ULB

Reference Governors *for* *Systems subject to constraints*

Emanuele Garone

Reference

REFERENCE for Reference Governor

Survey Paper: Garone, Di Cairano, Kolmanovsky, “Reference and Command Governors for Systems with Constraints: A Survey on Theory and Applications“, *Automatica*, 2017.

Automatica 75 (2017) 306–328

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

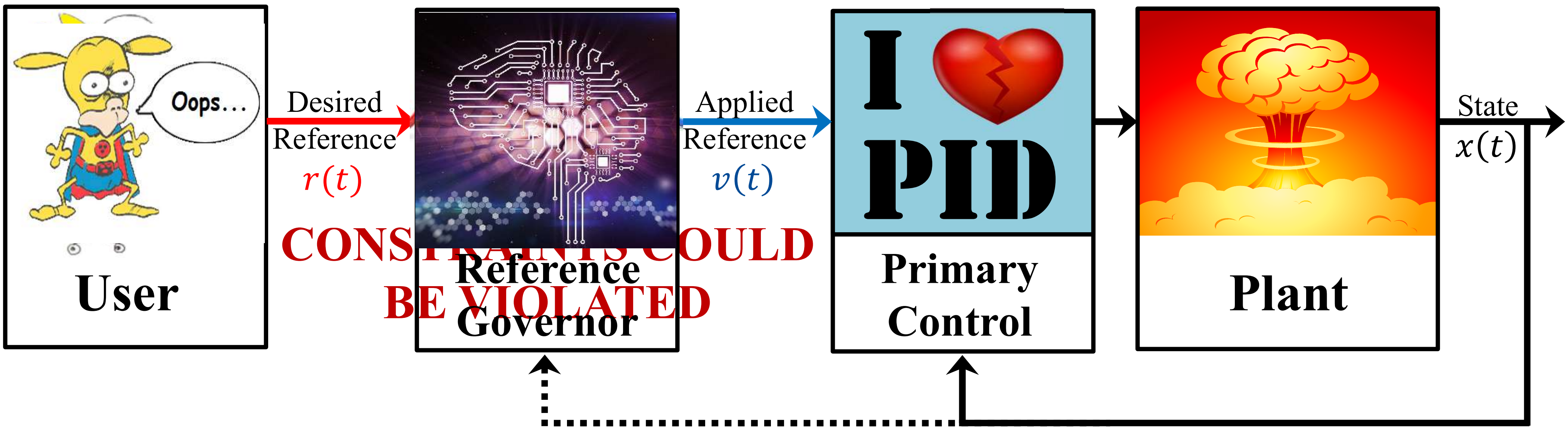
Survey paper

Reference and command governors for systems with constraints: A survey on theory and applications

Emanuele Garone [a], Stefano Di Cairano [b], Ilya Kolmanovsky [c]

[a] École Polytechnique de Bruxelles, Université Libre de Bruxelles (ULB), Belgium
[b] Mitsubishi Electric Research Laboratories, Cambridge, MA, United States
[c] Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI, United States

MOTIVATION



RG OBJECTIVES:

Generate an **applied reference $v(t)$** such that

- Constrains are **not violated**
- $v(t)$ tracks r(t) **“as close as possible”**

ULB

Problem

PRE-COMPENSATED SYSTEM

$$\boldsymbol{x}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{f}(\boldsymbol{x}(\boldsymbol{t}), \boldsymbol{v}(\boldsymbol{t}))$$

- For any constant reference $\boldsymbol{v}$, the equilibrium point $\boldsymbol{x}_{\boldsymbol{v}}$ is **Asymptotically Stable**

CONSTRAINTS

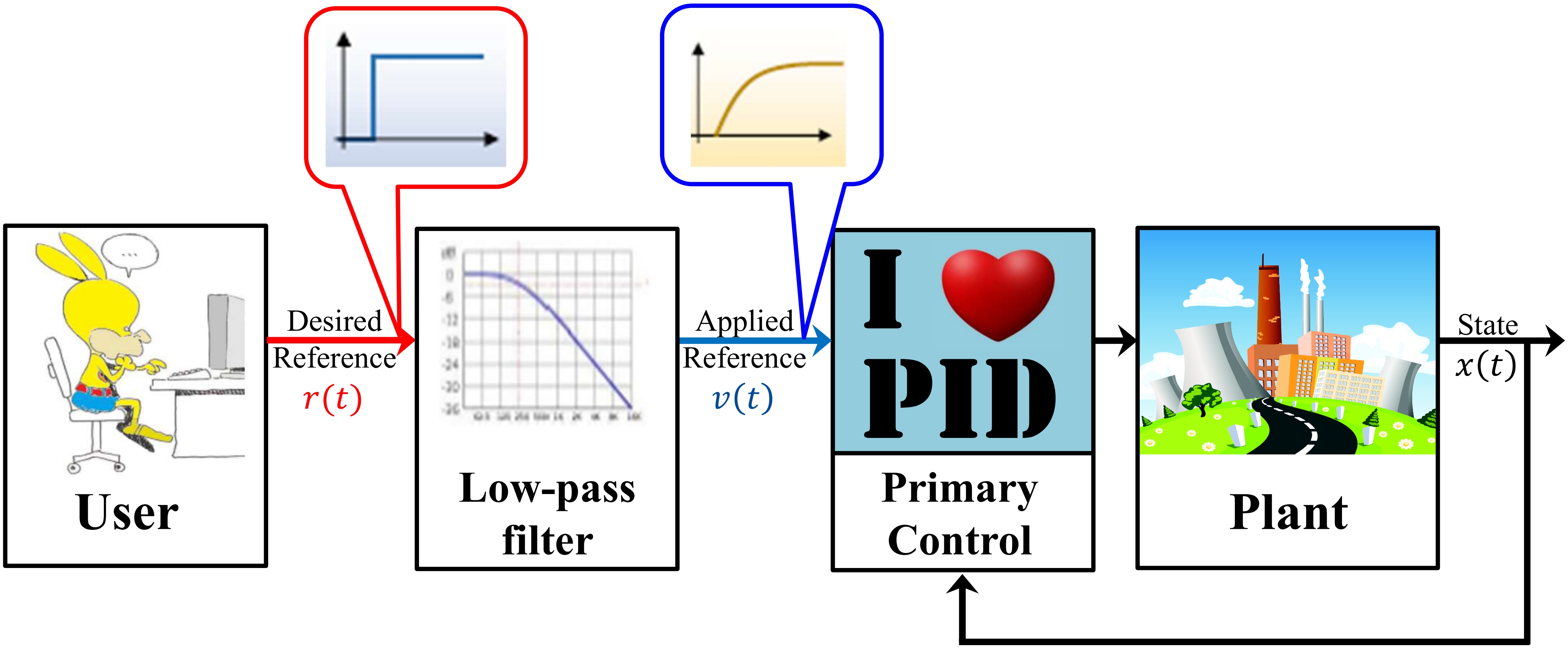
$$(\boldsymbol{x}(\boldsymbol{t}), \boldsymbol{v}(\boldsymbol{t})) \in \boldsymbol{C}$$

RG OBJECTIVES:

Given r(t), generate an **applied reference $\boldsymbol{v}(\boldsymbol{t})$** such that:

- **Constraint** are satisfied $\big(\boldsymbol{x}(\boldsymbol{t}), \boldsymbol{v}(\boldsymbol{t})\big) \in \boldsymbol{C}, \quad \forall \boldsymbol{t} > \mathbf{0}$
- **The desired reference** is tracked**:** $\boldsymbol{v}(\boldsymbol{t}) \approx \boldsymbol{r}(\boldsymbol{t})$

An old idea

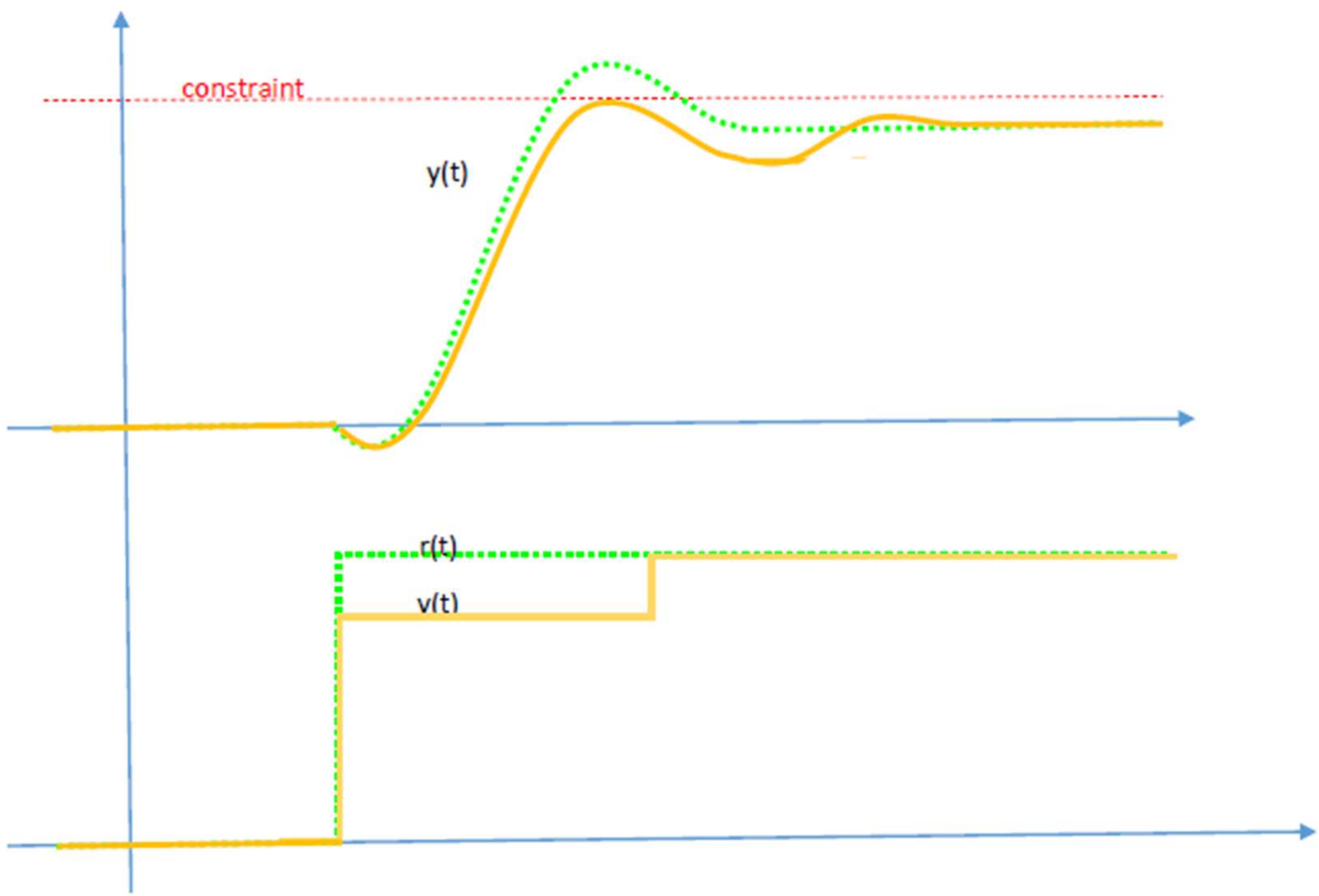


REFERENCE MANAGEMENT = LOW PASS FILTERING 2.0

Basic Idea

IDEA

compute $\boldsymbol{v}(\boldsymbol{t})$ so that **if constantly applied** it would not violate constraints



Maximum Constraint Admissible Set

IDEA

We need to select $\boldsymbol{v}$ in the set of initial conditions $\boldsymbol{x}(0)$ and constantly applied references $\boldsymbol{v}$ so that constraints are always satisfied

Notation - Linear

SYSTEM:

$$\boldsymbol{x}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{A}\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{B}\boldsymbol{v}(\boldsymbol{t})$$

- $\boldsymbol{A}$ is *Hurwitz*
- Equilibria $\boldsymbol{x}_{\boldsymbol{v}} = (\boldsymbol{I}-\boldsymbol{A})^{-\mathbf{1}}\boldsymbol{B}\boldsymbol{v}$
- Prediction for constant v: $\widehat{\boldsymbol{x}}(\boldsymbol{k}|\boldsymbol{x},\boldsymbol{v}) = \boldsymbol{x}_{\boldsymbol{v}} + \boldsymbol{A}^{\boldsymbol{k}}\boldsymbol{x} - (\boldsymbol{I}-\boldsymbol{A})^{-\mathbf{1}}\boldsymbol{A}^{\boldsymbol{k}}\boldsymbol{B}\boldsymbol{v}$

CONSTRAINTS:

$$\big(\boldsymbol{x}(\boldsymbol{t}),\boldsymbol{v}(\boldsymbol{t})\big) \in \boldsymbol{C}$$

