The Optimal Steady-State Control Problem

John W. Simpson-Porco

UNIVERSITY OF WATERLOO

Automatic Control Laboratory
ETH Zürich

January 29, 2019

This talk is based on this paper

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*

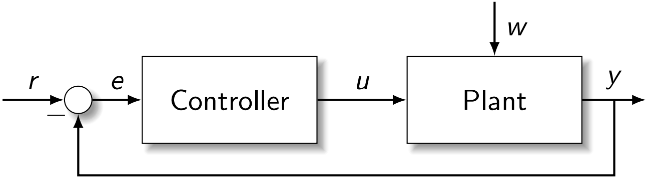
Submitted to IEEE Transactions on Automatic Control

Liam S. P. Lawrence
University of Waterloo

Enrique Mallada
John's Hopkins Univ.

Control Systems 101

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

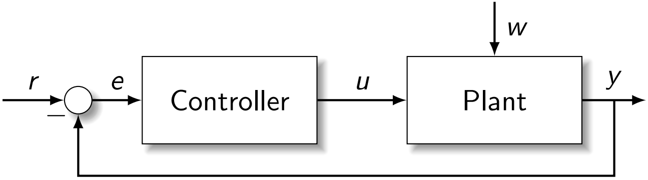


- **Exact robust asymptotic tracking** achieved if loop gain *"incorporates . . . a suitably reduplicated model of the dynamic structure of the exogenous signal"*

How is the reference r being determined?

Control Systems 101

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

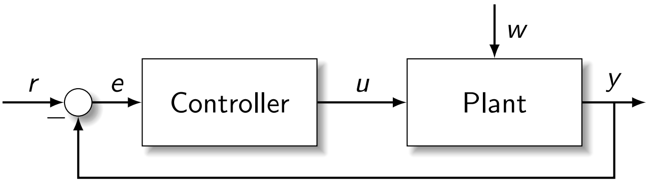


- **Exact robust asymptotic tracking** achieved if loop gain *"incorporates . . . a suitably reduplicated model of the dynamic structure of the exogenous signal"*

How is the reference r being determined?

Control Systems 101

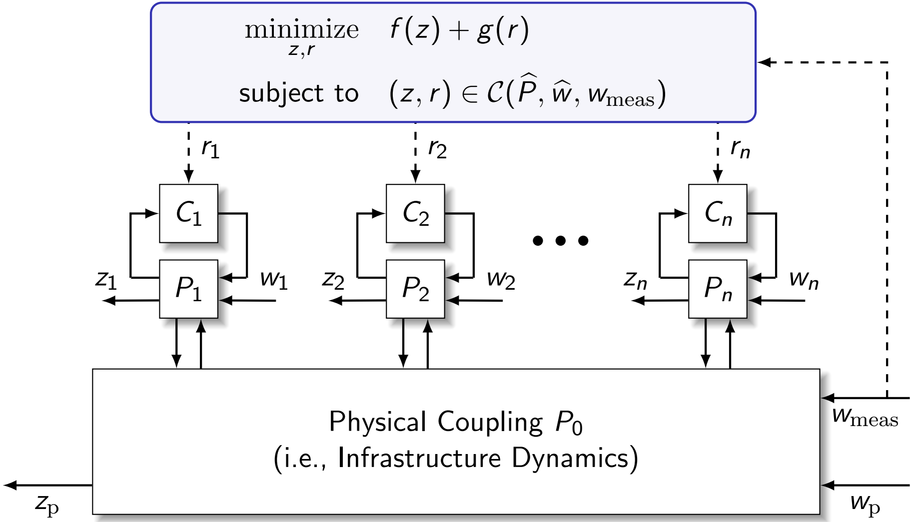
- Prototypical feedback control problem is **tracking** and **disturbance rejection** in the presence of non-negligible **model uncertainty**



