feedforward vs. Feedback Optimization

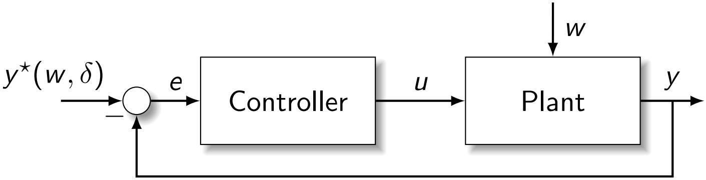
Property	**Feedforward**	**Feedback**
Setpoint Quality	$\approx$ Optimal	$\approx$ Optimal
High-Fidelity Model	Crucial	Not crucial
Robustness	No	Yes
Feedback Design/Analysis	Unchanged	More difficult
Computational Effort	Moderate	???

MPC: high computational effort, difficult analysis $\Rightarrow$ Alternatives?

Compared to MPC, if we **give a bit** on *trajectory* optimality, can we can **gain a lot** on ease of design, analysis, and implementation?

Here is a first cut of such an approach.

Is OSS Control just a standard tracking problem?

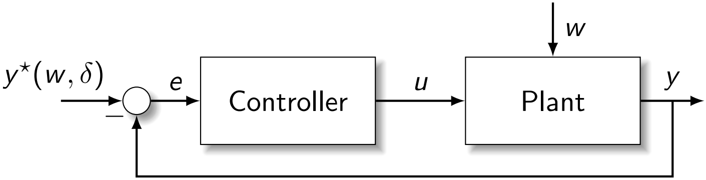


We want y to track $y^\star(w,\delta)$, but two problems:

1. unmeasured components of w change $y^\star$
2. $y^\star$ depends on uncertainty δ (relevant if $\overline{Y} \subset \mathbb{R}^p$)

Standard tracking approach **infeasible** for quickly varying $w(t)$, or large uncertainties δ, or particular choices of regulated outputs

Is OSS Control just a standard tracking problem?

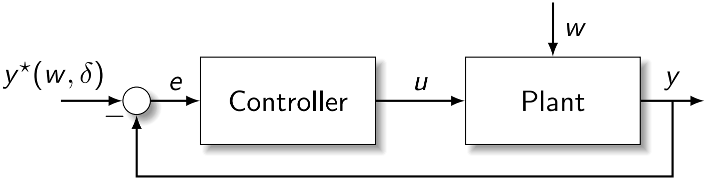


We want y to track $y^\star(w,\delta)$, but two problems:

1. unmeasured components of w change $y^\star$
2. $y^\star$ depends on uncertainty δ (relevant if $\overline{Y} \subset \mathbb{R}^p$)

Standard tracking approach **infeasible** for quickly varying $w(t)$, or large uncertainties δ, or particular choices of regulated outputs

Is OSS Control just a standard tracking problem?



We want y to track $y^\star(w,\delta)$, but two problems:

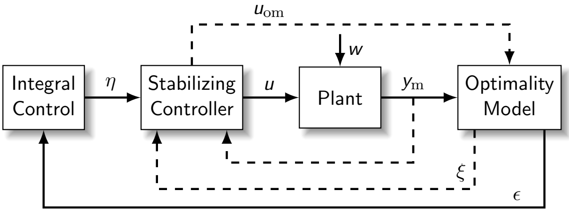
1. unmeasured components of w change $y^\star$
2. $y^\star$ depends on uncertainty δ (relevant if $\overline{Y} \subset \mathbb{R}^p$)

Standard tracking approach **infeasible** for quickly varying $w(t)$, or large uncertainties δ, or particular choices of regulated outputs

Towards an internal model principle . . .

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

$$\mathrm{range}(T_0) = \mathsf{V}(\delta) \cap \mathrm{null}(H)$$



Interpretation: Exact robust asymptotic optimization achieved if loop *incorporates a model of the optimal set of the optimization problem*

Slide on EOA Approach . . .

