Robust Control Workshop
Delft Center for Systems and Control

Mixed Deterministic/Randomized Methods for Fixed Order Controller Design

Roberto Tempo
IEIIT-CNR
Politecnico di Torino
tempo@polito.it

Acknowledgments

- Joint work with Yasumasa Fujisaki and Yasuaki Oishi
- Thanks to Shinji Hara and Toshiharu Sugie for technical discussions and for hosting me in their respective institutions

Computationally difficult problems

- Various robust control problems are computationally intractable
- This is generally meant for $\mathcal{NP}$-hard problems
- Examples:
  - stability of interval matrices
  - μ computation
  - static output feedback

Deterministic methods

- Deterministic methods can be successfully used only for computationally tractable problems
- Example: Convex problems
- Many (difficult) control problems are not convex
- Finding local solutions instead of global solutions
- Use of relaxation, overbounding or other techniques

Randomized methods

- Randomized methods are often used in computer science, computational geometry, optimization, …
- Problems: Data structuring and search trees, graph algorithms, linear program, …
- Remarkably, randomized methods are not used *systematically* in uncertain systems and control
- Development of mathematically rigorous methods, not straightforward use of Monte Carlo simulations

IEIIT-CNR

Randomized algorithms (RAs)

- Randomized algorithms are *probably approximately correct* (PAC)
- They provide an approximate solution only with some given probability
- Give up a guaranteed deterministic solution
- Obtain polynomial-time complexity

IEIIT-CNR

Early work on the topic

- First appearance in [Stengel, 80]
- Motivations: Flight control (applications) and stochastic optimal control (theory)
- CDC paper titled "*Probabilistic robust controller design*" by [Djavdan, Tulleken, Voetter, Verbruggen, Olsder, 89]
- Drawback: Not much theory, development of Monte Carlo methods for analysis problems

IEIIT-CNR

Subsequent work on the topic

- Finite sample size bounds given in [Khargonekar and Tikku, 96; Tempo, Bai and Dabbene, 96]
- Statistical learning theory for control design [Vidyasagar, 98]

IEIIT-CNR

Deterministic and randomized methods

Deterministic and randomized methods

deterministic methods

↓

randomized methods

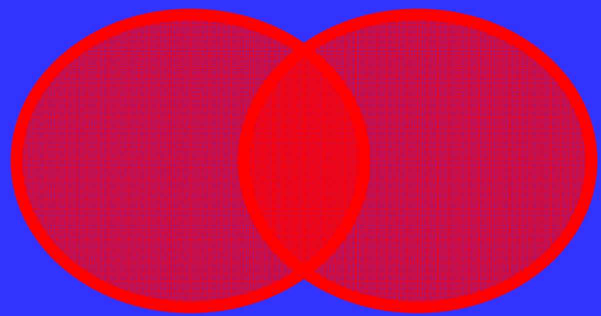
↓

This talk

Mixed deterministic/randomized methods

deterministic/randomized
methods

↓



Tractable/intractable parameters

Controller $\theta \quad \eta$

- Controller parameters θ and η

- Tractable parameters θ
- (Computationally) intractable parameters η
- ***Deterministic*** methods for θ
- ***Randomized*** methods for η

Plant and controller

- SISO strictly proper plant

$$P(s) = \frac{N_P(s)}{D_P(s)}$$

- Fixed order controller

$$C(s) = \frac{N_C(s)}{D_C(s)} = \frac{X(s^2)+sY(s^2)}{Z(s^2)+sV(s^2)}$$

Controller polynomials

- The even-order controller polynomials are of the form

$$X(s^2) = \theta_0 + \theta_2 s^2 + \theta_4 s^4 + \dots$$
$$Y(s^2) = \alpha_0 + \alpha_2 s^2 + \alpha_4 s^4 + \dots$$
$$Z(s^2) = \beta_0 + \beta_2 s^2 + \beta_4 s^4 + \dots$$
$$V(s^2) = \mu_0 + \mu_2 s^2 + \mu_4 s^4 + \dots$$

- Closed-loop polynomial is given by

$$p(s) = N_P(s)\left(X(s^2)+sY(s^2)\right) + D_P(s)\left(Z(s^2)+sV(s^2)\right)$$

- We assume that $p(s)$ has invariant degree (generic subset of controller parameters)

Parameter partitioning

1. Coefficients of $X(s^2)$ are tractable; coefficients of $Y(s^2)$, $Z(s^2)$, $V(s^2)$ are computationally intractable
2. Coefficients of $Y(s^2)$ are tractable; coefficients of $X(s^2)$, $Z(s^2)$, $V(s^2)$ are computationally intractable
3. Coefficients of $Z(s^2)$ are tractable; coefficients of $X(s^2)$, $Y(s^2)$, $V(s^2)$ are computationally intractable
4. Coefficients of $V(s^2)$ are tractable; coefficients of $X(s^2)$, $Y(s^2)$, $Z(s^2)$ are computationally intractable

IEIIT-CNR

Tractable/intractable parameters

- We consider the first case
- Then, we have

$\theta = [\theta_0 \;\; \theta_2 \;\; \theta_4 \; \ldots]^T$ ***tractable parameters***

