A robust optimization approach for dynamic input allocation

22nd World Congress of the International Federation of Automatic Control

A. Tenaglia, F. Oliva, M. Sassano, S. Galeani, D. Carnevale
alessandro.tenaglia@uniroma2.it

University of Rome Tor Vergata
Department of Civil Engineering and Computer Science Engineering

July 14, 2023

Roadmap

1. Input allocation
2. Robust input allocation
3. Numerical simulations
4. Conclusions

Roadmap

Input allocation

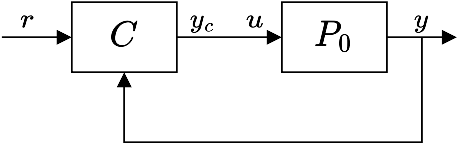

Figure 1: The closed-loop system Σ_0.

- Consider a **nominal LTI plant** P_0, where $u \in \mathbb{R}^m$ is the plant input, $y \in \mathbb{R}^p$ is the plant output and $m > p$.

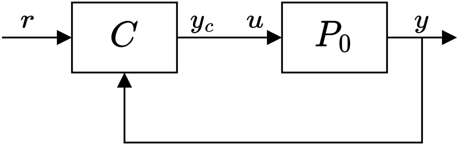

Figure 1: The closed-loop system Σ_0.

- Consider a **nominal LTI plant** P_0, where $u \in \mathbb{R}^m$ is the plant input, $y \in \mathbb{R}^p$ is the plant output and $m > p$.
- Consider a **controller** C, where $r \in \mathbb{R}^p$ is a constant reference signal and $y_c \in \mathbb{R}^m$ is the controller output.

Input allocation

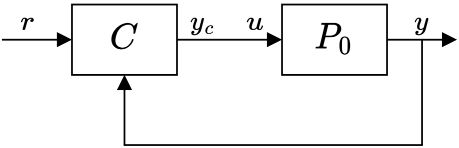

Figure 1: The closed-loop system Σ_0.

- Consider a **nominal LTI plant** P_0, where $u \in \mathbb{R}^m$ is the plant input, $y \in \mathbb{R}^p$ is the plant output and $m > p$.
- Consider a **controller** C, where $r \in \mathbb{R}^p$ is a constant reference signal and $y_c \in \mathbb{R}^m$ is the controller output.

Assumption

The closed-loop system Σ_0 is well-posed and asymptotically stable.

Input allocation

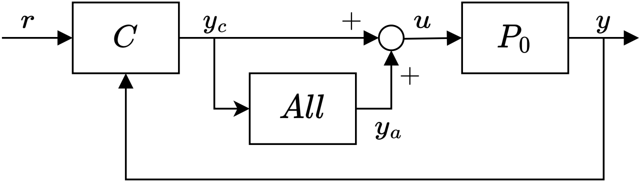

Figure 2: The allocated closed-loop system $\Sigma_{0,all}$.

Input allocation

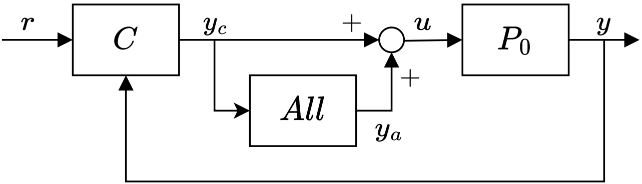

Figure 2: The allocated closed-loop system $\Sigma_{0,all}$.

- Consider an **Allocator** All, described by

$$\dot{x}_a = f_a(x_a, y_c), \quad (1a)$$

$$y_a = g_a(x_a, y_c), \quad (1b)$$

where $y_a \in \mathbb{R}^m$ is the allocator output and

$$u = y_c + y_a. \quad (2)$$

Input allocation problem

Consider the nominal closed-loop system Σ_0 and design an input **allocator** such that the allocated closed-loop system $\Sigma_{0,all}$:

(AS) is **well-posed** and **asymptotically stable**.