The set O_∞

Maximum Constraint Admissible Set

The **maximum constraint admissible set** is the set

$$O_\infty = \{(x,v)|\ (\hat{x}(k|x,v),v) \in C, k = 0,1,\dots\}$$

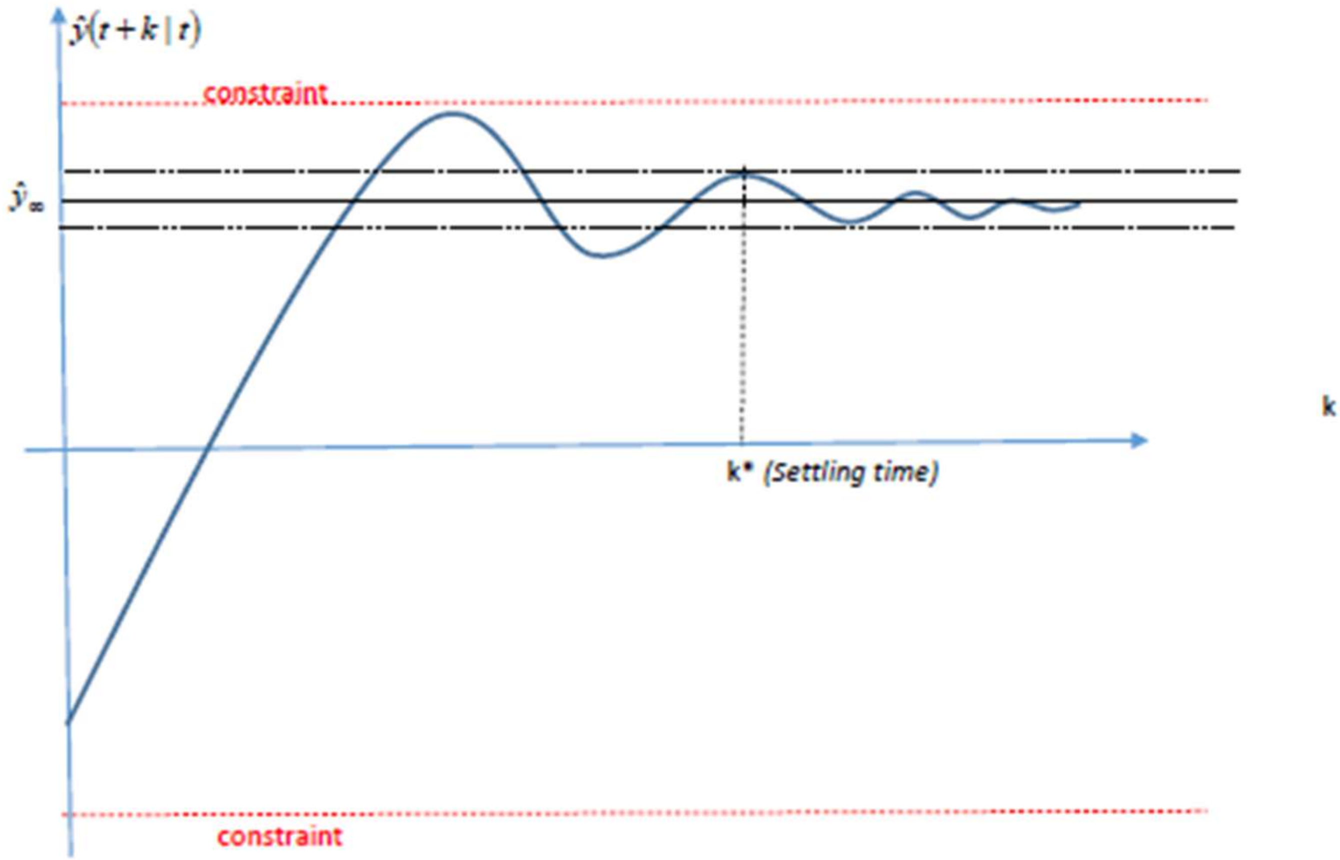
PROPERTIES:

- The set O_∞ is **positive invariant**, i.e. if $(x(t),v) \in O_\infty$ and $x(t+1) = Ax(t) + Bv$ then, $(x(t+1),v) \in O_\infty$
- If C is convex [polyhedral] The set O_∞ is convex [polyhedral]

The number of constraints that define O_∞ is infinite !

Approximating O_∞

IDEA: Find a suitable inner approximation $P \subset O_\infty$



The set $\tilde{O}_\infty$

STEP 1: Define the set of steady-state admissible references

$$R_C = \{v | (x_v, v) \in C\}$$

STEP 2: Define an arbitrary small inner approximation of $R_C, e.g.$

$$R_\varepsilon = R_C \sim Ball(\varepsilon)$$

STEP 3: Define

$$O_\varepsilon = \{(x, v) | v \in R_\varepsilon\}$$

STEP 4: Define

$$\tilde{O}_\infty = O_\infty \cap O_\varepsilon$$

The set $\tilde{O}_\infty$

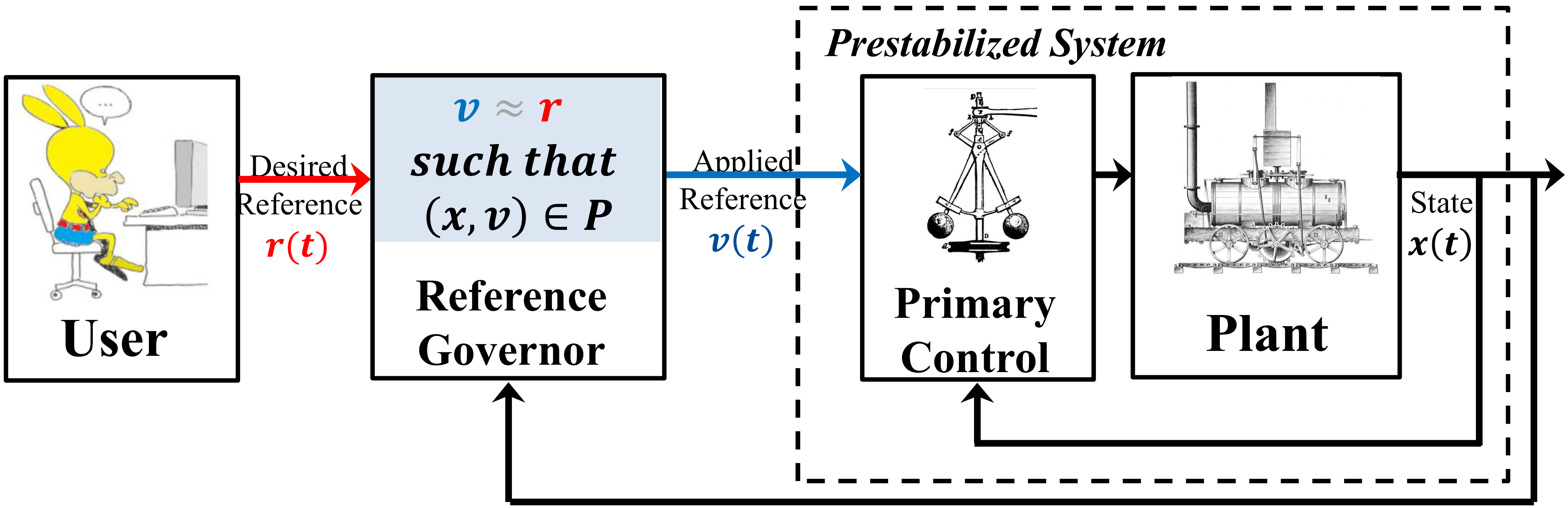
PROPERTIES:

- If O_∞ is a compact set then $\tilde{O}_\infty$ is determined by a finite number of constraints
- Let $\tilde{O}_k = \{(x,v) | \hat{x}(i|x,v) \in C, i = 0, \dots, k\} \cap O^\varepsilon$ then $\tilde{O}_\infty = \tilde{O}_{k^*}$ where k^* is the smallest integer such that $\tilde{O}_{k^*} = \tilde{O}_{k^*+1}$
- $\tilde{O}_\infty$ is positively invariant

RECAP:

- Given a state x(t), all the feasible v are the one so that $(x(t), v) \in O_\infty$
- O_∞ consists of an infinite amount of constraints
- Approximations $P \subseteq O_\infty$ must be used
- One of the most interesting approximations is $\tilde{O}_\infty$

Reference Management

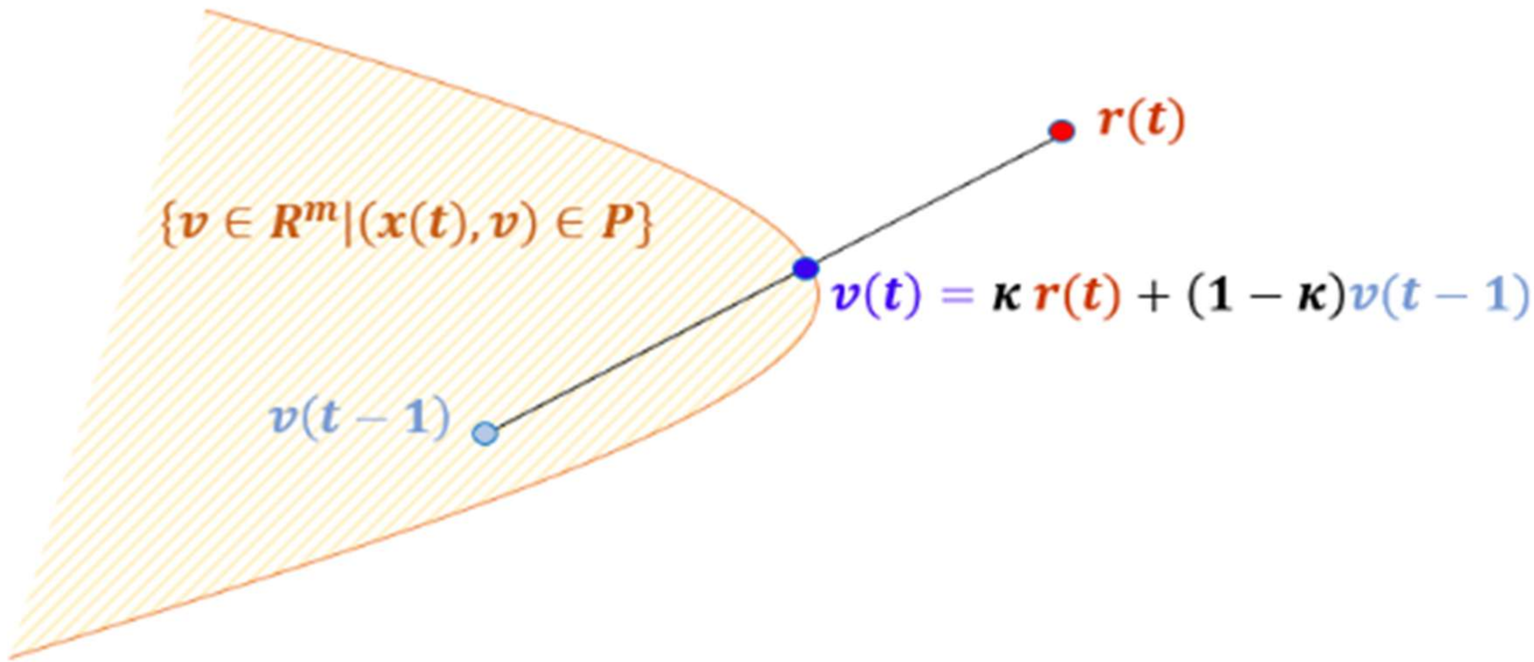


Two basic approaches:

- Scalar Reference Governor (Gilbert, Kolmanovsky, 1994)
- Command Governors (Bemporad, Casavola, Mosca, 1997)

Scalar RG

IDEA: If $v(t-1)$ was feasible, find the best approximation of $r(t)$ in the set P along the line segment between $v(t-1)$ and $r(t)$



Scalar RG

This can be formalized as

1. Solve

$$\max_{\kappa\in[0,1]} \kappa$$

subject to

$$\big(x(t), (1-\kappa)v(t-1)+\kappa\, r(t)\big) \in P$$

2. Set $v(t) = (1-\kappa)v(t-1)+\kappa\, r(t)$

Which is a single variable optimization that can be solved very efficiently !