$\eta = [\alpha_0 \, \alpha_2 \;\; \alpha_4 \; \ldots \; \beta_0 \;\; \beta_2 \; \beta_4 \; \ldots \; \mu_0 \, \mu_2 \mu_4 \; \ldots]^T$

intractable parameters

- We assume that $\theta \in \mathcal{M}$ and $\eta \in \mathcal{N}$

$n_\theta = \deg(X)/2 + 1$

$n_\eta = \deg(Y)/2 + \deg(V)/2 + \deg(Z)/2 + 3$

Example: Second order controller

- $N_C(s) = X(s^2)+sY(s^2) = \theta_0 + \alpha_0 s + \theta_2 s^2$

  $D_C(s) = Z(s^2)+sV(s^2) = \beta_0 + \mu_0 s + \beta_2 s^2$

  $\theta = [\theta_0 \;\; \theta_2]^T$ ***tractable parameters***

  $\eta = [\alpha_0 \, \beta_0 \;\; \beta_2 \;\; \mu_0]^T$ ***intractable parameters***

- $n_\theta = 2$

  $n_\eta = 4$

Randomized Methods for Intractable Parameters

IEIIT-CNR

RAs for intractable parameters

Definition (feasibility)

- Intractable parameter $\eta \in \mathcal{N}$ is feasible if the set of stabilizing controllers is not empty

- Take a probability measure $\mathcal{P}$ associated to $\mathcal{N}$
- Let $\varepsilon \in (0,1)$ and $\delta \in (0,1)$ be *confidence* and *accuracy* and consider

$$N_1 = \frac{\log(\delta)}{\log(1-\varepsilon)}$$

IEIIT-CNR

Feasibility algorithm

Algorithm 1

1. for $i := 1, \ldots, N_1$ do
   begin
2. draw a sample $\eta^{(i)} \in \mathcal{N}$ according to $\mathcal{P}$
3. if $\eta^{(i)}$ is feasible then return
   end

Probability of feasibility

- Let $\mathcal{A}$ be the set of all feasible intractable parameters η and $\mathcal{P}(\mathcal{A})$ its measure
- Suppose that $\mathcal{P}(\mathcal{A}) > \varepsilon$
- Theorem

  The probability that no $\eta^{(i)}$ given by Algorithm 1 is feasible is less than δ
- Proof follows from the "log-over-log" bound

Comments

- Theorem says that Algorithm 1 gives a feasible $\eta^{(i)}$ with confidence higher than $1 - \delta$
- Sample size N_1 depends only on confidence and accuracy
- Algorithm 1 is polynomial-time because sample generation (step 2) and feasibility check (step 3) are polynomial-time
- Bound N_1 is given in [Khargonekar and Tikku, 96; Tempo, Bai and Dabbene, 96]
- Specific instance of fpras theory, see [Tempo, Calafiore and Dabbene, 05]

RAs for intractable parameters

- Let $\varepsilon \in (0,1)$ and $\delta \in (0,1)$ and consider

$$N_2 = \frac{1}{2\varepsilon^2} \log \frac{2}{\delta}$$

- This is the well-known *Chernoff bound* [Chernoff, 52]
- Bound N_2 is independent of the number of intractable parameters, and depends only on accuracy and confidence

Algorithm for measure computation

Algorithm 2

1. set $N_s := 0$
2. for $i := 1, \ldots, N_2$ do

   begin
3. draw a sample $\eta^{(i)} \in \mathcal{N}$ according to $\mathcal{P}$
4. if $\eta^{(i)}$ is feasible then set $N_s := N_s + 1$

   end

Measure computation

- Let $\mathcal{A}$ be the set of all feasible intractable parameters η and $\mathcal{P}(\mathcal{A})$ its measure
- <u>Theorem</u>

  The probability that

  $$\left| N_s/N_2 - \mathcal{P}(\mathcal{A}) \right| > \varepsilon$$

  holds is less than δ
- <u>Proof</u> follows easily from the Chernoff bound

IEIIT-CNR

Comments

- Theorem says that Algorithm 2 gives an estimate of the feasibility measure $\mathcal{P}(\mathcal{A})$ of $\mathcal{A}$ with accuracy at least ε and confidence higher than 1 – δ
- Sample size N_2 depends only on confidence and accuracy
- Algorithm 2 is polynomial-time because sample generation (step 2) and feasibility check (step 3) are polynomial-time
- "Best" proof of the Chernoff bound is based on *Hoeffding inequality*