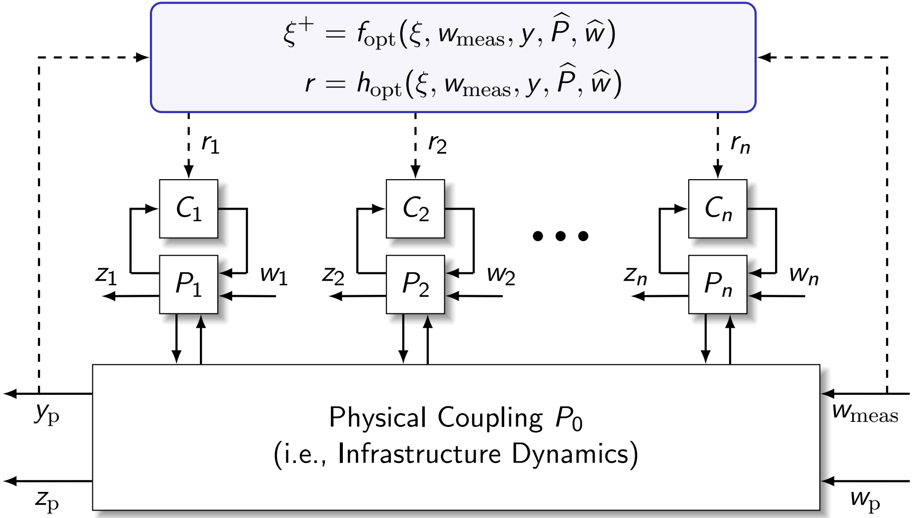
- **Exact robust asymptotic tracking** achieved if loop gain *"incorporates . . . a suitably reduplicated model of the dynamic structure of the exogenous signal"*

How is the reference r being determined?

Feedforward Optimization of Large-Scale Systems



Feedback Optimization of Large-Scale Systems



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 feedback design/analysis and computational effort?
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 feedback design/analysis and computational effort?
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 feedback design/analysis and computational effort?
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 feedback design/analysis and computational effort?
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 feedback design/analysis and computational effort?
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 feedback design/analysis and computational effort?
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 feedback design/analysis and computational effort?
Here is a first cut of such an approach.

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - an 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

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

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - an 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

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

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - an 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

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

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - an 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

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

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - an 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

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

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - an 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

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

Optimal Steady-State Control Problem Statement

Given:

1. a dynamic system model with
   - a *class* of external disturbances $w(t)$
   - an 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

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

LTI-Convex OSS Control: Setup Overview

1. **Uncertain LTI dynamics**

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

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

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

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

LTI-Convex OSS Control: Setup Overview

1. **Uncertain LTI dynamics**

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

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

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

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

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 model matrices (A, B, C, D, B_w, Q)

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

constraint $\iff \bar{y} \in \overline{Y}(w, \delta)$ 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 model matrices (A, B, C, D, B_w, Q)

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

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

LTI-Convex OSS Control: Setup II

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

$$
\begin{aligned}
\underset{y\in\mathbb{R}^p}{\text{minimize}} \quad & f(y,w) && \text{(convex cost)}\\
\text{subject to} \quad & y \in \overline{Y}(w,\delta) = \mathrm{y}(w,\delta) + V(\delta) && \text{(equilibrium)}\\
& Hy = Lw && \text{(engineering equality)}\\
& k_i(y,w) \le 0 && \text{(engineering inequality)}
\end{aligned}
$$

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

$$\begin{aligned} &\underset{y\in\mathbb{R}^p}{\text{minimize}} && f(y,w) && \text{(convex cost)}\\ &\text{subject to} && y\in\overline{Y}(w,\delta)=\mathrm{y}(w,\delta)+V(\delta) && \text{(equilibrium)}\\ & && Hy=Lw && \text{(engineering equality)}\\ & && k_i(y,w)\le 0 && \text{(engineering inequality)}\end{aligned}$$

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

$$\begin{aligned}
\underset{y\in\mathbb{R}^p}{\text{minimize}} \quad & f(y,w) && \text{(convex cost)}\\
\text{subject to} \quad & y \in \overline{Y}(w,\delta) = \mathrm{y}(w,\delta) + V(\delta) && \text{(equilibrium)}\\
& Hy = Lw && \text{(engineering equality)}\\
& k_i(y,w) \leq 0 && \text{(engineering inequality)}
\end{aligned}$$