(I) ensures **output invisibility**.

(O) ensures **steady-state optimality**, namely, the steady-state plant input $u_\infty = y_{c,\infty} + y_{a,\infty}$ solves

$$J(u_\infty) = \min_{v \in U_{\Sigma_0}} J(v). \tag{3}$$

Allocator Design

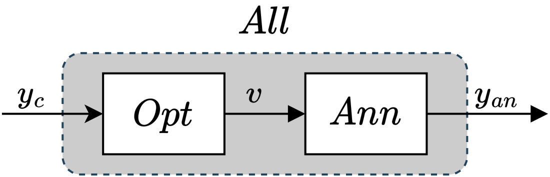

Figure 3: The internal structure of the *Allocator*

The **allocator** can be designed as the cascade of an **optimizer** and an **annihilator**.

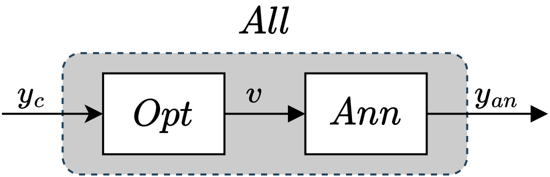

Figure 3: The internal structure of the *Allocator*

The **allocator** can be designed as the cascade of an **optimizer** and an **annihilator**.

- The **optimizer** ensures **steady-state optimality**.

Allocator Design

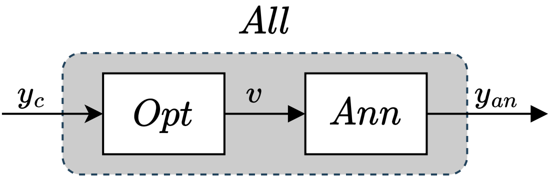

Figure 3: The internal structure of the *Allocator*

The **allocator** can be designed as the cascade of an **optimizer** and an **annihilator**.

- The **optimizer** ensures **steady-state optimality**.
- The **annihilator** ensures **output invisibility**.

Allocator Design

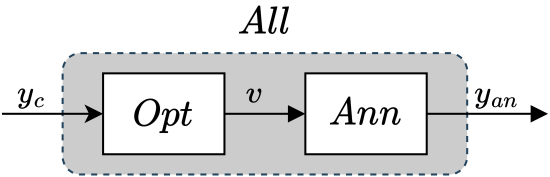

Figure 3: The internal structure of the *Allocator*

The **allocator** can be designed as the cascade of an **optimizer** and an **annihilator**.

- The **optimizer** ensures **steady-state optimality**.
- The **annihilator** ensures **output invisibility**.

We introduce a novel design based on **polynomial factorization**.

- Compute a **left factorization** of the plant transfer function

$$W_0(s) = C_0(sI - A_0)^{-1}B_0 + D_0 = D^{-1}(s)N(s). \tag{4}$$

- Compute a **left factorization** of the plant transfer function

$$W_0(s) = C_0(sI - A_0)^{-1}B_0 + D_0 = D^{-1}(s)N(s). \tag{4}$$

- By using the **adjoin method** to compute $(sI - A_0)^{-1}$, with

$$\text{det}(sI - A_0) \triangleq s^n + a_{n-1}s^{n-1} + ... + a_1 s + a_0, \tag{5a}$$

$$\text{adj}(sI - A_0) \triangleq E_{n-1}s^{n-1} + ... + E_1 s + E_0, \tag{5b}$$

the following relationships are obtained

$$D(s) = \sum_{k=0}^{n} a_k s^k \, I, \quad N(s) = \sum_{k=0}^{n} (C_0 E_k B_0 + a_k D_0) s^k. \tag{6}$$