Scalar RG

THEOREM: If the set P is

A1 – positive invariant

A2 – so that the set $R_P = \{v | (x_v, v) \in P\}$ is convex

A3 – There exists a scalar $\varepsilon > 0$ such that $(x_v + z, v) \in P, \forall z: ||z|| < \varepsilon$

and if a $v(0)$ such that $(\mathrm{x}(0), v(0)) \in P$ is **known,**

Then:

- The Scalar RG ensure **recursive feasibility**
- For a constant $r(t) = r$, $v(t)$ **converges in finite time** to
  - r if feasible
  - to a suitable approximation of r otherwise

A1 can be relaxed

Relaxing A1

1. Solve

$$\max_{\kappa \in [0,1]} \kappa$$

subject to

$$\big(x(t), (1-\kappa)v(t-1) + \kappa\, r(t)\big) \in \ P$$

2. If a solution exists

$$Set \ \ v(t) = (1-\kappa)v(t-1) + \kappa\, r(t)$$

else

$$Set \ \ v(t) = v(t-1)$$

This allows to relax the need for an invariant P

Scalar RG

THEOREM: If the set P is

A2 – so that the set $R_P = \{v | (x_v, v) \in P\}$ is convex

A3 – There exists a scalar $\varepsilon > 0$ such that $(x_v + z, v) \in P, \forall z\text{:} \left\|z\right\| < \varepsilon$

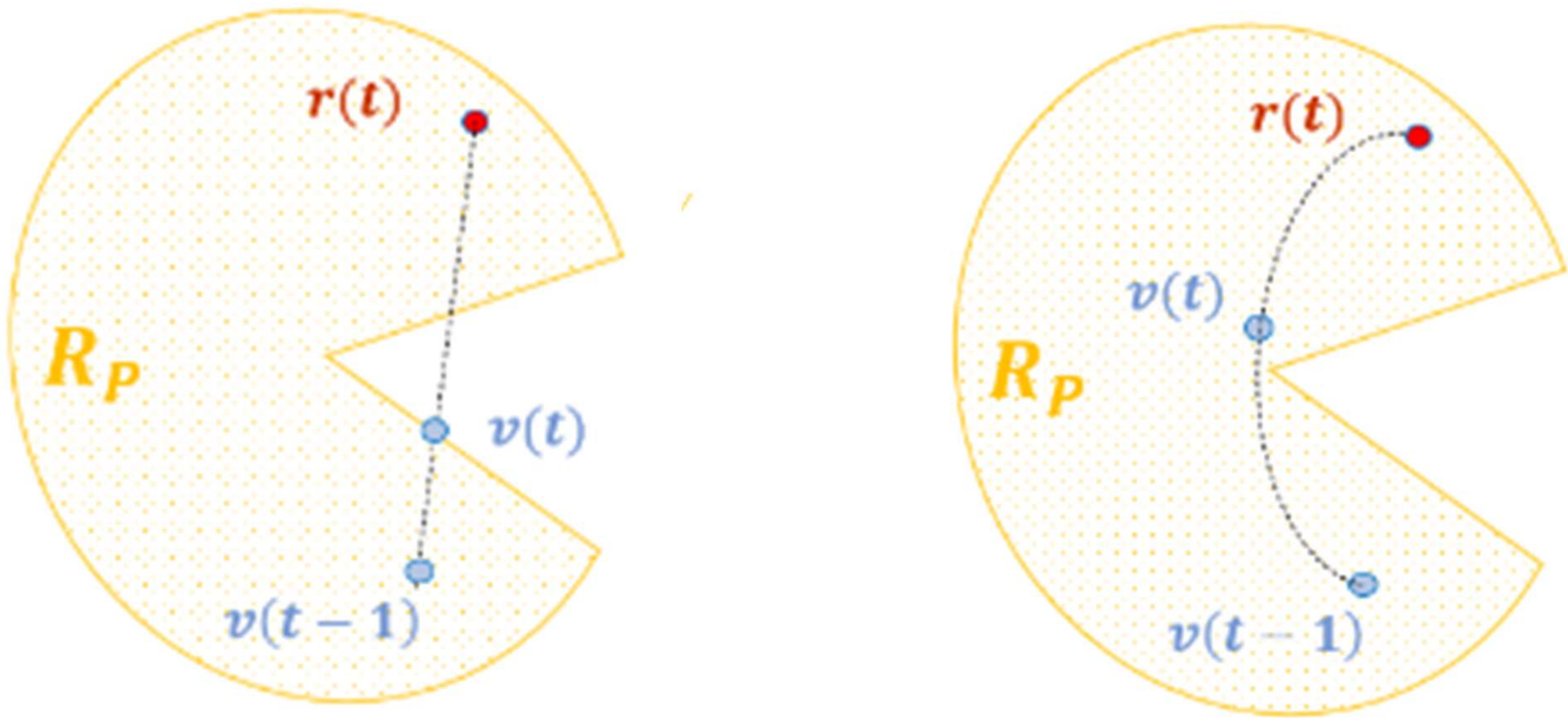
and if a $v(0)$ such that $(\mathrm{x}(0), v(0)) \in P$ is **known**,

Then:

- The Modified Scalar RG ensure **recursive feasibility**
- For a constant $r(t) = r$, $v(t)$ **converges in finite time** to
  - r if feasible
  - to a suitable approximation of r otherwise

Also A2 can be relaxed

Relaxing A2



If paths of steady-state admissible references are known, they can be used to perform a linear search along this path

Relaxing A2

ULB

Nonlinear MPC for Tracking for a Class of Non-Convex Admissible Output Sets

Andres Cotorruelo, Daniel R. Ramirez, Daniel Limon, Emanuele Garone

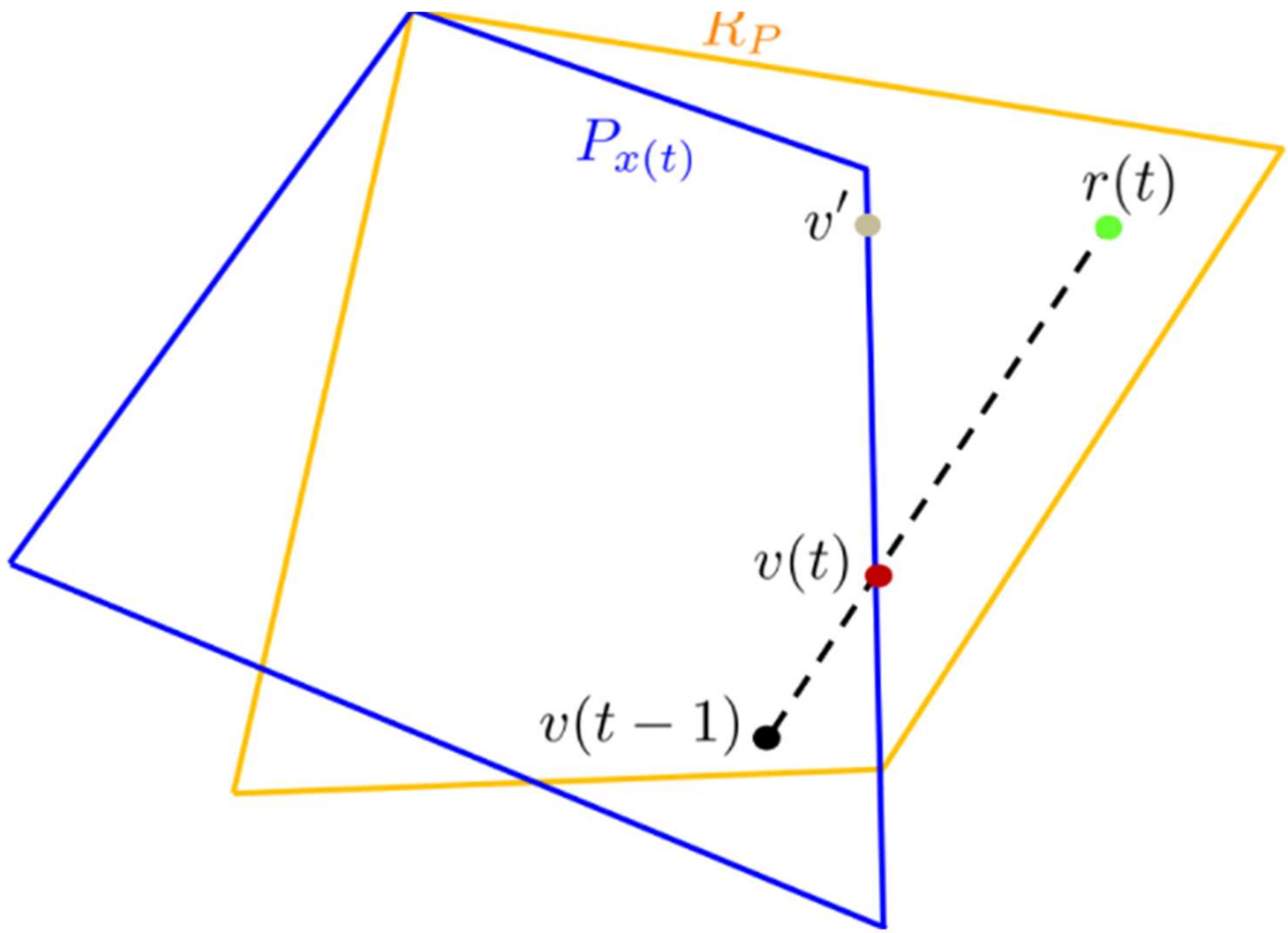
***Abstract*—This paper presents an extension to the nonlinear Model Predictive Control for Tracking scheme able to guarantee convergence even in cases of non-convex output admissible sets. This is achieved by incorporating a convexifying homeomorphism in the optimization problem, allowing it to be solved in the convex space. A novel class of non-convex sets is also defined for which a systematic procedure to construct a convexifying homeomorphism is provided. This homeomorphism is then embedded in the Model Predictive Control optimization problem in such a way that the homeomorphism is no longer required in closed form. Finally, the effectiveness of the proposed method is showcased through an illustrative example.**

presence of non-convex admissible output sets, this formulation might present convergence issues.

Although this limitation is not very stringent for some classical applications (*e.g.* in process control), there are several cases in which state constraints are non-convex, such as mobile robot navigation [14], formation flight control [15], aerospace problems like rendezvous, orbital transfer, optimal launch [16], or soft landing maneuvers [17]. In the tracking scheme of [13], the way to deal with non-convex constraints is to restrict the operation of the MPC to a convex subset of admissible outputs. Although this practice can work for some applications, it introduces a relevant amount of conservativity

Command Governor

STARTING POINT: Searching along a line is often suboptimal



Command Governor

IDEA: Solve an optimization problem to find the best approximation of $r(t)$ in the set P

1. Solve

$$\min_{v} ||v - r(t)||^2$$

subject to

$$(x(t), v) \in \ P$$

2. Set $v(t) = v$

This is an actual optimization problem to be solved with a solver

CG properties

THEOREM: If the set P is

A1 – positive invariant

A2 – so that the set $R_P = \{v|(x_v, v) \in P\}$ is convex

A3 – There exists a scalar $\varepsilon > 0$ such that $(x_v + z, v) \in P, \forall z: ||z|| < \varepsilon$

and if a $v(0)$ such that $(\mathrm{x}(0), v(0)) \in P$ **exists**,

Then:

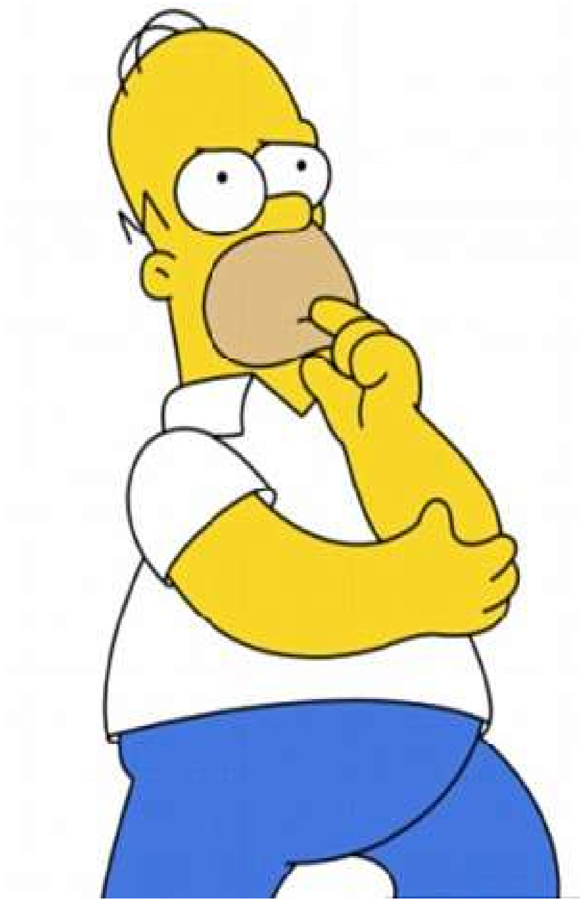
- The CG ensure **recursive feasibility**
- For a constant $r(t) = r$, $v(t)$ **converges in finite time** to the best feasible steady state approximation of r

A bit harder to relax assumptions

Reference and Command Governor

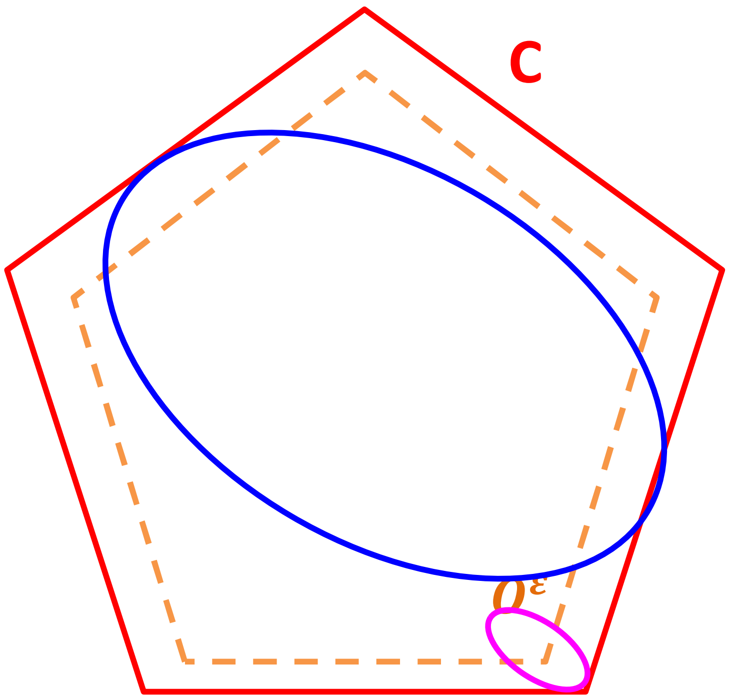
Implementation Aspects

OK now I want to use this thing so...