Example 1: Necessity of Equilibrium Constraints

Consider the OSS control problem:

1. Dynamics:

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

$$y = \begin{bmatrix} x_1 \\ u \end{bmatrix}$$

2. Optimization problem:

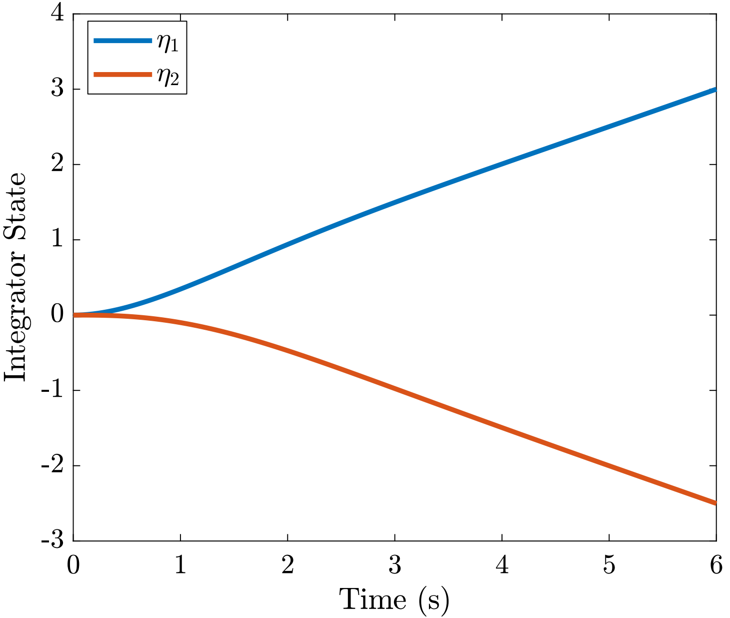
$$\underset{y \in \mathbb{R}^2}{\text{minimize}} \quad g(y) := \frac{1}{2} y_1^2 + \frac{1}{2} y_2^2$$

What happens if we omit the equilibrium constraints?

$$\dot{\eta} = \nabla f_0(y)$$

$$u = -K\eta$$

Example 1: Necessity of Equilibrium Constraints (cont.)



Example 2: Necessity of Robust Feasible Subspace

Consider the OSS control problem:

1. Dynamics:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1-\delta & 0 \\ 1+\delta & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w$$

$$y = \begin{bmatrix} x_1 \\ u \end{bmatrix}$$

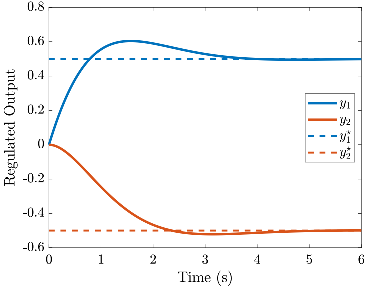
2. Optimization problem:

$$\begin{aligned} \underset{y \in \mathbb{R}^2}{\text{minimize}} \quad & \frac{1}{2} y_1^2 + \frac{1}{2} y_2^2 \\ \text{subject to} \quad & y \in \overline{Y}(w,\delta) = \mathrm{y}(w,\delta) + V(\delta) \end{aligned}$$

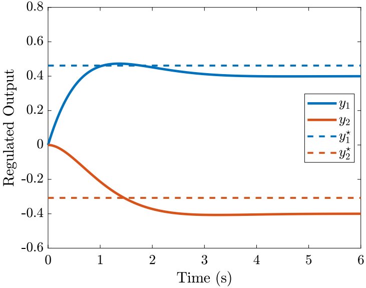
We can show $V(\delta) = \operatorname{span}\left\{ \begin{bmatrix} 1 \\ \delta \end{bmatrix} \right\} \Rightarrow V(\delta)$ **dependent** on δ.

Example 2: Necessity of Robust Feasible Subspace (cont.)

- We apply our scheme anyway supposing $\delta = 0$
- Optimality model + integral control yields...



If $\delta = 0$ in the true plant
$\Rightarrow$ achieve optimal cost of 0.1538.



If $\delta = 0.5$ in the true plant
$\Rightarrow$ achieve sub-optimal cost of 0.1599.

Robust Output Subspace Optimality Model

If furthermore $V(\delta)$ itself is independent of δ, then

$$\begin{aligned}\dot{\mu} &= Hy - Lw\\ \dot{\nu} &= \boldsymbol{max}(\nu + Jy - Mw, \mathbb{0}) - \nu\\ \epsilon &= \boldsymbol{R_0}^\mathsf{T}(\nabla f_0(y,w) + H^\mathsf{T}\mu + J^\mathsf{T}\nu)\end{aligned}$$

range $R_0 = \mathsf{V}(\delta)$

(Design freedom!)

is also an optimality model for the LTI-Convex OSS Control Problem.