- Compute a **left factorization** of the plant transfer function

$$W_0(s) = C_0(sI - A_0)^{-1}B_0 + D_0 = D^{-1}(s)N(s). \tag{4}$$

- By using the **adjoin method** to compute $(sI - A_0)^{-1}$, with

$$\mathrm{det}(sI - A_0) \triangleq s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0, \tag{5a}$$

$$\mathrm{adj}(sI - A_0) \triangleq E_{n-1}s^{n-1} + \ldots + E_1 s + E_0, \tag{5b}$$

the following relationships are obtained

$$D(s) = \sum_{k=0}^{n} a_k s^k \, I, \quad N(s) = \sum_{k=0}^{n} (C_0 E_k B_0 + a_k D_0) s^k. \tag{6}$$

- The **Souriau-Leverrier-Faddeev algorithm** yields

$$E_{n-k} = a_{n-k+1} I_n + A_0 E_{n-k+1}, \qquad k = 1, \ldots n+1, \tag{7a}$$

$$a_{n-k} = -\frac{1}{k}\,\mathrm{tr}(A_0 E_{n-k}), \qquad k = 1, \ldots n. \tag{7b}$$

Allocator Design
Annihilator

- Define the **annihilator** transfer function $W_{an}(s) = N^{\perp}(s)\Psi^{-1}(s)$:

$$W_0(s)\, W_{an}(s) = 0 \quad \rightarrow \quad D^{-1}(s)N(s)N^{\perp}(s)\Psi^{-1}(s) = 0. \qquad (8)$$

- Define the **annihilator** transfer function $W_{an}(s) = N^{\perp}(s)\Psi^{-1}(s)$:

$$W_0(s)\, W_{an}(s) = 0 \quad \rightarrow \quad D^{-1}(s)N(s)N^{\perp}(s)\Psi^{-1}(s) = 0. \tag{8}$$

- The following **necessary and sufficient condition** must hold true

$$N(s)N^{\perp}(s) = (N_n s^n + ... + N_0)(N_\eta^{\perp} s^\eta + ... + N_0^{\perp}) = 0. \tag{9}$$

Allocator Design

Annihilator

- Define the **annihilator** transfer function $W_{an}(s) = N^{\perp}(s)\Psi^{-1}(s)$:

$$W_0(s)\,W_{an}(s) = 0 \quad \rightarrow \quad D^{-1}(s)N(s)N^{\perp}(s)\Psi^{-1}(s) = 0. \tag{8}$$

- The following **necessary and sufficient condition** must hold true

$$N(s)N^{\perp}(s) = (N_n s^n + ... + N_0)(N_{\eta}^{\perp} s^{\eta} + ... + N_0^{\perp}) = 0. \tag{9}$$

- By expanding and rearranging the terms in a matrix form, one has that

$$\begin{bmatrix} N_0 & 0 & \dots & 0 \\ N_1 & N_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ N_n & N_{n-1} & \cdots & N_0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & N_n & N_{n-1} \\ 0 & \dots & 0 & N_n \end{bmatrix} \begin{bmatrix} N_0^{\perp} \\ N_1^{\perp} \\ \vdots \\ N_{\eta-1}^{\perp} \\ N_{\eta}^{\perp} \end{bmatrix} = \bar{N}\bar{N}^{\perp} = 0. \tag{10}$$

- The **annihilator numerator** $N^{\perp}(s)$ is obtained by computing the null space of the band matrix $\bar{N}$.

- The **annihilator numerator** $N^{\perp}(s)$ is obtained by computing the null space of the band matrix $\bar{N}$.
- The **annihilator denominator** $\Psi(s)$ is chosen as $\Psi(s) = \psi(s)I$, where $\psi(s)$ is any Hurwitz polynomial with $\deg(\psi(s)) \geq \eta$, in order to have a **stable** and **realizable** $W_{an}(s)$.

- The **annihilator numerator** $N^{\perp}(s)$ is obtained by computing the null space of the band matrix $\bar{N}$.
- The **annihilator denominator** $\Psi(s)$ is chosen as $\Psi(s) = \psi(s)I$, where $\psi(s)$ is any Hurwitz polynomial with $\deg(\psi(s)) \geq \eta$, in order to have a **stable** and **realizable** $W_{an}(s)$.