Deterministic Methods for Tractable Parameters

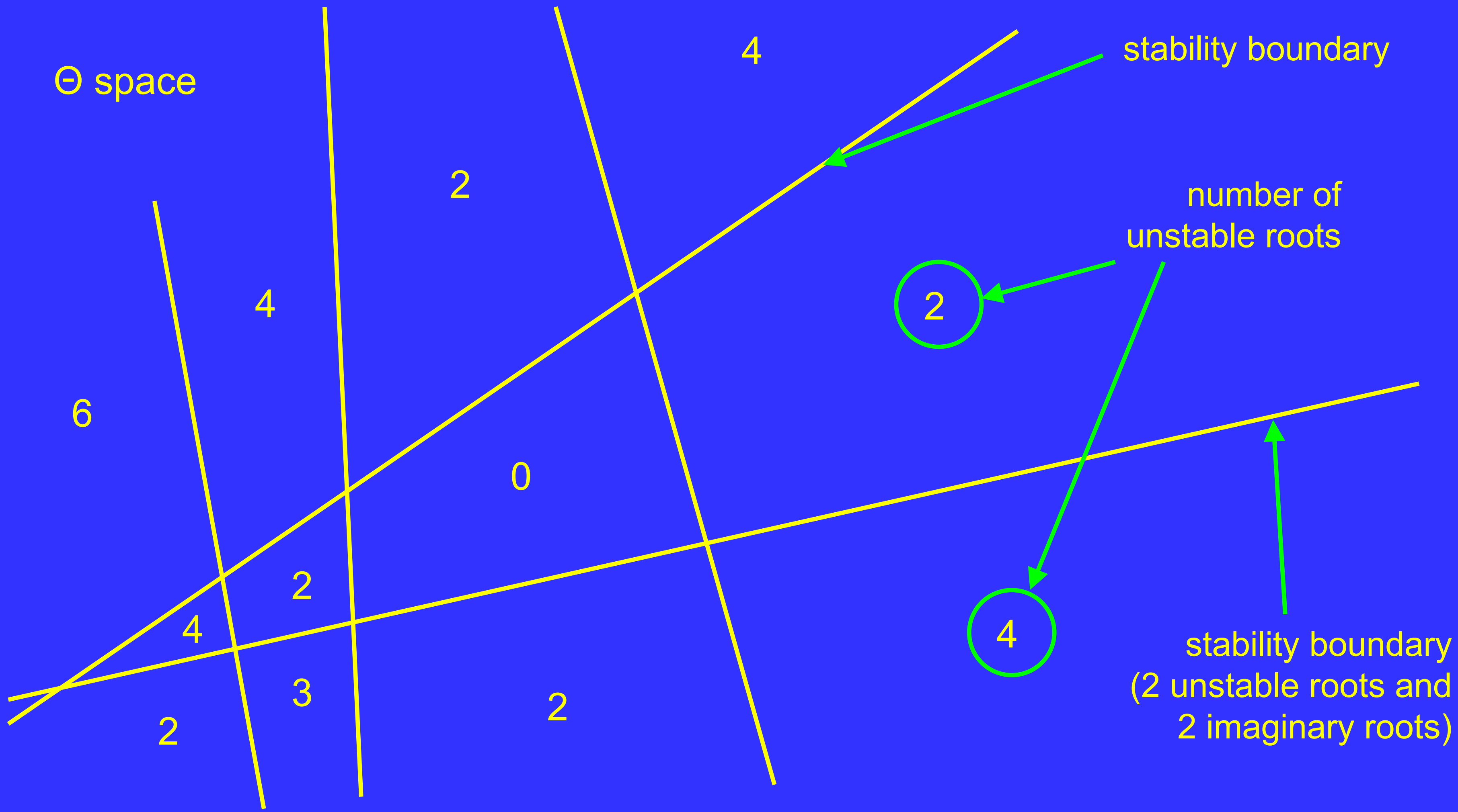
Set of stabilizing controllers

- Suppose that an intractable parameter vector η is selected according to Algorithm 1