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

$$
\begin{aligned}
\underset{y\in\mathbb{R}^p}{\text{minimize}} \quad & f(y,w) && \text{(convex cost)}\\
\text{subject to} \quad & y \in \overline{Y}(w,\delta) = \mathrm{y}(w,\delta) + V(\delta) && \text{(equilibrium)}\\
& Hy = Lw && \text{(engineering equality)}\\
& k_i(y,w) \leq 0 && \text{(engineering inequality)}
\end{aligned}
$$

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

$$\begin{array}{lll} \underset{y\in\mathbb{R}^p}{\text{minimize}} & f(y,w) & \text{(convex cost)} \\ \text{subject to} & y \in \overline{Y}(w,\delta) = \mathrm{y}(w,\delta) + V(\delta) & \text{(equilibrium)} \\ & Hy = Lw & \text{(engineering equality)} \\ & k_i(y,w) \leq 0 & \text{(engineering inequality)} \end{array}$$

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 $y^\star(w,\delta)$

Optimality Models for OSS Control

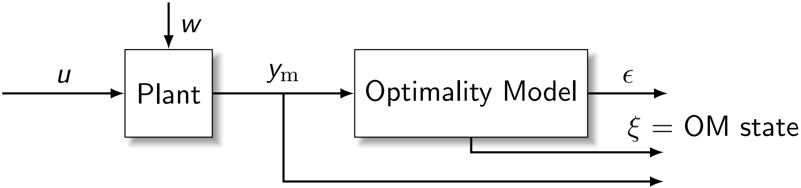
An **optimality model** filters the available measurements to robustly produce a proxy error ϵ for the true tracking error $e = y^\star(w, \delta) - y$

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 $e = y^{\star}(w, \delta) - y$

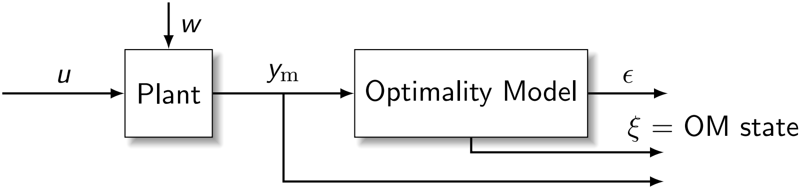


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^{\star}(w, \delta)$.

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 $e = y^\star(w, \delta) - y$

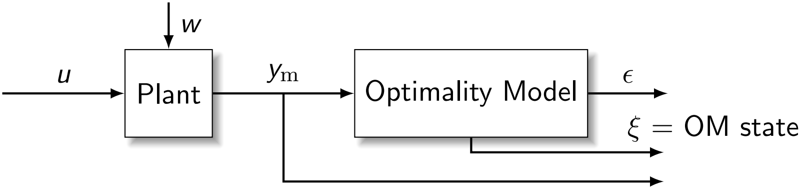


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^\star(w, \delta)$.

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

Optimality Models for OSS Control

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

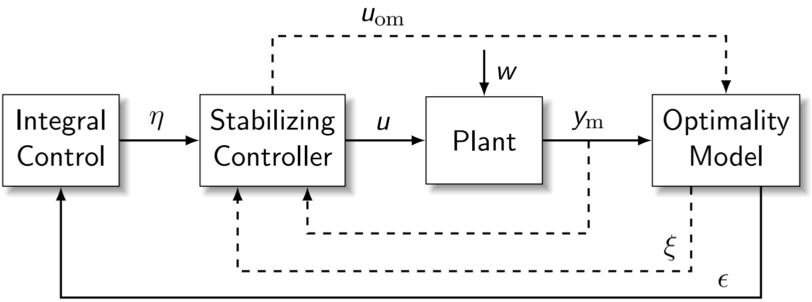


Steady-state requirement: if the plant and optimality model are both in equilibrium and $\epsilon = \mathbb{0}$, then $y = y^\star(w, \delta)$.