$\Psi(s)$ is a degree of freedom that can be exploited to build the **allocator**.

- Choose the **cost function** to be minimized at steady-state $J(u_\infty)$.

- Choose the **cost function** to be minimized at steady-state $J(u_\infty)$.
- The **optimizer** is defined by

$$\dot{x}_{op} = -\Gamma \nabla J(y_c + \Omega_{an} x_{op}), \tag{11a}$$

$$v = x_{op}, \tag{11b}$$

where $\Omega_{an} = W_{an}(0)$ and $\Gamma > 0$ regulates the convergence rate.

- Choose the **cost function** to be minimized at steady-state $J(u_\infty)$.
- The **optimizer** is defined by

$$\dot{x}_{op} = -\Gamma \nabla J(y_c + \Omega_{an} x_{op}), \tag{11a}$$
$$v = x_{op}, \tag{11b}$$

where $\Omega_{an} = W_{an}(0)$ and $\Gamma > 0$ regulates the convergence rate.

Γ is a degree of freedom that can be exploited to build the **allocator**.

Roadmap

An **allocator** designed on the nominal plant P_0 successfully **solve the nominal allocation problem**.

An **allocator** designed on the nominal plant P_0 successfully **solve the nominal allocation problem**.

Problem

In the case of **plant uncertainties**, $P \neq P_0$, the **output invisibility** and **steady-state optimality** properties are lost.

An **allocator** designed on the nominal plant P_0 successfully **solve the nominal allocation problem**.

Problem

In the case of **plant uncertainties**, $P \neq P_0$, the **output invisibility** and **steady-state optimality** properties are lost.

Approach

Exploit the **allocator's degrees of freedom** to robustify the allocator with respect to plant uncertainties.

Robust input allocation

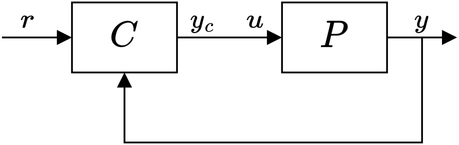

Figure 4: The closed-loop system Σ.

- Consider a **finite set of LTI systems** $\mathcal{P}$, with cardinality $|\mathcal{P}| = N$, one of which is considered to be the **nominal plant** P_0.

Robust input allocation

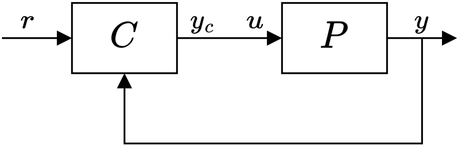

Figure 4: The closed-loop system Σ.

- Consider a **finite set of LTI systems** $\mathcal{P}$, with cardinality $|\mathcal{P}| = N$, one of which is considered to be the **nominal plant** P_0.
- Consider a **robust controller** C for the set $\mathcal{P}$.

Robust input allocation

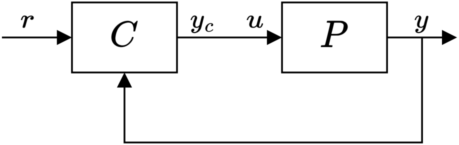

Figure 4: The closed-loop system Σ.

- Consider a **finite set of LTI systems** $\mathcal{P}$, with cardinality $|\mathcal{P}| = N$, one of which is considered to be the **nominal plant** P_0.
- Consider a **robust controller** C for the set $\mathcal{P}$.

Assumption

The closed-loop system Σ is **well-posed** and **asymptotically stable** for each $P \in \mathcal{P}$.