- <u>Lemma</u>

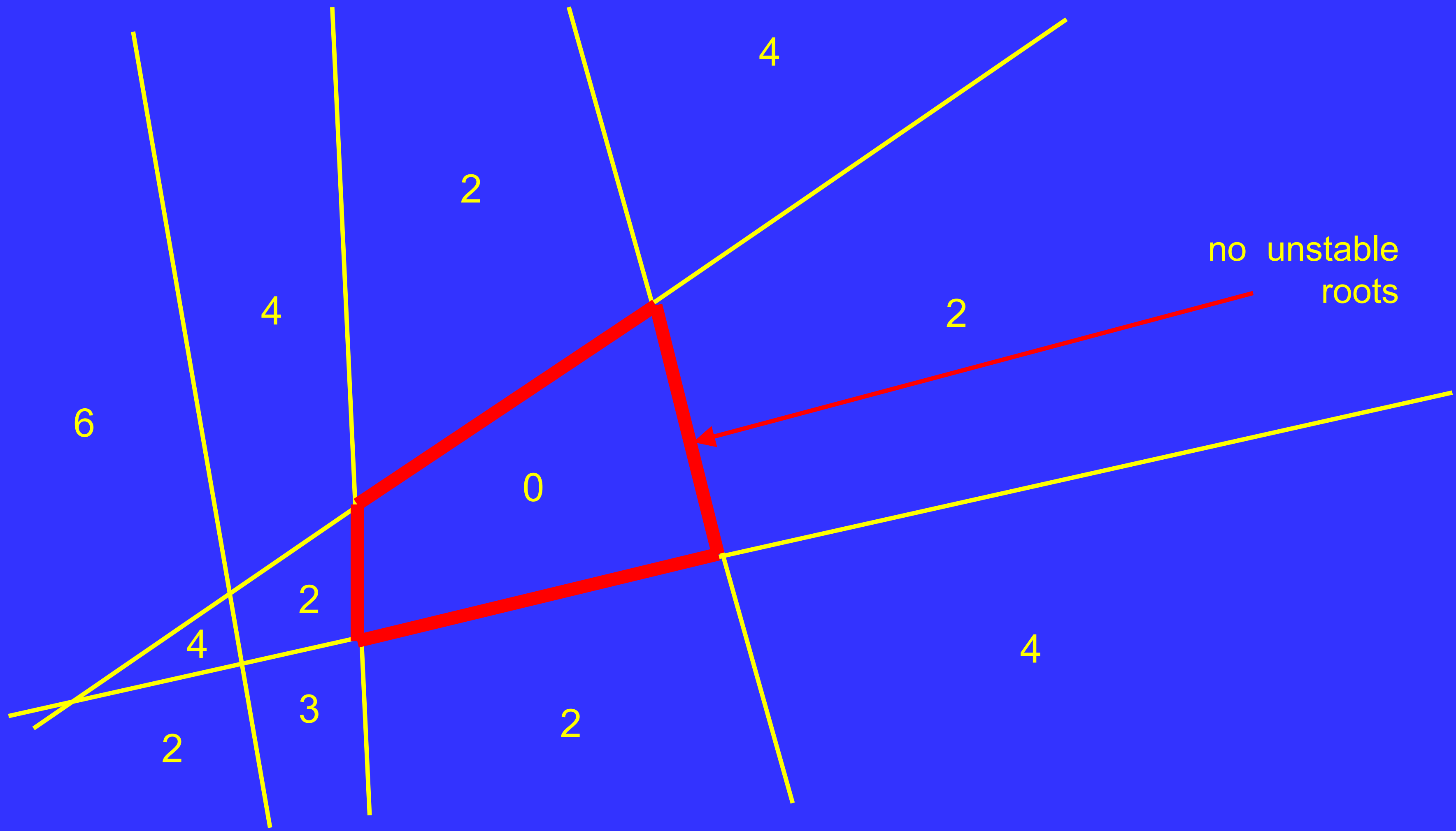
  The set of all tractable parameter vectors θ giving a stabilizing controller is either empty or is a union of a finite number of polyhedral sets

Tractable parameter space



IEIIT-CNR

Set of stabilizing controllers



Related literature

- Previous lemma is a minor extension of results first developed for PID controllers [Ho, Datta, Bhattacharyya, 97]
- Linear programming approach proposed, even if there are only two gains K_P and K_D
- Additional results in [Ackermann, Kaesbauer, 03; Soylemez, Munro, Baki, 03]
- Extensions to lead-lag in [Blanchini, Lepsky, Miani, Viaro, 04]

IEIIT-CNR

Closed-loop polynomial

- Consider the closed-loop polynomial

$$p(s) = p_0(s) + p_1(s)\, X(s^2)$$

where

$$p_0(s) = sN_P(s)Y(s^2) + D_P(s)\left(Z(s^2) + s\, V(s^2)\right)$$

$$p_1(s) = N_P(s)$$

- For $s=j\omega$, we have

$$p_0(j\omega) = R_0(\omega^2) + j\omega\, I_0(\omega^2)$$

$$p_1(j\omega) = R_1(\omega^2) + j\omega\, I_1(\omega^2)$$

Critical frequencies

- The set of critical frequencies are values of ω such that

$$p(j\omega)=0 \text{ for all } X(-\omega^2)$$

- This set is given by

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_{n_f}\}$$

where ω_i are solutions of the polynomial equation

$$I_0(\omega_i^2)\, R_1(\omega_i^2) \; - I_1(\omega_i^2)\, R_0(\omega_i^2) = 0$$

Number of critical frequencies

- Lemma

  The number of critical frequencies is a polynomial function of n_N, n_D, n_Y, n_Z and n_V

- Remark: Number of critical frequencies does not depend on the number of tractable parameters θ

Stability boundaries

- For each critical frequency we have an hyperplane which defines part of the stability boundary
- Each hyperplane has the form

$$\psi_0(\omega_i)\,\theta_0 + \psi_2(\omega_i)\,\theta_2 + \ldots + \psi_{n_\theta}(\omega_i)\,\theta_{n_\theta} = \upsilon(\omega_i)$$