1. Can take $R_0 = I$ if $V(\delta) = \mathbb{R}^p$, which holds if

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ has full row rank } \iff \text{No transmission zeros at } s = 0$$

2. Again, different equivalent formulations of optimization problem give different optimality models

Robust Output Subspace Optimality Model

If furthermore $V(\delta)$ itself is independent of δ, then

$$\dot{\mu} = Hy - Lw$$
$$\dot{\nu} = \boldsymbol{max}(\nu + Jy - Mw, \mathbb{0}) - \nu$$
$$\epsilon = \boldsymbol{R_0}^{\mathsf{T}}(\nabla f_0(y, w) + H^{\mathsf{T}}\mu + J^{\mathsf{T}}\nu)$$

range $R_0 = \mathrm{V}(\delta)$

(Design freedom!)

is also an optimality model for the LTI-Convex OSS Control Problem.

1. Can take $R_0 = I$ if $V(\delta) = \mathbb{R}^p$, which holds if

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ has full row rank } \iff \text{No transmission zeros at } s = 0$$

2. Again, different equivalent formulations of optimization problem give different optimality models

Robust Output Subspace Optimality Model

If furthermore $V(\delta)$ itself is independent of δ, then

$$\dot{\mu} = Hy - Lw$$
$$\dot{\nu} = \boldsymbol{max}(\nu + Jy - Mw, \mathbb{0}) - \nu$$
$$\epsilon = \boldsymbol{R_0}^\mathsf{T}(\nabla f_0(y, w) + H^\mathsf{T}\mu + J^\mathsf{T}\nu)$$

range $R_0 = \mathrm{V}(\delta)$

(Design freedom!)

is also an optimality model for the LTI-Convex OSS Control Problem.

1. Can take $R_0 = I$ if $V(\delta) = \mathbb{R}^p$, which holds if

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ has full row rank} \quad \Longleftrightarrow \quad \text{No transmission zeros at } s = 0$$

2. Again, different equivalent formulations of optimization problem give different optimality models

OSS Control in the Literature

The OSS controller architecture found throughout the literature on **real-time optimization**.

Problem [Nelson and Mallada '18]

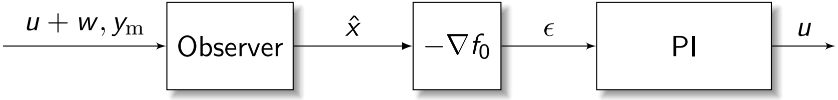
Design a feedback controller to drive the system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B(u(t) + w) \\ y_{\mathrm{m}}(t) &= Cx(t) + D(u(t) + w)\end{aligned}$$

to the solution of the optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f(x)\,.$$

OSS Control in the Literature (cont.)



Controller Design

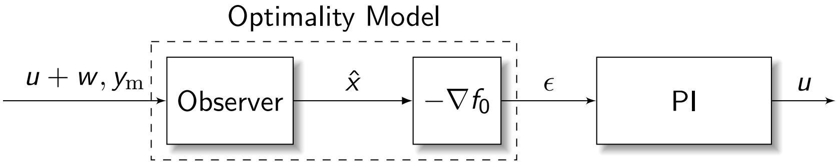
The **optimality model** is an observer with gradient output

$$\begin{aligned} \dot{\hat{x}} &= (A - LC)\hat{x} + (B - LD)(u + w) + Ly_{\mathrm{m}} \\ \epsilon &= -\nabla f_0(\hat{x})\,. \end{aligned}$$

A PI controller serves as **internal model and stabilizer**

$$\dot{e}_I = \epsilon\,, \quad u = K_I e_I + K_p \epsilon\,.$$

OSS Control in the Literature (cont.)



Controller Design

The **optimality model** is an observer with gradient output

$$\begin{aligned}\dot{\hat{x}} &= (A - LC)\hat{x} + (B - LD)(u + w) + Ly_{\mathrm{m}} \\ \epsilon &= -\nabla f_0(\hat{x})\,.\end{aligned}$$

A PI controller serves as **internal model and stabilizer**

$$\dot{e}_I = \epsilon\,, \quad u = K_I e_I + K_p \epsilon\,.$$