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

General Architecture for OSS Control

Optimality model reduces OSS control to output regulation



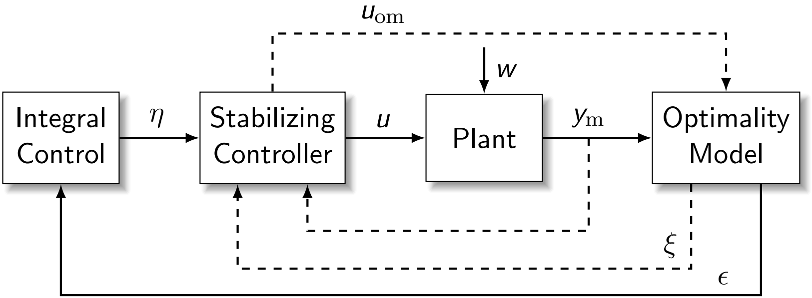
Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

General Architecture for OSS Control

Optimality model reduces OSS control to output regulation



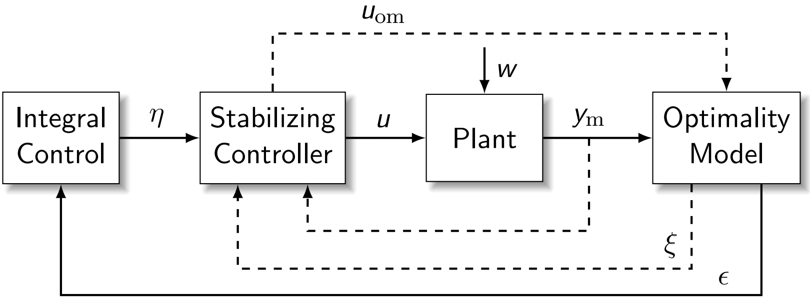
Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

General Architecture for OSS Control

Optimality model reduces OSS control to output regulation



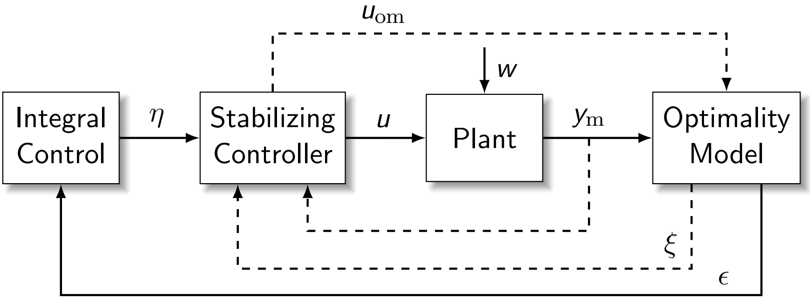
Optimality Model: creates proxy error signal ϵ

Integral Control: integrates ϵ

Stabilizing Controller: stabilizes closed-loop system

General Architecture 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

Robustness Issues in Constructing Optimality Models

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

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

Optimality condition:

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

possibly depends on uncertain parameter δ.

Robustness Issues in Constructing Optimality Models

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

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

Optimality condition:

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

possibly depends on uncertain parameter δ.

Robustness Issues in Constructing Optimality Models

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

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

Optimality condition:

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

possibly depends on uncertain parameter δ.

Robustness Issues (cont.)

When can an optimality model encode the gradient KKT condition?

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

Robust Feasible Subspace Property

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

Robustness Issues (cont.)

When can an optimality model encode the gradient KKT condition?

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

Robust Feasible Subspace Property

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

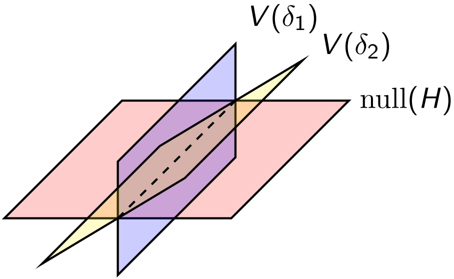
Robustness Issues (cont.)

When can an optimality model encode the gradient KKT condition?