- In vector form we write

$$\psi_i = [\ \psi_0(\omega_i)\ \ \psi_2(\omega_i)\ \ldots\ \psi_n(\omega_i)\ ]$$
$$\upsilon = [\ \upsilon(\omega_i)\ \upsilon(\omega_i)\ \ldots\ \upsilon(\omega_i)\ ]^T$$

and

$$\Psi = [\ \psi_1\,\psi_2 \ldots \psi_{n_\theta}\ ]^T$$

IEIIT-CNR

LP solution

- Use of linear programming to determine a stabilizer
- Since previous equations define only part of the stability boundary, this results into a combinatorial problem
- We need to consider *inequality* constraints rather than equalities

$$\psi_0(\omega_i)\,\theta_0 + \psi_2(\omega_i)\,\theta_2 + \ldots + \psi_n(\omega_i)\,\theta_n \leq \upsilon(\omega_i)$$
$$\psi_0(\omega_i)\,\theta_0 + \psi_2(\omega_i)\,\theta_2 + \ldots + \psi_n(\omega_i)\,\theta_n \geq \upsilon(\omega_i)$$

- Drawback: worst case number of linear programs $N_{LP}(n_f)$ is exponential

IEIIT-CNR

Algorithms for fixed order stabilization

- We propose a different approach
- Use of *vertices* of the stability boundary insead of hyperplanes
- This method is based on two stages

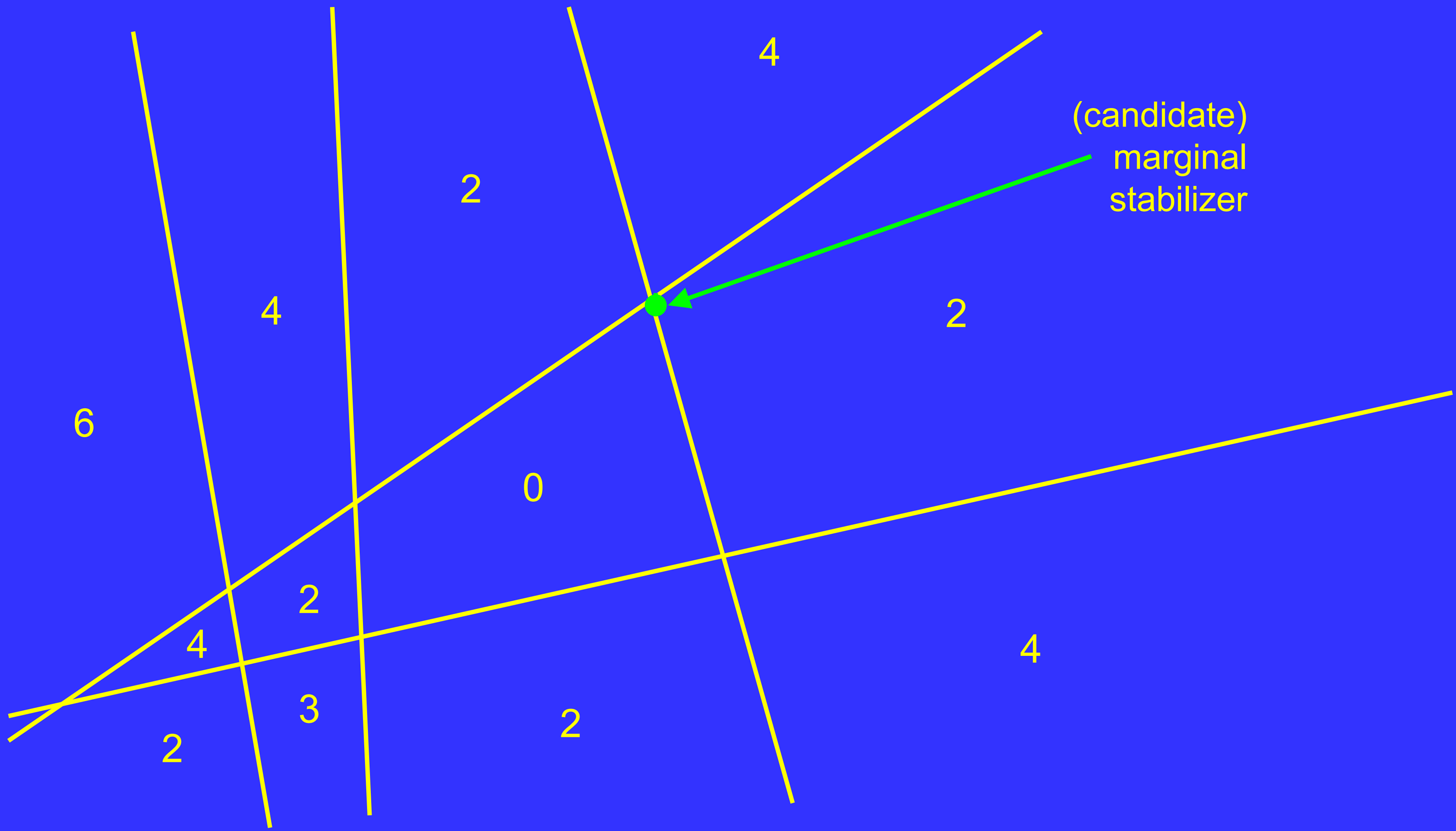
  - computation of ***marginal stabilizer***

  - computation of ***fixed order stabilizer***

- First step is polynomial-time
- Second step requires one-parameter optimization