Robust input allocation problem

Consider the closed-loop system Σ and design an input **allocator** such that for each $P \in \mathcal{P}$ the allocated closed-loop system Σ_{all}:

- **(AS)** is **well-posed** and **asymptotically stable**.
- **(NI)** ensures **output invisibility** in the nominal case.
- **(NO)** ensures **steady-state optimality** in the nominal case.
- **(RIO)** **minimizes** the cost function

$$\tilde{J}(\Theta) = \sum_{i=1}^{N} \left(\int_0^T \|\delta y_i(t)\|^2 + \alpha J(u_{all,i}(t)) \mathrm{d}t \right), \tag{12}$$

where Θ is the set of the **allocator's design parameters** and the subscript i refers to the trajectories yield by the i-th plant in $\mathcal{P}$.

1. Design the **annihilator** following the previous algorithm.

1. Design the **annihilator** following the previous algorithm.
2. Design the steady-state **optimizer** according to the cost function to be minimized.

1. Design the **annihilator** following the previous algorithm.
2. Design the steady-state **optimizer** according to the cost function to be minimized.
3. Solve the following **optimization problem**, using $\Theta = (\Gamma, \psi(s))$ previously chosen as the initial condition of the numerical solver.

$$\begin{aligned} \min_{\Gamma,\psi} \quad & \sum_{i=1}^{N} \left(\int_0^T \|\delta y_i(t)\|^2 + \alpha J(u_i(t)) \mathrm{d}t \right) \\ \text{s.t.} \quad & \Gamma = \mathsf{diag}(\gamma_1, \ldots \gamma_{m-p}) > 0, \\ & \psi(s) \text{ is Hurwitz with } \mathsf{deg}(\psi(s)) = n + 1. \end{aligned} \tag{13}$$

Roadmap

- The **nominal plant** P_0 is described by the matrices

$$A_0 = \begin{bmatrix} -0.157 & 0.094 \\ -0.416 & 0.45 \end{bmatrix}, \qquad B_0 = \begin{bmatrix} 0.87 & 0.253 & 0.743 \\ 0.39 & 0.354 & 0.65 \end{bmatrix}, \tag{14a}$$

$$C_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \qquad D_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}. \tag{14b}$$

- The **nominal plant** P_0 is described by the matrices

$$A_0 = \begin{bmatrix} -0.157 & 0.094 \\ -0.416 & 0.45 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.87 & 0.253 & 0.743 \\ 0.39 & 0.354 & 0.65 \end{bmatrix}, \tag{14a}$$

$$C_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}. \tag{14b}$$

- The **controller** C is described by the matrices

$$A_c = \begin{bmatrix} -1.57 & 0.5767 & 0.822 & -0.65 \\ -0.9 & -0.501 & -0.94 & 0.802 \\ 0 & 1 & -1.61 & 1.614 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{15a}$$

$$C_c = \begin{bmatrix} 1.81 & -1.2 & -0.46 & 0 \\ -0.62 & 1.47 & 0.89 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D_r = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{15b}$$

- The **annihilator** is calculated on the nominal plant P_0 by choosing for the denominator an initial Hurwitz polynomial $\psi(s) = (s+1)(s+2)(s+3)$.

- The **annihilator** is calculated on the nominal plant P_0 by choosing for the denominator an initial Hurwitz polynomial $\psi(s) = (s+1)(s+2)(s+3)$.
- The **optimizer** is defined according to the cost function to be minimized, with an initial gain matrix $\Gamma = I$.

- The **annihilator** is calculated on the nominal plant P_0 by choosing for the denominator an initial Hurwitz polynomial $\psi(s) = (s+1)(s+2)(s+3)$.
- The **optimizer** is defined according to the cost function to be minimized, with an initial gain matrix $\Gamma = I$.
- **Set of plants** $\mathcal{P}$, with $|\mathcal{P}| = N = 10$, where each $P \in \mathcal{P}, P \neq P_0$ is obtained by perturbing all elements of P_0 by 10% randomly.