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

Robust Feasible Subspace Property

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



Robust Feasible Subspace Optimality Model

If Robust Feasible Subspace property holds, then

$$\dot{\nu} = \boldsymbol{max}(\nu + Jy - Mw, \mathbb{0}) - \nu$$

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

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

(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 $\mathsf{V}(\delta) \cap \operatorname{null}(H)$:

$$\epsilon_2 = \mathbb{0} \iff \nabla f(y, w) + J^\mathsf{T}\nu \perp V(\delta) \cap \operatorname{null}(H)$$

2. Different equivalent formulations of optimization problem give different optimality models

Robust Feasible Subspace Optimality Model

If Robust Feasible Subspace property holds, then

$$\dot{\nu} = \boldsymbol{max}(\nu + Jy - Mw, \mathbb{0}) - \nu$$
$$\epsilon = \begin{bmatrix} Hy - Lw \\ \boldsymbol{T_0}^{\mathsf{T}}(\nabla f(y, w) + J^{\mathsf{T}}\nu) \end{bmatrix}$$

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

(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 $\mathsf{V}(\delta) \cap \mathrm{null}(H)$:

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

2. Different equivalent formulations of optimization problem give different optimality models

Robust Feasible Subspace Optimality Model

If Robust Feasible Subspace property holds, then

$$\dot{\nu} = \boldsymbol{max}(\nu + Jy - Mw, \mathbb{0}) - \nu$$
$$\epsilon = \begin{bmatrix} Hy - Lw \\ \boldsymbol{T_0}^\mathsf{T}(\nabla f(y,w) + J^\mathsf{T}\nu) \end{bmatrix}$$

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

(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 $\mathsf{V}(\delta) \cap \mathrm{null}(H)$:

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

2. 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

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

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

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

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

(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

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

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

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

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

(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

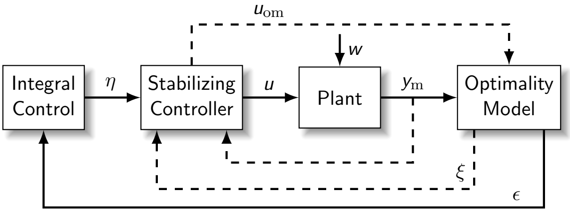
$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ has full row rank} \iff \text{No transmission zeros at } s = 0$$

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

Towards an internal model principle ...

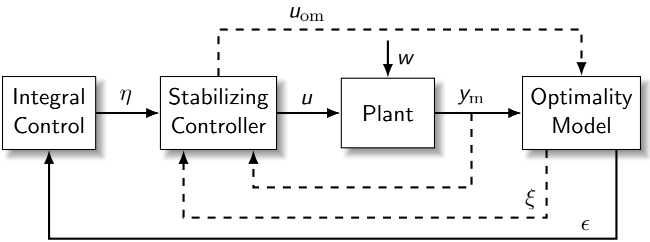
$$\epsilon = \begin{bmatrix} Hy - Lw \\ T_0^{\mathsf{T}} \nabla f(y, w) \end{bmatrix}$$

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



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

Stabilizer Design



Stabilizer design options:

1. full-order dynamic robust controller synthesis
2. low-gain integral control $u = -K\eta$ (Davison '76)
3. any heuristic, local linearized LQ design, ...

Closed-loop analysis:

1. Time-scale separation, robust control

Optimal Frequency Regulation Problem

1. Linearized dynamics of **network of generators**

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

2. Economically select equilibrium reserve powers $\Delta P_i^{\mathrm{ref}}$ subject to balance of supply and demand (control and disturbances)

$$
\begin{aligned}
&\underset{\Delta P_i^{\mathrm{ref}} \in \mathbb{R}^n}{\text{minimize}} \quad \sum\nolimits_{i=1}^{n} J_i(P_i^{\mathrm{ref}} + \Delta P_i^{\mathrm{ref}}) \\
&\text{subject to} \sum\nolimits_{i=1}^{n} (\Delta P_{\mathrm{u},i} + \Delta P_i^{\mathrm{ref}}) = 0
\end{aligned}
$$

Optimal Frequency Regulation Problem

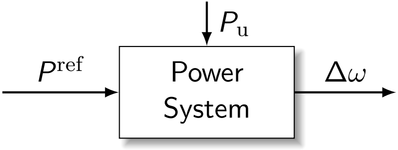
1. Linearized dynamics of **network of generators**

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

2. Economically select equilibrium reserve powers $\Delta P_i^{\mathrm{ref}}$ subject to balance of supply and demand (control and disturbances)

$$
\begin{aligned}
\underset{\Delta P_i^{\mathrm{ref}} \in \mathbb{R}^n}{\text{minimize}} \quad & \sum\nolimits_{i=1}^{n} J_i(P_i^{\mathrm{ref}} + \Delta P_i^{\mathrm{ref}}) \\
\text{subject to} \; & \sum\nolimits_{i=1}^{n} (\Delta P_{\mathrm{u},i} + \Delta P_i^{\mathrm{ref}}) = 0
\end{aligned}
$$