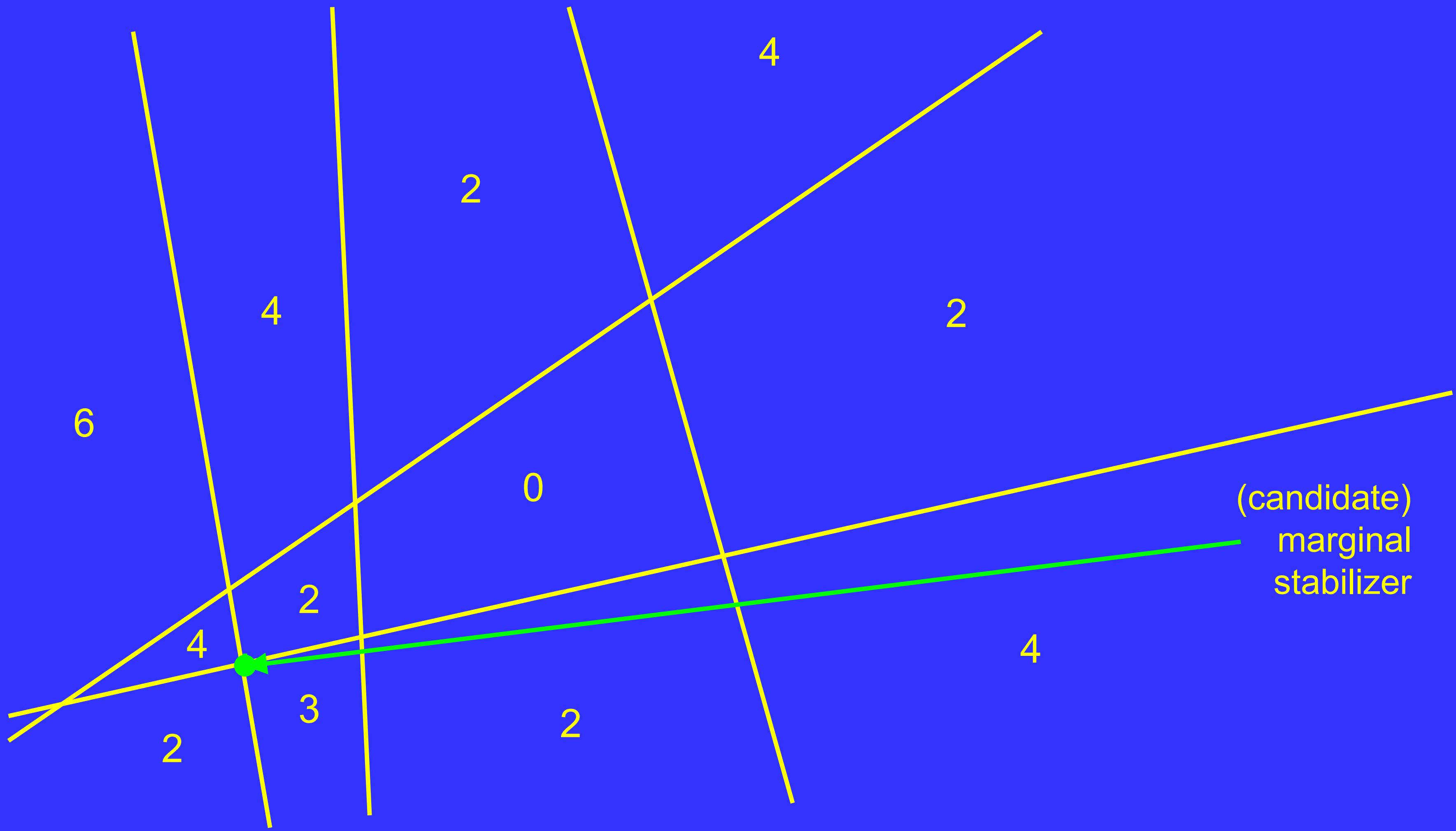
IEIIT-CNR

(Candidate) marginal stabilizer



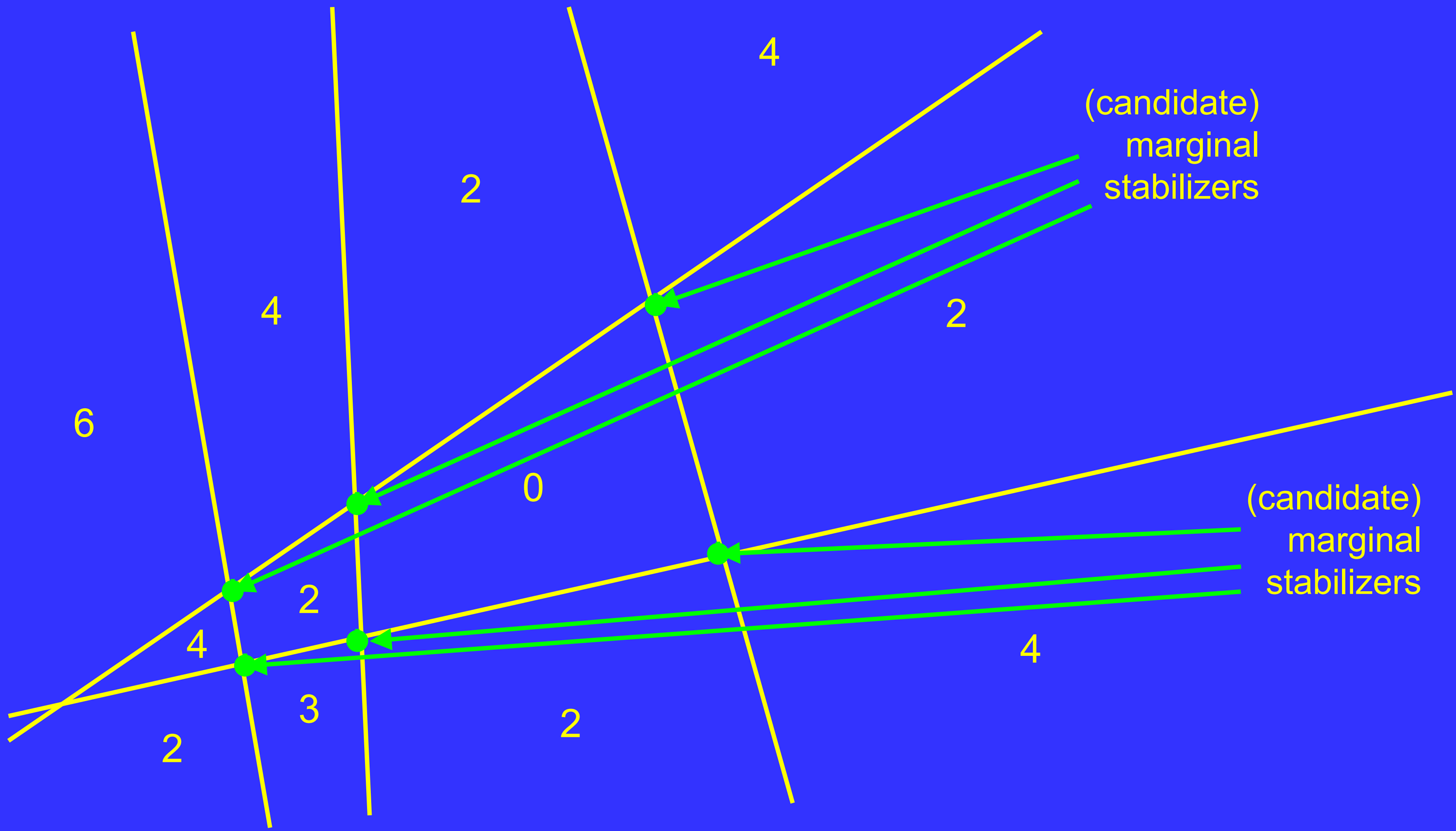
IEIIT-CNR

(Candidate) marginal stabilizer



IEIIT-CNR

All (candidate) marginal stabilizers



Marginal stabilizer

- <u>Definition</u>

  A marginal stabilizer is a controller $C(s,\theta)$ with the property that closed-loop polynomial $p(s,\theta)$ has a fixed number of roots on the imaginary axis and no roots in the open right half plane

IEIIT-CNR

Marginal stabilizer computation

- Recall that we have a matrix Ψ and a vector υ
- Solution of the combinatoric problem using vertices of the stability boundary
- Construct a square matrix $\underline{\Psi}_i$ with n rows of Ψ and a vector $\underline{\upsilon}_i$ with n corresponding elements of υ
- Obtain a linear system of the form

$$\underline{\Psi}_i \,\theta = \underline{\upsilon}_i$$

- Marginal stabilizer is immediately given by

$$\theta^{(i)} = \underline{\Psi}_i^{-1} \underline{\upsilon}_i$$

- Number of vertices i of stability boundary is $N_{MI}(n_f, n_\theta)$