... how can I compute k^* and $\tilde{O}_\infty$?

Approach 1: Lyapunov



STEP 1: Consider a Lyapunov function

$$\mathrm{V}(x) = x^T Q x$$

where $\mathrm{A}^{\mathrm{T}} Q A - Q = -I$, and thus

$$\Delta \mathrm{V}(x) \leq -x^T x$$

STEP 2: Compute the optimal solution to

$$\gamma_M = \max_{\substack{v \in R^{\epsilon} \\ x:(x,v) \in C}} \mathrm{V}(x)$$

STEP 3: Compute the optimal solution to

$$\gamma_{\mathrm{m}} = \min_{\substack{v \in R^{\epsilon} \\ x:(x,v) \notin C}} \mathrm{V}(x)$$

STEP 4: Set $\bar{k} = \left\lceil \log_{(1-\lambda_M^{-1}\{Q\})} \frac{\gamma_m}{\gamma_M} \right\rceil$

STEP 5: $\tilde{O}_{\bar{\boldsymbol{k}}} = \tilde{O}_{\infty}$

Very conservative method

Approach 2: Redundancy

PROPERTIES:
- Let $\tilde{O}_k = \{(x,v)|\hat{x}(i|x,v) \in C, i = 0, \ldots, k\} \cap O^{\varepsilon}$ then $\tilde{O}_{\infty} = \tilde{O}_{k^*}$ where k^* is the smallest integer such that $\tilde{O}_{k^*} = \tilde{O}_{k^*+1}$

Notation:
- Let $C = \{(x,v): c(x,v) \geq 0\}$
- Let $h(x,v,k) = c(\hat{x}(k|x,v), v)$

Compare:

$$\tilde{O}_k = \{(x,v) : h(x,v,t) \geq 0, t = 0, \ldots, k\} \cap O_{\epsilon}$$

$$\tilde{O}_{k+1} = \{(x,v) : h(x,v,t) \geq 0, t = 0, \ldots, k+1\} \cap O_{\epsilon}$$

Approach 2: Redundancy

Conclusion:

$$\widetilde{O}_{k^*} = \widetilde{O}_{k^*+1} \quad \textbf{if and only if}$$

$h(x, v, k+1)$ are all redundant constraints w.r.t. $\widetilde{O}_{k^*}$

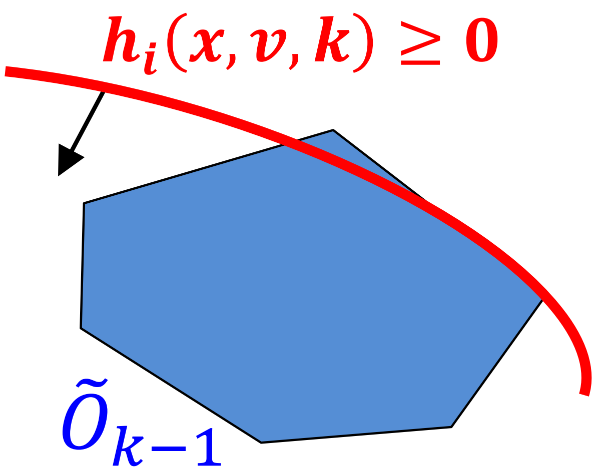
PROCEDURE to compute k^*:

increase k

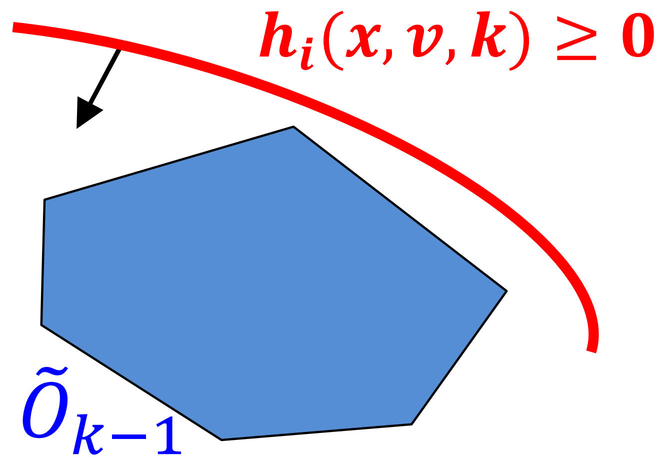
Until $h(x, v, k+1) \geq 0$ are redundant w.r.t $\widetilde{O}_k$

Approach 2: Redundancy

Non-redundant



Redundant



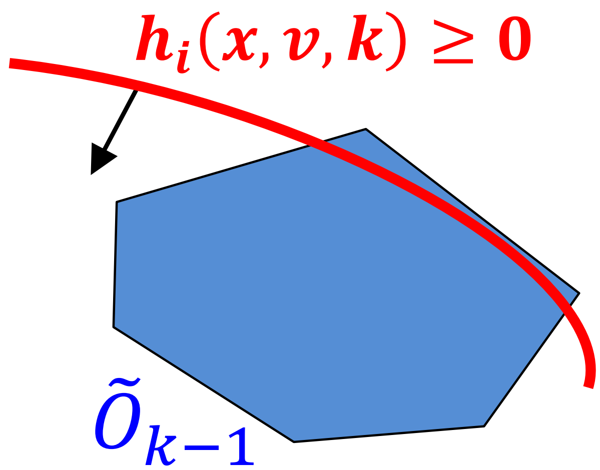
$$\rho = min_{x,r}\, h_i(x, v, k)$$

subject to

$$(x, v) \in \tilde{O}_{k-1}$$

- $\rho < 0 \quad \Rightarrow$ The constraint is **non-redundant**
- $\rho \geq 0 \quad \Rightarrow$ The constraint is **redundant**

Approach 2: Redundancy



$$\rho = \min_{x,r} h_i(x,v,k)$$

subject to

$$(x,v) \in \tilde{O}_{k-1}$$

ATTENTION: Convex only if $\tilde{O}_{k-1}$ and $h_i(x,v,k)$ are convex

IN PRACTICE: Usable only for linear systems with linear constraints

Approach 2: Example

ULB

Unstable linear system subject to input and state constraints

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y_p(t) = Cx(t) \end{cases} \qquad A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$-1.2 \leq y_p \leq 1.2 \qquad -0.3 \leq u \leq 0.3$$

Stabilizing controller, input gain for unitary dc-gain

$$K = \mathtt{place}(0.4 \pm 0.6j) \qquad A_{cl} = A + BK \qquad F = (\mathtt{dcgain}(A_{cl}, B, C_z, 0))^{-1}$$

Resulting closed-loop system for governor design

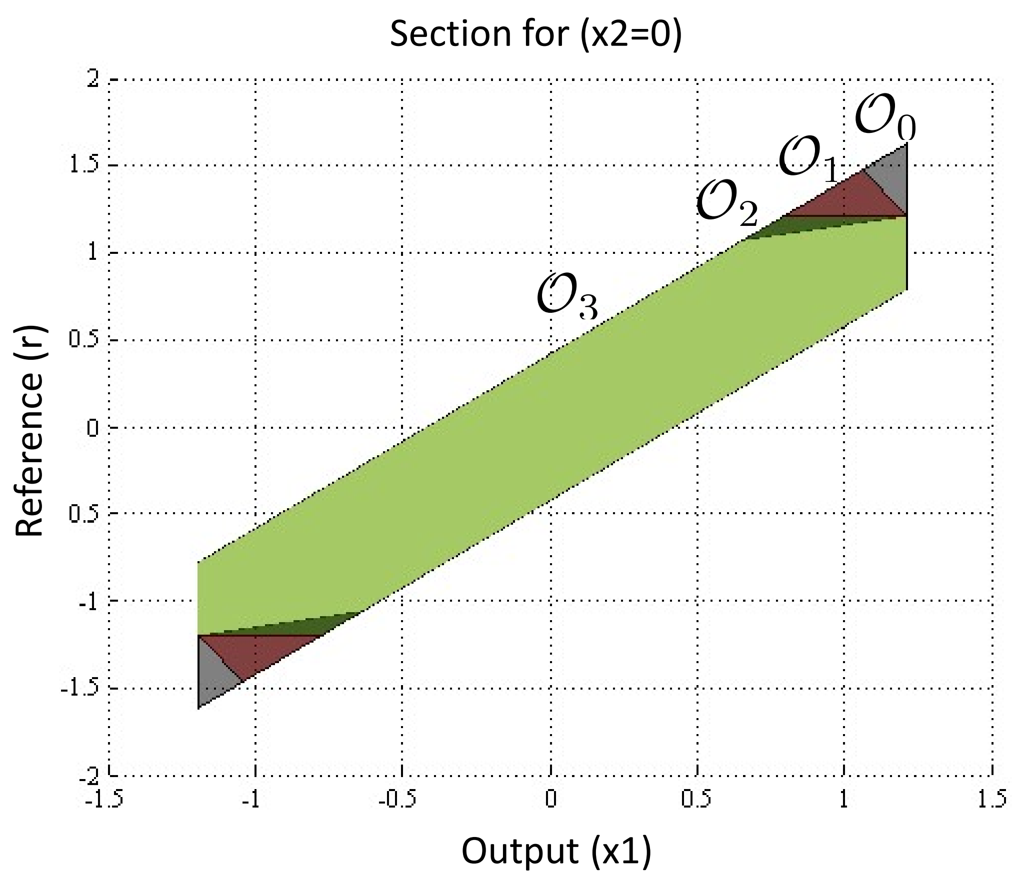
$$\begin{cases} x(t+1) = A_{cl}x(t) + B_{cl}v(t) \\ y_{cl}(t) = C_{cl}x(t) + D_{cl}v(t) \end{cases} \qquad B_{cl} = BF \quad D_{cl} = \begin{bmatrix} 0 \\ F \end{bmatrix} \quad C_{cl} = \begin{bmatrix} 1 & 0 \\ & K \end{bmatrix}$$

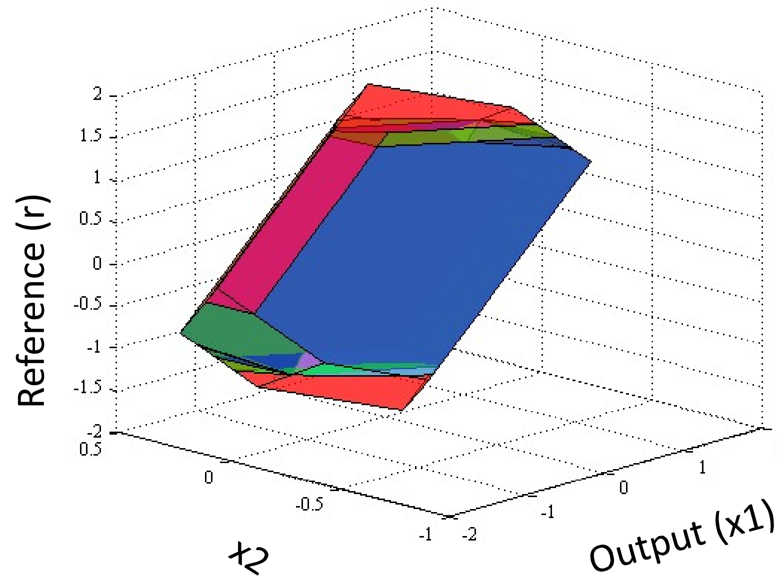
$$\mathcal{S} = \left\{ y : -\begin{bmatrix} 1.2 \\ 0.3 \end{bmatrix} \leq y \leq \begin{bmatrix} 1.2 \\ 0.3 \end{bmatrix} \right\} = \{y : Sy \leq s\}$$

Approach 2: Example

Construction of $\tilde{O}_\infty$ by forward constraint enumeration and redundancy check.