Optimal Frequency Regulation Problem



Equivalent formulations of steady-state dispatch problem

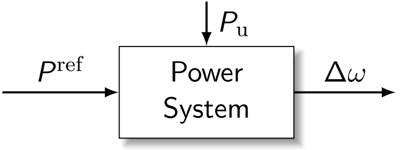
$$\underset{P^\mathrm{ref} \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{i=1}^{n} J_i(P_i^\mathrm{ref})$$
$$\text{subject to} \sum_{i=1}^{n} (P_{\mathrm{u},i} + P_i^\mathrm{ref}) = 0$$

(ROS Property ✓)

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

(RFS Property ✓)

Optimal Frequency Regulation Problem



Equivalent formulations of steady-state dispatch problem

$$\underset{P^{\mathrm{ref}} \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{i=1}^{n} J_i(P_i^{\mathrm{ref}})$$
$$\text{subject to} \sum_{i=1}^{n} (P_{\mathrm{u},i} + P_i^{\mathrm{ref}}) = 0$$

(ROS Property ✓)

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

(RFS Property ✓)

OSS Framework Recovers Recent Controllers

1. Distributed-Averaging Integral Control

$$k_i\dot{\eta}_i = -\Delta\omega_i - \sum_{j=1}^{n} a_{ij}(\eta_i - \eta_j)\,, \quad P_i^{\mathrm{ref}} = (\nabla J_i)^{-1}(\eta_i)$$

2. AGC / Gather-and-Broadcast Control

$$\dot{\eta} = \mathrm{average}(\omega_i)\,, \qquad P_i^{\mathrm{ref}} = (\nabla J_i)^{-1}(\eta)$$

3. Primal-dual algorithm

$$\dot{\mu}_i = -\nabla J_i(\mu_i) - \nu\,, \qquad P_i^{\mathrm{ref}} = \mu_i$$

$$\dot{\nu} = \sum_{i=1}^{n}(P_{\mathrm{u},i} + \mu_i)$$

OSS Framework Recovers Recent Controllers

1. Distributed-Averaging Integral Control

$$k_i\dot{\eta}_i = -\Delta\omega_i - \sum_{j=1}^{n} a_{ij}(\eta_i - \eta_j)\,, \quad P_i^{\mathrm{ref}} = (\nabla J_i)^{-1}(\eta_i)$$

2. AGC / Gather-and-Broadcast Control

$$\dot{\eta} = \text{average}(\omega_i)\,, \qquad P_i^{\mathrm{ref}} = (\nabla J_i)^{-1}(\eta)$$

3. Primal-dual algorithm

$$\dot{\mu}_i = -\nabla J_i(\mu_i) - \nu\,, \qquad P_i^{\mathrm{ref}} = \mu_i$$
$$\dot{\nu} = \sum_{i=1}^{n}(P_{\mathrm{u},i} + \mu_i)$$

OSS Framework Recovers Recent Controllers

1. Distributed-Averaging Integral Control

$$k_i\dot{\eta}_i = -\Delta\omega_i - \sum_{j=1}^{n} a_{ij}(\eta_i - \eta_j)\,, \quad P_i^{\mathrm{ref}} = (\nabla J_i)^{-1}(\eta_i)$$

2. AGC / Gather-and-Broadcast Control

$$\dot{\eta} = \text{average}(\omega_i)\,, \qquad P_i^{\mathrm{ref}} = (\nabla J_i)^{-1}(\eta)$$