IEIIT-CNR

Invertibility condition

- Suppose that $n_f \geq n_\theta$

- Lemma

  The square matrices $\underline{\Psi}_i$ are full rank for $i=1, 2, \ldots, N_{MI}$

- Proof based on the Vandermonde-like structure of the matrix Ψ

IEIIT-CNR

Marginal stabilizer computation

- For fixed i, $\theta^{(i)}$ is found by matrix inversion
- This requires O($n_\theta{}^3$) operations
- Once a *candidate* marginal stabilizer $\theta^{(i)}$ is computed, we check with Routh test if it is a marginal stabilizer
- Routh test has polynomial (quadratic) complexity
- Worst case number of matrix inversions is given by

$$N_{MI}(n_f, n_\theta) = n_f!/(n_\theta!\,(n_f - n_\theta)!)$$

Number of matrix inversions

- Suppose that $n_f \geq n_\theta$
- <u>Theorem</u>

1. We have

$$N_{MI}(n_f, n_\theta) = \mathrm{O}(n_f^{\,n_\theta})$$

2. Furthermore, we obtain

$$N_{MI}(n_f, n_\theta) \leq N_{LP}(n_f)$$

where equality is attained only if $n_f = 0$

- <u>Proof</u> is based on the so-called bimodal theorem and properties of the factorial