Determinedness index: $k^* = 5$





Note:
For this section $\tilde{O}_3, \tilde{O}_4, \tilde{O}_5$ coincide

Some remarks

Approaches exists to generate both invariant and not invariant set $P \subseteq O_\infty$:

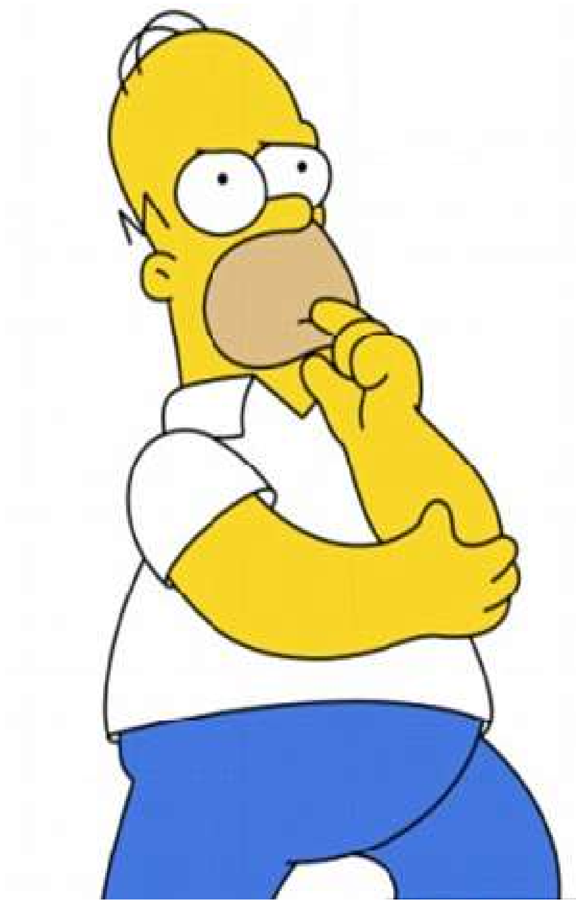
- Approaches based on **Reference Dependent Lyapunov level sets** (see e.g. *Cotorruelo, Hosseinzadeh, Ramirez, Limon, Garone "Reference dependent invariant sets: Sum of squares based computation and applications in constrained control". Automatica, 2021)*

- Approaches based on mixing **RDLP and predictions**

- Approaches based on direct **inner approximation of $\widetilde{O}_\infty$ through pull-in procedures** *Gilbert, Kolmanovsky. "Fast reference governors for systems with state and control constraints and disturbance inputs." International Journal of Robust and Nonlinear Control, 1999*.

Some remarks

OK now I've the *P*...



... what I've to implement ?

Reference Governor

Optimization Problem:

$$\max_{\kappa \in [0,1]} \kappa$$

subject to

$$\big(x(t), (1-\kappa)v(t-1) + \kappa\, r(t)\big) \in \; P$$

Note that this is a problem in 1 variable !

EFFICIENT ALGORITHMS:

- *Search by Bisection*
- *Analytic Solutions (linear constraints)*

RG - Bisection

```
INITIALIZATION
Kfeas = 0
Kinfeas = 1
K = Kinfeas

WHILE (Kinfeas-Kfeas<eps){
    IF ((x,vold*(1-K)+K*r)∈P)
        Kfeas = K
    ELSE
        Kinfeas = K
    K =(Kfeas+Kinfeas)/2
}

OUTPUT
K = Kfeas
```

Analytic - Linear

If $P = \{(x, v) |\ H_x x + H_v v \leq \bar{h}\}$

RG optimization Problem:

$$\max_{\kappa \in [0,1]} \kappa$$
$$subject\ to$$
$$H_x x(t) + H_v(\ (1-\kappa)v(t-1) + \kappa\, r(t)) \leq \bar{h}$$

We can rewrite the constraints as

$$\kappa\ \ (H_v(r(t) - v(t-1))) \leq \bar{h} - H_x\, x(t) + H_v v(t-1)$$

and then for $i = 1, \dots, n_c$

$$\kappa\ \ (h_{v,i}^T(r(t) - v(t-1))) \leq \bar{h_i} - h_{x,i}^T x(t) + h_{v,i}^T v(t-1)$$

Analytic - Linear

$$\max_{\kappa \in [0,1]} \kappa$$

$$subject\ to$$

$$\kappa \ \left(h_{v,i}^T(r(t) - v(t-1))\right) \leq \bar{h}_i - h_{x,i}^T x(t) + h_{v,i}^T v(t-1)$$

$$i = 1, \dots, n_c$$

Two cases:

- If $h_{v,i}^T\big(r(t) - v(t-1)\big) \leq 0$

  The condition becomes $0 \leq \kappa \ \leq 1$

- If $h_{v,i}^T\big(r(t) - v(t-1)\big) > 0$

  The condition becomes $0 \leq \kappa \leq \dfrac{\bar{h}_i - h_{x,i}^T x(t) + h_{v,i}^T v(t-1)}{h_{v,i}^T\big(r(t) - v(t-1)\big)}$

Analytic – Quadratic et al.

ULB

JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 40, No. 2, February 2017

Check for updates

Engineering Notes

Fast Reference Governor for Linear Systems

Marco M. Nicotra* and Emanuele Garone†
Free University of Brussels, 1050 Brussels, Belgium
and
Ilya V. Kolmanovsky‡
University of Michigan, Ann Arbor, Michigan 48109

DOI: 10.2514/1.G000337

where $x \in \mathbb{R}^n$, $v \in \mathbb{R}^p$, and A is a Schur matrix. The system is subject to J convex constraints

$$\mathcal{C}_j(x_t, v_t) \leq 0 \quad \begin{array}{l} \forall\, j = 1, \ldots, J \\ \forall\, t \geq 0 \end{array} \tag{2}$$

which reflect both state and input constraints, since system (1) represents a closed-loop system and v_t is the set point of the primary controller.

Given a desired reference r_t and a previously applied reference v_{t-1} that, if kept constant, ensures constraint satisfaction over the infinite horizon, the scalar reference governor assigns v_t using the linear interpolation

$$v_t = (1 - \lambda)v_{t-1} + \lambda r_t \tag{3}$$

oi: 10.2514/1.G000337

- Linear and quadratic constraints
- Mixing closed-form and bisection

Command Governor

Optimization Problem:

$$\min_{v} ||v - r(t)||^2$$
$$subject\ to$$
$$(x(t), v) \in P$$

OBSERVATIONS:

- This is a problem in m variables
- It is a tractable problem if P convex

THIS IS A (SIMPLE) CONVEX PROBLEM

Command Governor

EXAMPLE: Using MPT (Multi Parametric Toolbox)

$$\min_{v} \left|\left|v - r(t)\right|\right|^2$$
$$subject\ to$$
$$H_x x(t) + H_v v \leq \bar{h}$$

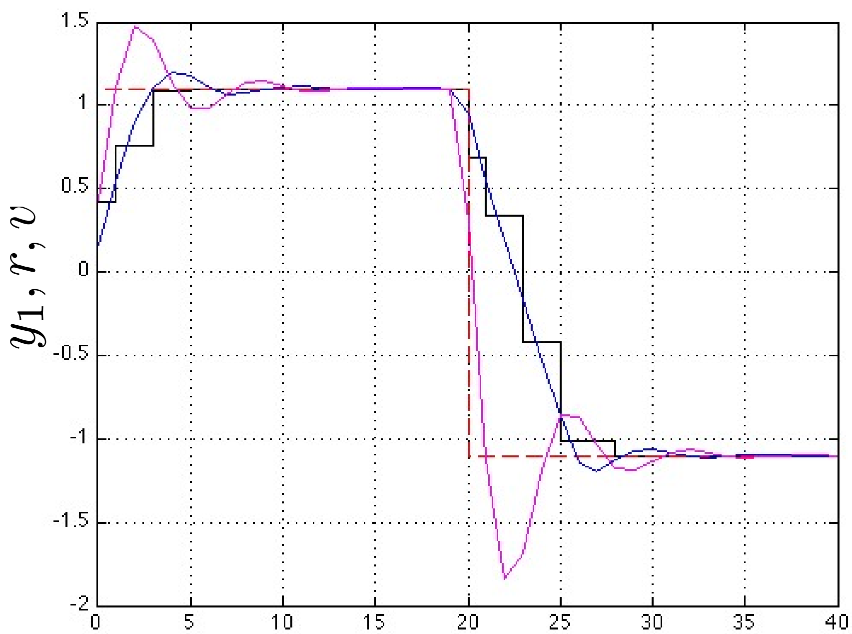
The solution of the above problems maps into

```
v=sdpvar(m,1)                 %define variables
J= (v-rt)'*(v-rt)             %define cost function
C=[H_x*xt+H_v*v <= barh]      % define constraints
optimize(J,C)                 % run the optimization
vt=double(v)                  % retrieve solution
```

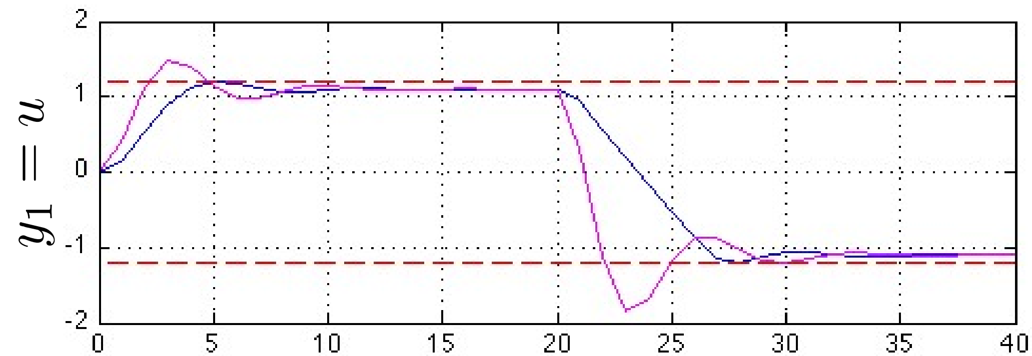
RG/CG how they work

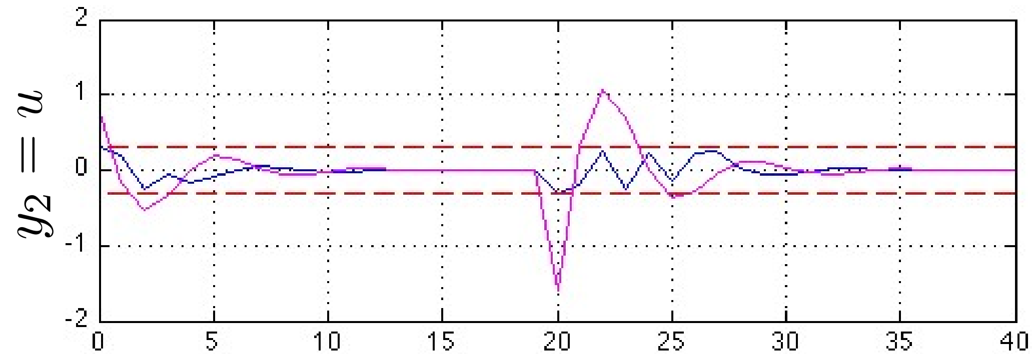
Back to our constrained controlled double integrator.

Reference tracking



Constrained outputs





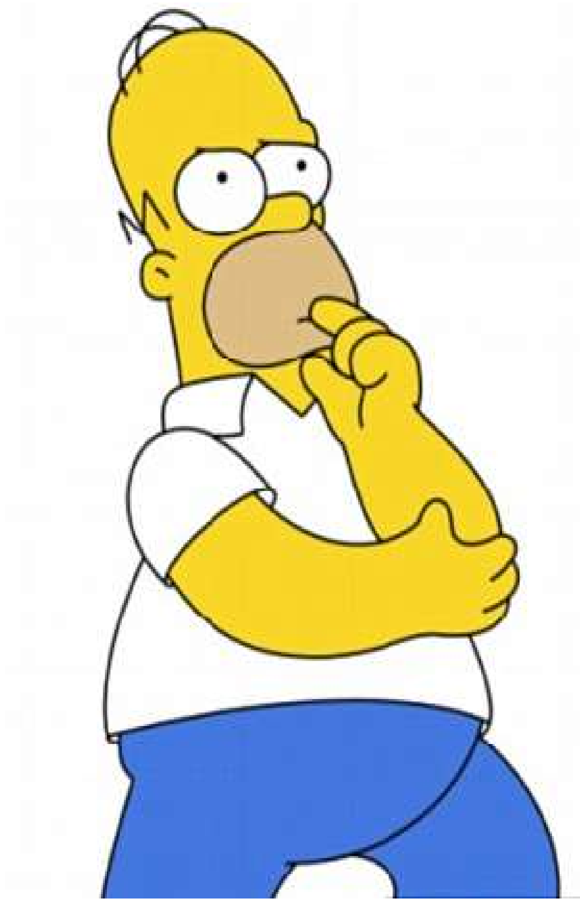
outputs with actual reference

outputs with reference governor

constraints

Extended Command Governor

OK cool but...