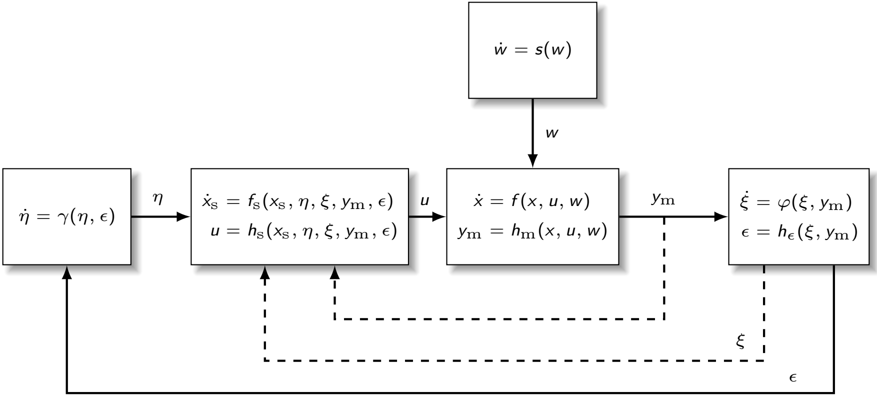
3. Primal-dual algorithm

$$\dot{\mu}_i = -\nabla J_i(\mu_i) - \nu\,, \qquad P_i^{\mathrm{ref}} = \mu_i$$

$$\dot{\nu} = \sum_{i=1}^{n}(P_{\mathrm{u},i} + \mu_i)$$

Nonlinear OSS Control Problem

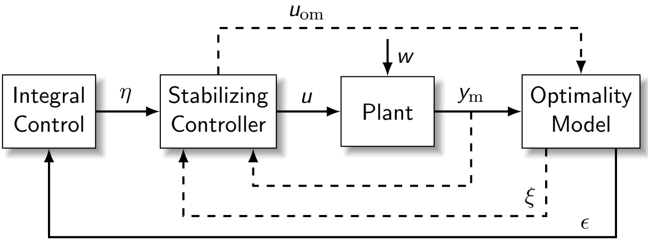
Nonlinear systems with **time-varying** disturbances in continuous or discrete-time



Conclusions

New control framework: Optimal Steady-State (OSS) Control

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



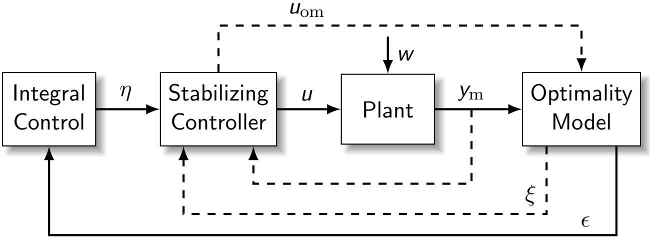
Many pieces of theory wide open ...

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

Conclusions

New control framework: Optimal Steady-State (OSS) Control

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



Many pieces of theory wide open ...

1. Sampled-data, 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

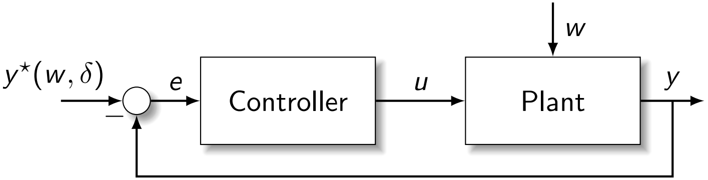
Enrique Mallada
John's Hopkins Univ.

Questions

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

appendix

Is OSS Control just a standard tracking problem?

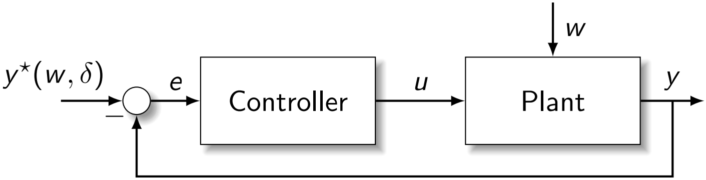


We want y to track $y^\star(w,\delta)$, but two problems:

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?

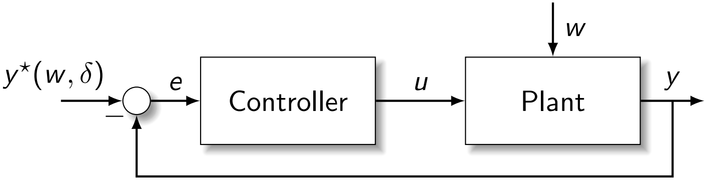


We want y to track $y^\star(w,\delta)$, but two problems:

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

Slide on EOA Approach . . .