- The **annihilator** is calculated on the nominal plant P_0 by choosing for the denominator an initial Hurwitz polynomial $\psi(s) = (s+1)(s+2)(s+3)$.
- The **optimizer** is defined according to the cost function to be minimized, with an initial gain matrix $\Gamma = I$.
- **Set of plants** $\mathcal{P}$, with $|\mathcal{P}| = N = 10$, where each $P \in \mathcal{P}, P \neq P_0$ is obtained by perturbing all elements of P_0 by 10% randomly.
- The **robustification process** is carried out by solving the optimization problem on the set of plants $\mathcal{P}$.

Numerical simulations
Linear allocation

Goal

Minimize the Euclidean norm of the steady-state plant input.

Goal

Minimize the Euclidean norm of the steady-state plant input.

- **Cost function** to be minimized

$$J(u_\infty) = \frac{1}{2}\|u_\infty\|^2. \tag{16}$$

Goal

Minimize the Euclidean norm of the steady-state plant input.

- **Cost function** to be minimized

$$J(u_\infty) = \frac{1}{2}\|u_\infty\|^2. \tag{16}$$

- A quadratic cost function yields a **linear steady-state optimizer**

$$\dot{x}_{op} = -\Gamma\Omega_{an}^{\mathrm{T}}\Omega_{an}x_{op} - \Gamma\Omega_{an}^{\mathrm{T}}y_c, \tag{17a}$$
$$v = x_{op}. \tag{17b}$$

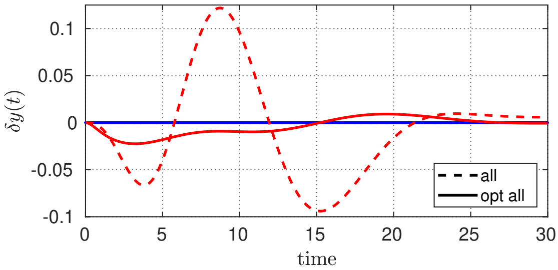

Figure 5: Output variation with a non-optimized *Allocator* (dashed), and with the optimized *Allocator* (solid); blue signals refer to the nominal plant and red ones to a perturbed plant.

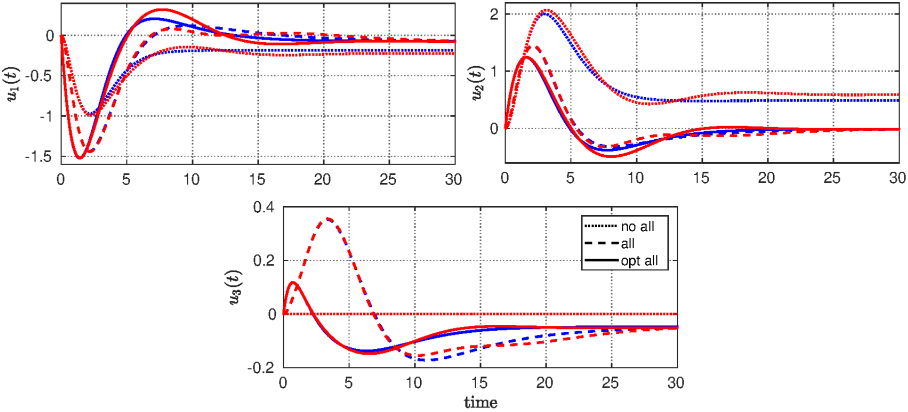

Figure 6: Plant inputs without the *Allocator* (dotted), with a non-optimized *Allocator* (dashed), and with the optimized *Allocator* (solid); blue signals refer to the nominal plant, red ones to a perturbed plant.

Numerical simulations

Nonlinear allocation

Goal

Keeping the steady-state plant input away from a saturation region.

Goal

Keeping the steady-state plant input **away from a saturation region**.

- **Cost function** to be minimized

$$J(u_\infty) = \frac{1}{2}\|\mathsf{dz}(u_\infty)\|^2, \tag{18}$$

where $\mathsf{dz}(u_\infty) = \mathsf{sign}(u_\infty)\mathsf{max}\{0, |u_\infty| - \bar{u}\}$.