... if I want to go faster ?

Extended CG

IDEA: Instead of generate directly v, generate it through a fictitious stable system

$\rho(t)$ → and $\chi(t)$ →

$$\chi(t+1) = A_\chi \chi(t)$$
$$v(t) = C_\chi \chi(t) + \rho(t)$$

$\xrightarrow{v(t)}$

$$x(t+1) = Ax(t) + Bv(t)$$
$$y(t) = Cx(t) + Dv(t)$$

Equivalent system:

$$\begin{bmatrix} x(t+1) \\ \chi(t+1) \end{bmatrix} = \begin{bmatrix} A & BC\chi \\ 0 & A_\chi \end{bmatrix} \begin{bmatrix} x(t) \\ \chi(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \rho$$

We can compute $\widetilde{O}_\infty$ for the extended system

Choice of the system

- Shift register of length n_χ:

$$A_\chi = \begin{bmatrix} 0 & I & 0 & \cdots \\ 0 & 0 & I & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix}, \qquad C_\chi = [I \quad 0 \quad \cdots \quad 0]$$

- Laguerre Sequence Generator:

$$\bar{A} = \begin{bmatrix} \varepsilon I & \beta I & -\varepsilon\beta I & \varepsilon^2\beta I & \cdots \\ 0 & \varepsilon I & \beta I & -\varepsilon\beta I & \cdots \\ 0 & 0 & \varepsilon I & \beta I & \cdots \\ 0 & 0 & 0 & \varepsilon \mathrm{I} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

$$\bar{C} = \sqrt{\beta}[I \quad -\varepsilon \mathrm{I} \quad \varepsilon^2 I \quad -\varepsilon^3 I \quad \cdots \quad (-\varepsilon)^{N-1} I]$$

where $\beta = 1 - \varepsilon^2$, and $0 \leq \varepsilon \leq 1$.

Extended CG

Optimization Problem:

$$\min_{\rho,\chi} \left\| v - r(t) \right\|^2 + \left\| \chi \right\|_S^2$$

subject to

$$\left(\begin{bmatrix} x(t) \\ \chi \end{bmatrix}, \rho \right) \in \tilde{O}_\infty$$

Remarks:

1. Same Structure of Command Governor
2. The performance are improved
3. Higher computational cost
4. How to choose A_χ and C_χ ?

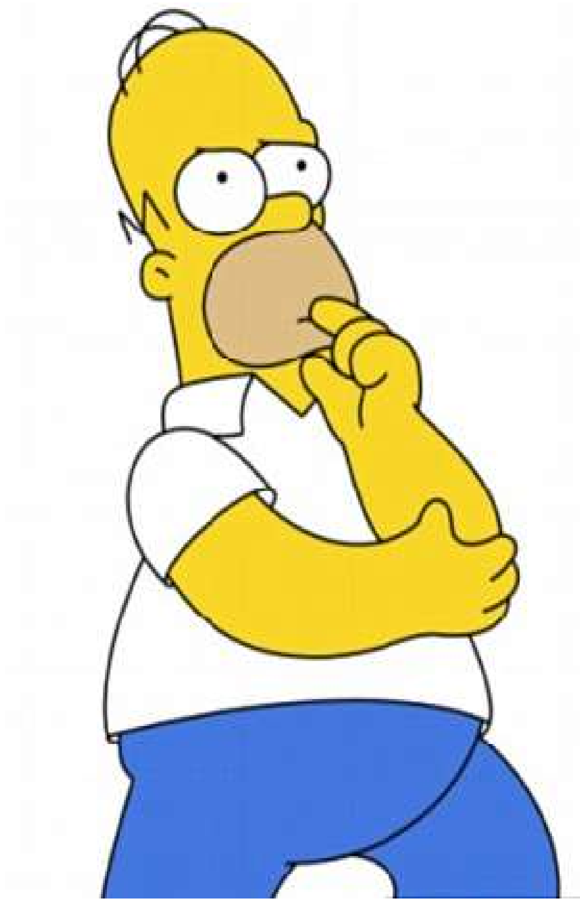
Disturbances

Reference and Command Governor

The case with disturbance

Disturbances

ok...



... and if there are disturbances ?

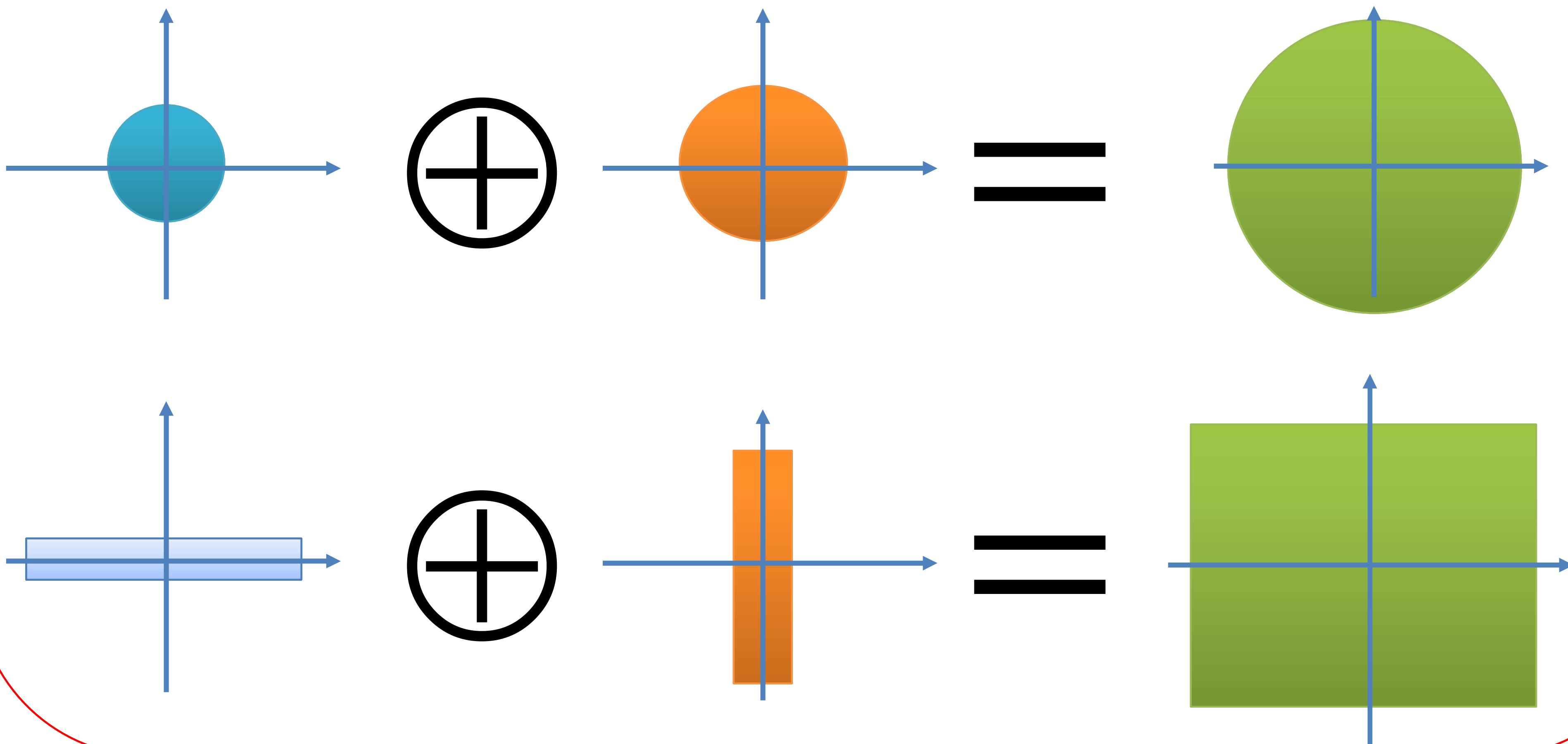
Tools

Some useful tool

Tools

USEFUL TOOL 1: Minkowsky Set-sum

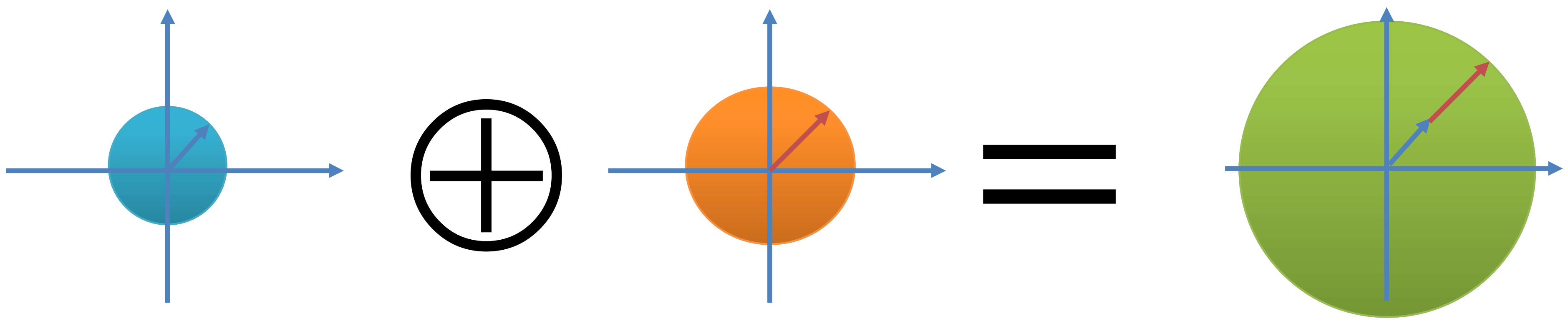
$$S_1 \oplus S_2 = \{s = s_1 + s_2 \mid s_1 \in S_1, s_2 \in S_2\}$$



Tools

Example 1

$$S_1 \oplus S_2 = \{s = s_1 + s_2 |\, s_1 \in S_1, s_2 \in S_2\}$$

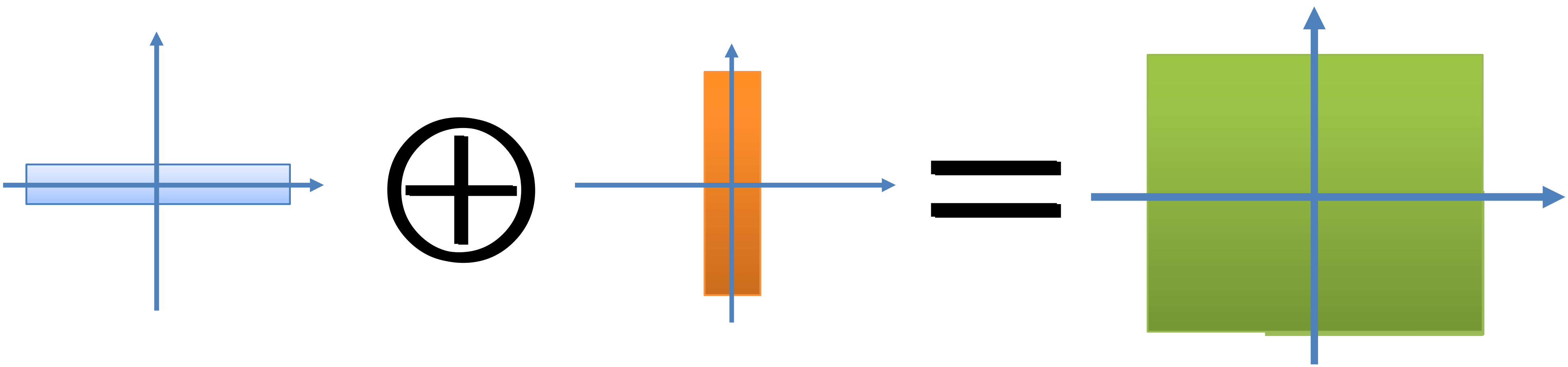


- $S_1 = \{s : ||s|| \leq a_1\}$
- $S_2 = \{s : ||s|| \leq a_2\}$
- $S_1 \oplus S_2 = \{s = s_1 + s_2 : ||s_1|| \leq a_1, ||s_2|| \leq a_2\} =$
  $= \{s : ||s|| \leq a_1 + a_2\}$