Example 1: Necessity of Equilibrium Constraints

Consider the OSS control problem:

1. Dynamics:

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

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

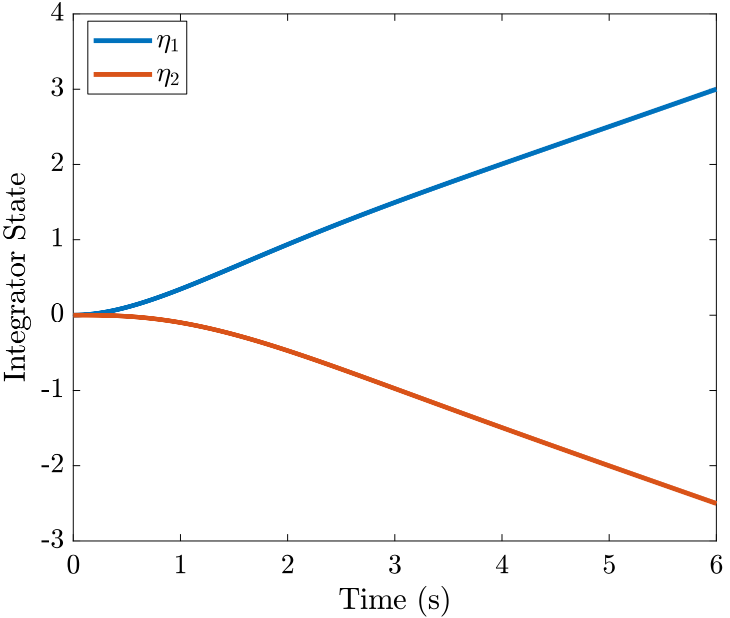
2. Optimization problem:

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

What happens if we omit the equilibrium constraints?

$$\dot{\eta} = \nabla g(y)$$
$$u = -K\eta$$

Example 1: Necessity of Equilibrium Constraints (cont.)



Example 2: Necessity of Robust Feasible Subspace

Consider the OSS control problem:

1. Dynamics:

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

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

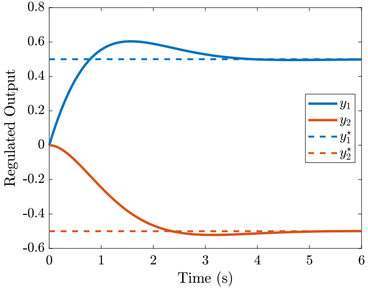
2. Optimization problem:

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

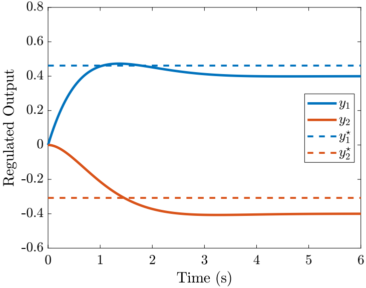
We can show $V(\delta) = \operatorname{span}\left\{ \begin{bmatrix} 1 \\ \delta \end{bmatrix} \right\} \Rightarrow V(\delta)$ **dependent** on δ.

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.

OSS Control in the Literature

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

Problem [Nelson and Mallada '18]

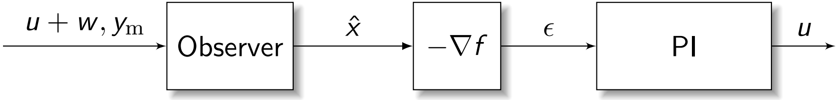
Design a feedback controller to drive the system

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

to the solution of the optimization problem

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

OSS Control in the Literature (cont.)



Controller Design

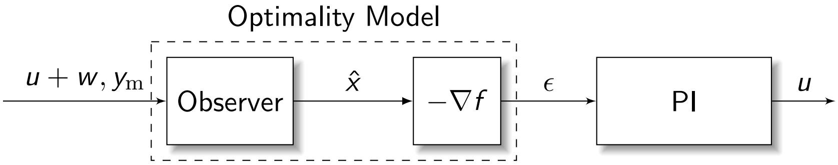
The **optimality model** is an observer with gradient output

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

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

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

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

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