Goal

Keeping the steady-state plant input **away from a saturation region**.

- **Cost function** to be minimized

$$J(u_\infty) = \frac{1}{2}\|\mathsf{dz}(u_\infty)\|^2, \tag{18}$$

where $\mathsf{dz}(u_\infty) = \mathsf{sign}(u_\infty)\mathsf{max}\{0, |u_\infty| - \bar{u}\}$.

- A nonlinear cost function yields a **nonlinear steady-state optimizer**

$$\dot{x}_{op} = -\Gamma\Omega_{an}^{\mathrm{T}}\mathsf{dz}(y_c + \Omega_{an}x_{op}), \tag{19a}$$

$$v = x_{op}. \tag{19b}$$

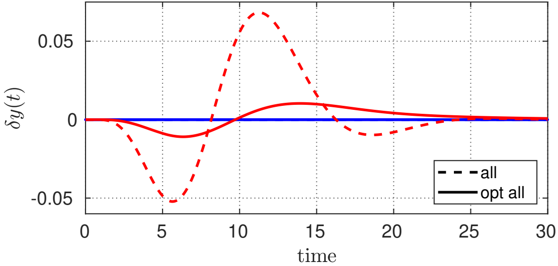

Figure 7: Output variation with a non-optimized *Allocator* (dashed), and with the optimized *Allocator* (solid); blue signals refer to the nominal plant and red ones to a perturbed plant.

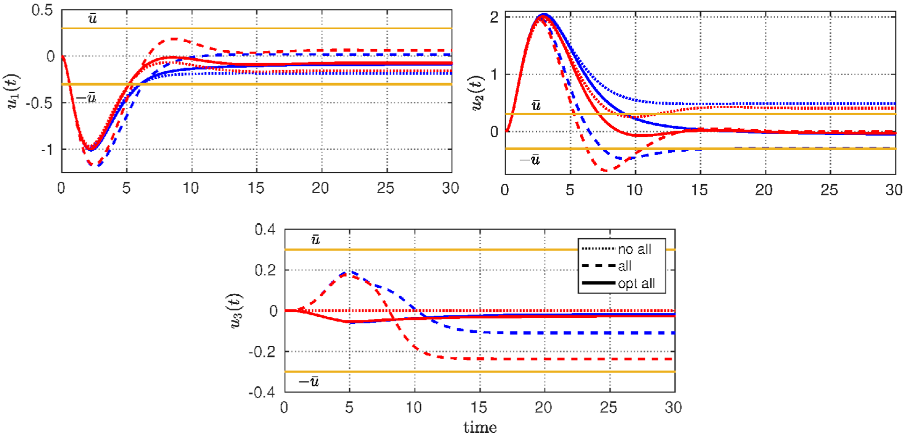

Figure 8: Plant inputs without the *Allocator* (dotted), with a non-optimized *Allocator* (dashed), and with the optimized *Allocator* (solid); blue signals refer to the nominal plant, red ones to a perturbed plant, and in yellow the region bounds.

Roadmap

Conclusions

Topic

The **dynamic input allocation** problem has been addressed in the context of **plant uncertainties**.

Conclusions

Topic

The **dynamic input allocation** problem has been addressed in the context of **plant uncertainties**.

Novel contributions

- Provide an algorithm to **design** an **allocator**.
- Provide an algorithm for **robustify** and **allocator** in the presence of uncertainties in the plant.

Conclusions

Topic

The **dynamic input allocation** problem has been addressed in the context of **plant uncertainties**.

Novel contributions

- Provide an algorithm to **design** an **allocator**.
- Provide an algorithm for **robustify** and **allocator** in the presence of uncertainties in the plant.

Future directions

- Specific approaches for structured **plant uncertainties**.
- Design for **nonlinear plants**.