ULB

Example 2

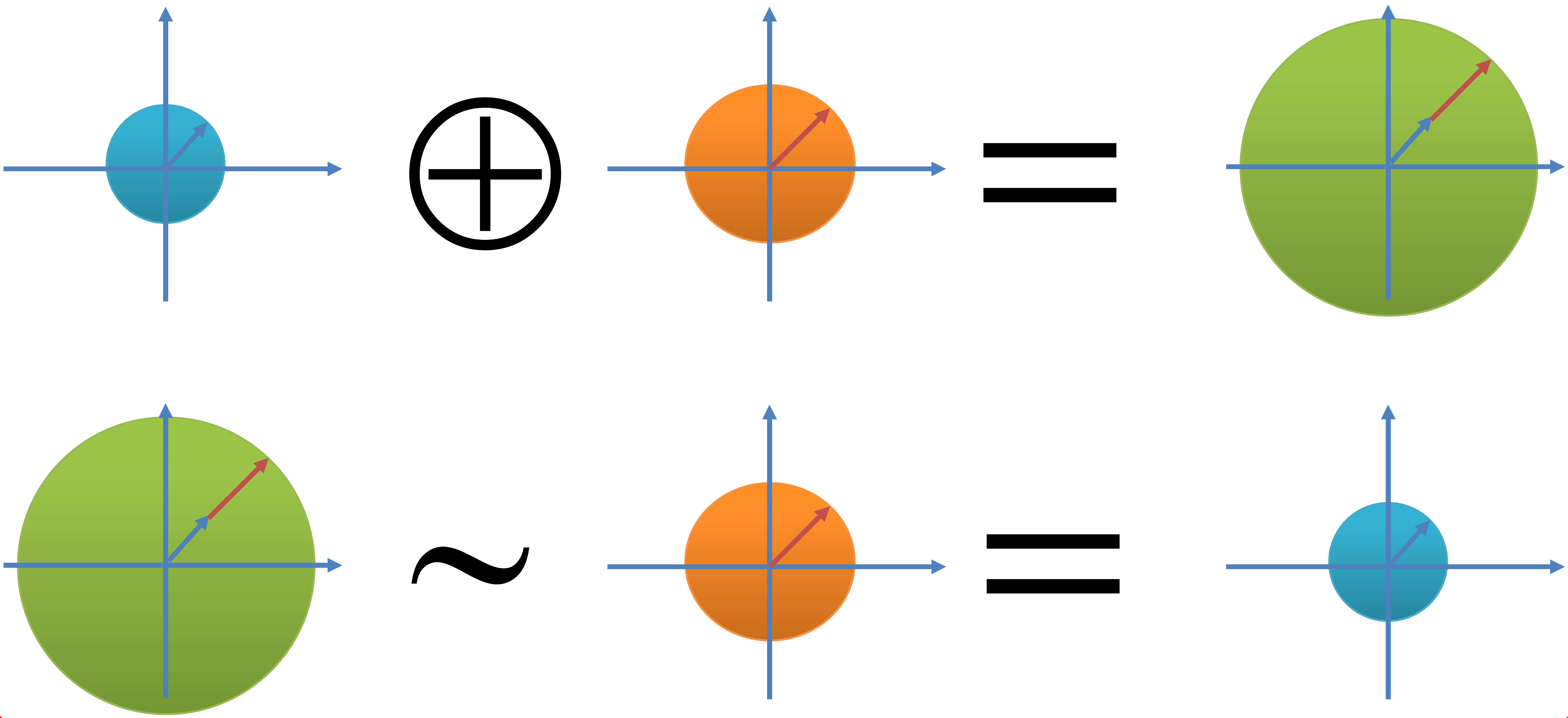
$$S_1 \oplus S_2 = \{s = s_1 + s_2 |\, s_1 \in S_1, s_2 \in S_2\}$$



- $S_1 = convh\{v_{1,i} | i = 1, .., n_1\}$
- $S_2 = convh\{v_{2,i} | i = 1, .., n_2\}$
- $S_1 \oplus S_2 = convh\{v_{1,i} + v_{2,j} | i = 1, .., n_1, j = 1, .., n_2\}$

USEFUL TOOL: Pontryagin-Minkowsky Set-difference

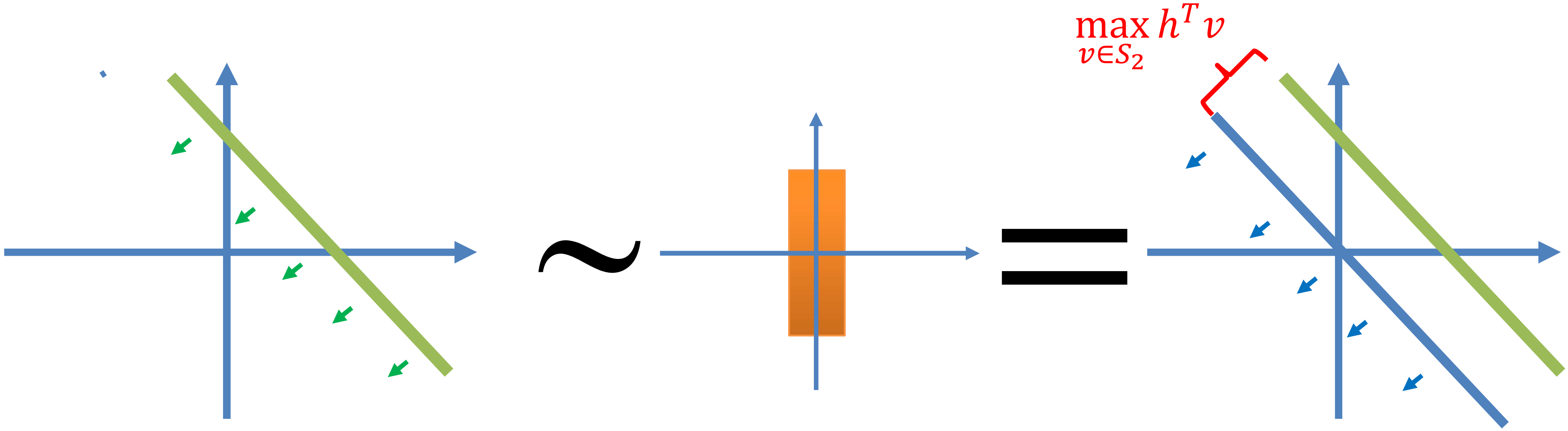
$$S_1 \sim S_2 = \{s\colon\ s + s_2 \in S_1, \forall s_2 \in S_2\}$$



ULB

Pontryagin-Minkowsky Set-difference

$$S_1 \sim S_2 = \{s\colon\ s + s_2 \in S_1, \forall s_2 \in S_2\}$$

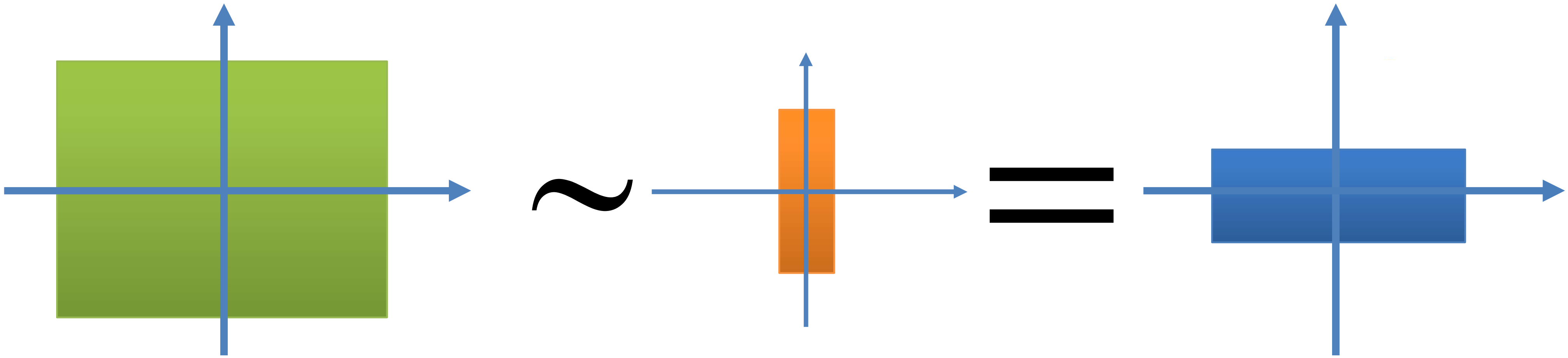


- $S_1 = \{s \mid h^T s \le \bar{h}\}$
- $S_2 = \{Ws \le \bar{w}\}$
- $S_1 \sim S_2 = \{s \mid h^{\mathrm{T}} s \le \bar{h} - \max_{v \in S_2} Sv\}$

Tools

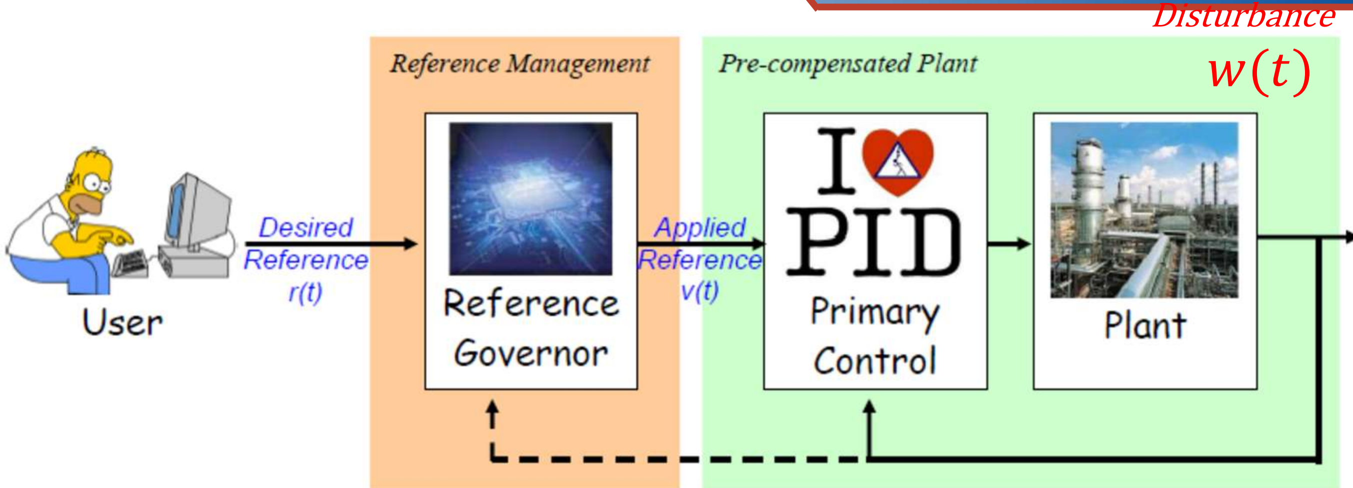
Pontryagin-Minkowsky Set-difference

$$S_1 \sim S_2 = \{s:\ s + s_2 \in S_1, \forall s_2 \in S_2\}$$



- $S_1 = \{s|\ h_i^T s \leq \bar{h}_i, \quad i = 1, \ldots, l\}$
- $S_2 = \{Ws \leq \bar{w}\}$
- $S_1 \sim S_2 = \{s|h_i^{\mathrm{T}} s \leq \bar{h}_i - \max_{v \in S_2} h_i^T v, \quad i = 1, \ldots, l$

Idea



- **System**:

$$x(t+1) = Ax(t) + Bv(t) + B_w w(t)$$

- **Constraints**: $x(t) \in C$
- **Bounded Disturbance**: $w(t) \in W$

The IDEA: To use the Superposition Principle

Idea

Nominal Prediction:

$$x_n(k|x,v) = A^k x_n(t) + \sum_{i=0}^{k-1} A^{k-i-1} Bv + Dv$$

+

Disturbance Prediction:

$$\hat{x}_d(k) = \sum_{i=0}^{k-1} A^{k-i-1} B_w w(i)$$

=

Overall Prediction:

$$\hat{x}(k|x,v) = A^k x + \sum_{i=0}^{k-1} A^{k-i-1} \begin{bmatrix} B & B_w \end{bmatrix} \begin{bmatrix} v \\ w(i) \end{bmatrix}$$

Problem: how to evaluate $\hat{x}_d(k)$

Idea

Disturbance System:

$$\hat{x}_d(k) = \sum_{i=0}^{k-1} A^{k-i-1} B_w w(i)$$

Problem : The sequence $\{w(i)\}_{i=0}^{k}$ is not known in advance

Consequences:

- Predictions $\forall\, w(i) \in W, i = 0, \ldots k$
- $\hat{x}_d(k)$ belong to a set, X_k

ULB

Idea

k=1

$$\hat{x}_d(1) = B_w w(0), \qquad w(0) \in \mathrm{W}$$

$$X_1 = B_w W$$

Remark: if $\quad W = convh\{w_1, \dots, w_p\}$

then $\quad B_w W = convh\{B_w w_1, \dots, B_w w_p\}$

k=2

$$\hat{x}_d(2) = \underbrace{B_w w(1)}_{B_w W} + \underbrace{A B_w w(0)}_{A B_w W}, \forall\, w(0), w(1) \in W$$

$$X_2 = \{x = x_1 + x_2 |\, x_1 \in C B_w W, x_2 \in C A B_w W\ \} =$$
$$= B_w W \oplus A B_w W \qquad = X_1 \oplus A B_w W$$

Idea

For any k

Disturbance Predictions:

$$\hat{x}_d(k) = \sum_{i=0}^{k-1} A^{k-i-1} B_w w(i), \forall w(i) \in W$$

Set Predictions:

$$X_k = \oplus_{i=0}^{k-1} A^{k-i-1} B_w W$$

$$X_{k+1} = X_k \oplus A^k B W$$

Properties:

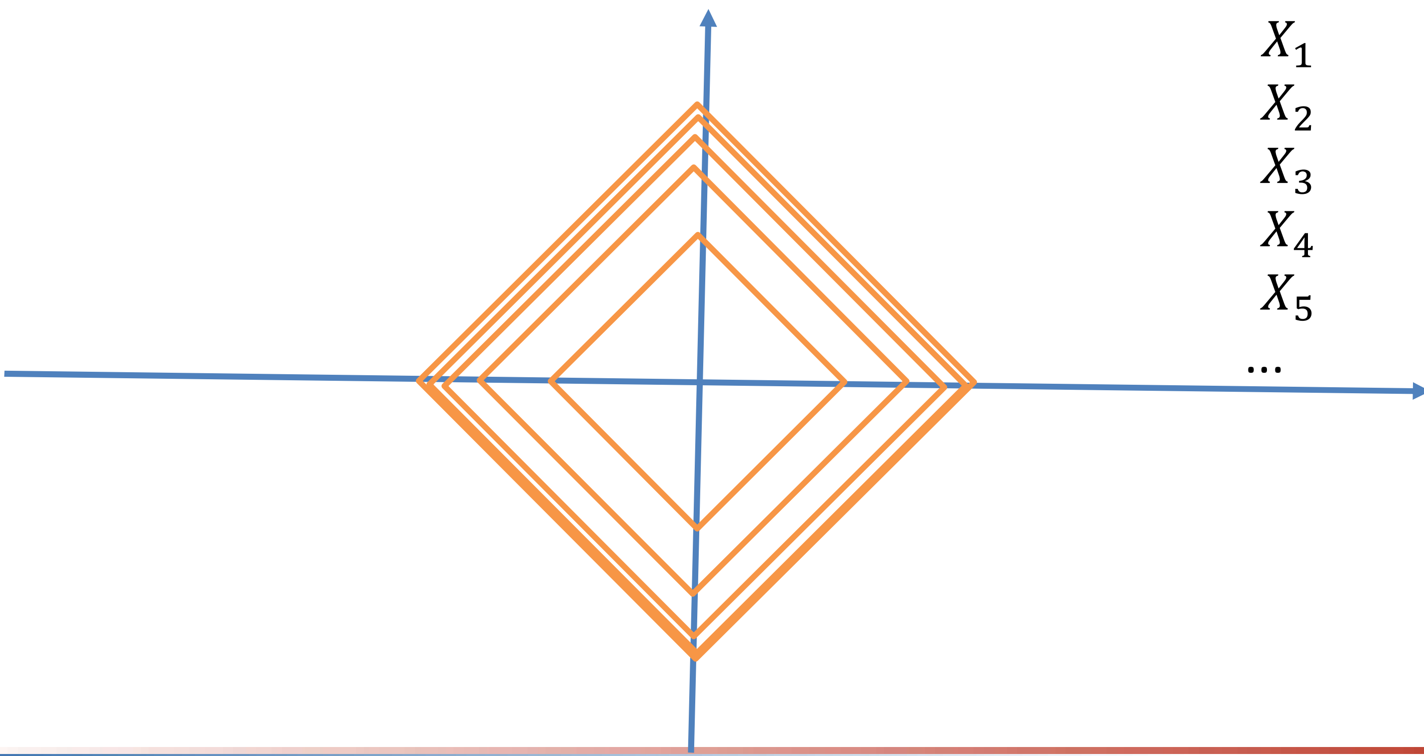
- $if\ 0 \in W,$ $\quad X_{k+1} \supseteq X_k$
- $if\ A\ is\ A.S.,$ $\quad \lim_{k\to\infty} X_k = X_\infty$

Idea

Example:

$$x(t+1) = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w(t)$$

$$w(t) \in W$$



Idea

Nominal Prediction:

$$\hat{x}_n(k|x,v) = A^k x + \sum_{i=0}^{k-1} A^{k-i-1} B v$$

Disturbance Prediction:

$$\hat{x}_d(k) = \sum_{i=0}^{k-1} A^{k-i-1} B_w w(i)$$

$$X_k = \bigoplus_{i=0}^{k-1} A^{k-i-1} B_w W$$

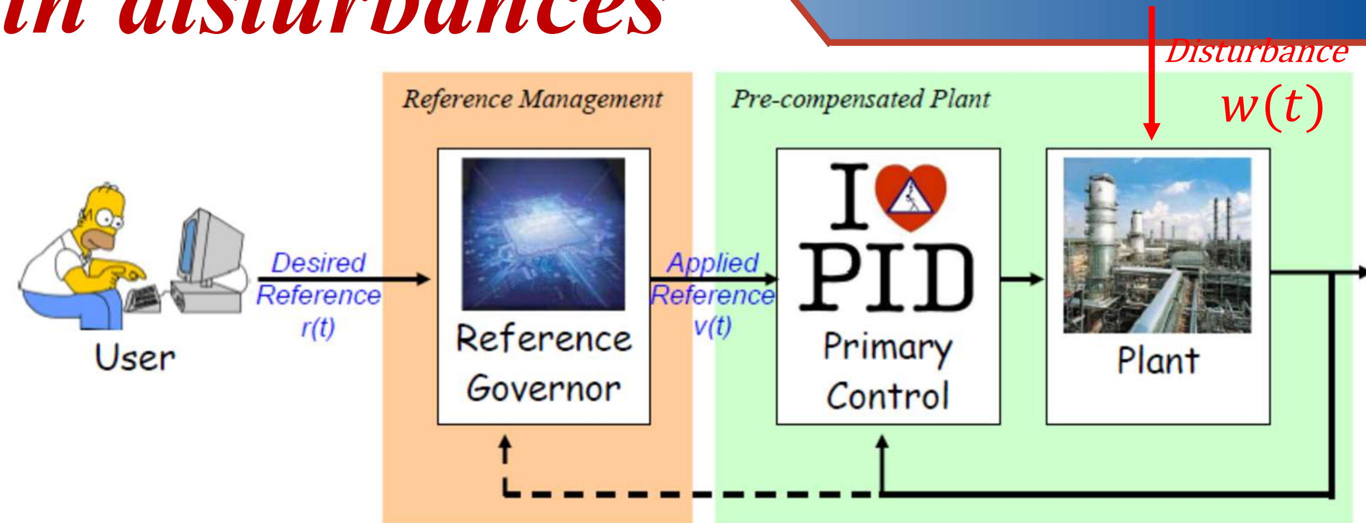
Overall Prediction:

$$\hat{x}(k|x,v) = A^k x + \sum_{i=0}^{k-1} A^{k-i-1} \begin{bmatrix} B & B_w \end{bmatrix} \begin{bmatrix} v \\ w(i) \end{bmatrix}$$

$$\hat{x}(k|x,v) = \hat{x}_n(k|x,v) \oplus X_k$$

Idea: $x(k|x,v) \in C \quad \Leftrightarrow \quad \hat{x}_n(k|x,v) \in C \sim X_k$

RG with disturbances



Nominal $\mathcal{O}_\infty$

$$\hat{x}(k|x,v) \in C, \qquad k = 0,\dots,\infty$$

$\mathcal{O}_\infty$ with Disturbance

$$\hat{x}_n(k|x,v) \in C \sim X_k, \qquad k = 0,\dots,\infty$$

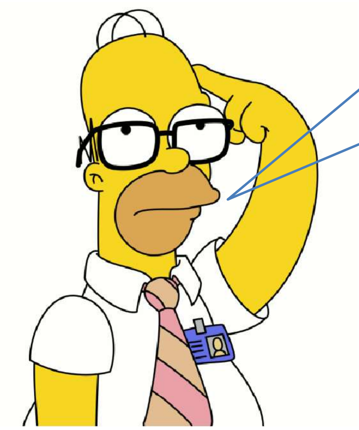
RG with disturbances

$\mathcal{O}_\infty$ with Disturbance

$$\hat{x}_n(k|x,v) \in C{\sim}X_k, \qquad k=0,\ldots,\infty$$

The linear RG can be reused !

RG with disturbances



Wait,
$\hat{x}_n(k|x,v) \in C \sim X_k, \qquad k = 0, \ldots, \infty$
is a time-varying constraint

Answer:
X_k converge asymptotically to $X_\infty \supseteq X_k$

Expedient 1:
$$\hat{x}_n(k|x,v) \in C \sim X_\infty, \;\; k = 0, \ldots, \infty$$

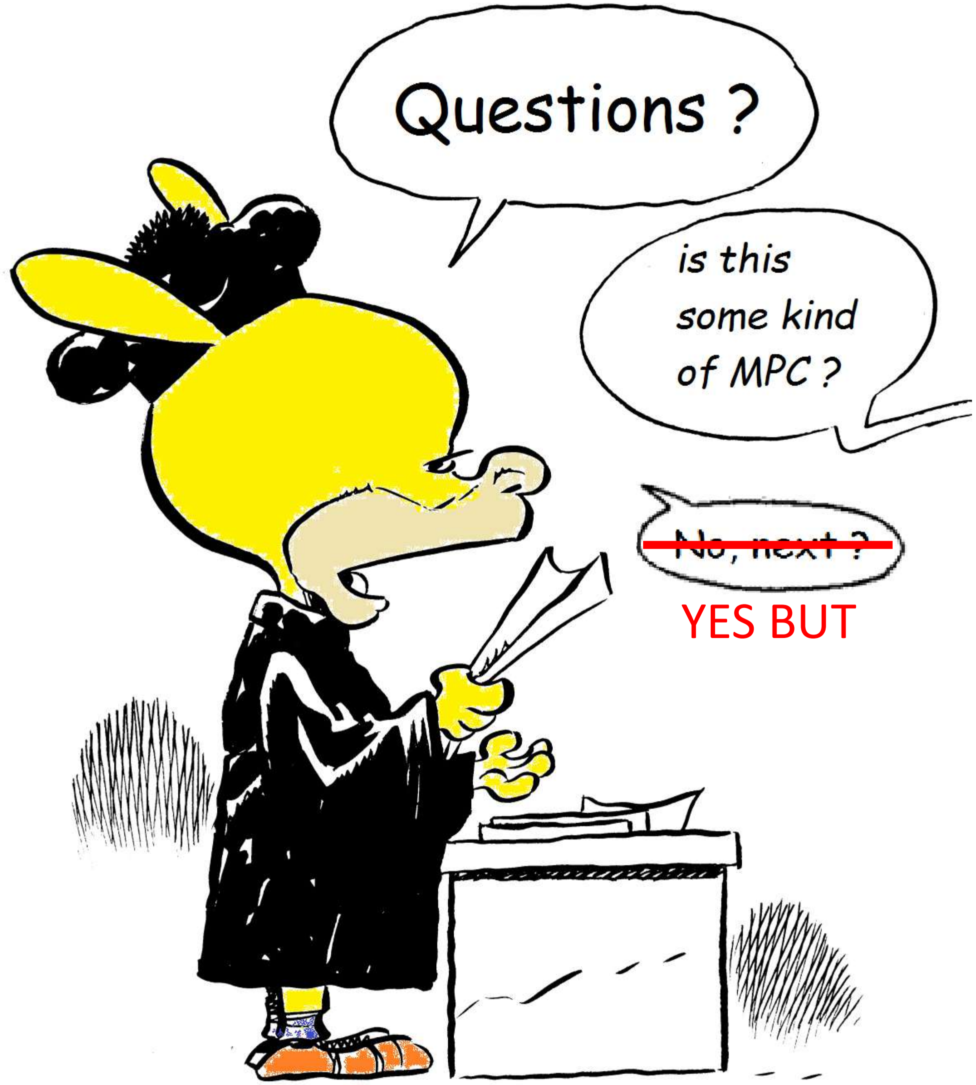
Expedient 2:
$$\hat{x}_n(k|x,v) \in C \sim X_k, \qquad k = 0, \ldots, N$$
$$\hat{x}_n(k|x,v) \in C \sim X_\infty, k = N+1, \ldots, \infty$$

Summary

What we have seen today:

- Maximum Constraint Admissible Set
- Linear RG and CG
- How to implement it
- How to improve performance
- How to incorporate disturbances

Thank you



A huge thanks to Leo Ortolani for authorizing the (ab)use of his artwork and of the Ratman character.
No rat has been harmed during the realization of this presentation.