Comments

- Study fixed order stabilization, hence the number n_θ of controller parameters θ is fixed
- We have $N_{MI}(n_f, n_\theta) = O(n_f^{\,n_\theta})$
- Polynomial-time complexity in the number of critical frequencies and, consequently, in n_N, n_D, n_Y, n_Z and n_V
- Number of worst case matrix inversions is *much smaller* than the number of LPs, see next table

Comparison of N_{LP} and N_{MI}

n_f	18	16	32	64
$N_{MI}(n_f,2)$	28	120	496	2,016
$N_{MI}(n_f,4)$	70	1,820	35,960	$6.8 \cdot 10^5$
$N_{LP}(n_f)$	256	65,536	$4.2 \cdot 10^9$	$1.8 \cdot 10^{19}$

Theorem

Let $p(s,\theta^{(i)})$ be the closed-loop polynomial corresponding to tractable parameter $\theta^{(i)}$

- Theorem

  There exists a marginal stabilizer if and only if there exists $\theta^{(i)}$, $i = 1, 2 \ldots, N_{MI}$, such that $p(s,\theta^{(i)})$ has its zeros within the closed left half plane

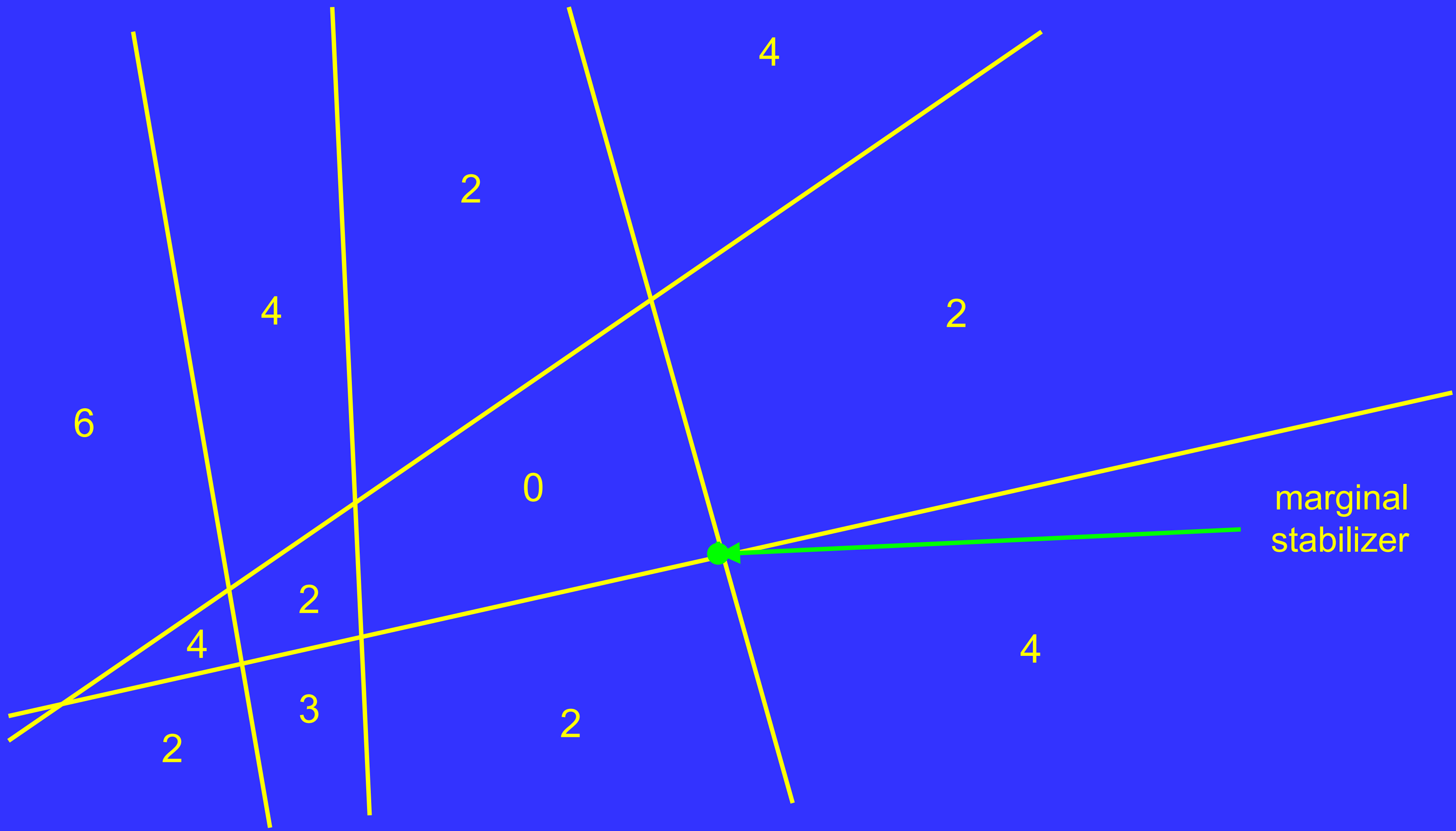
- Proof easily follows from previous discussions

Vertices of stabilizing controller set

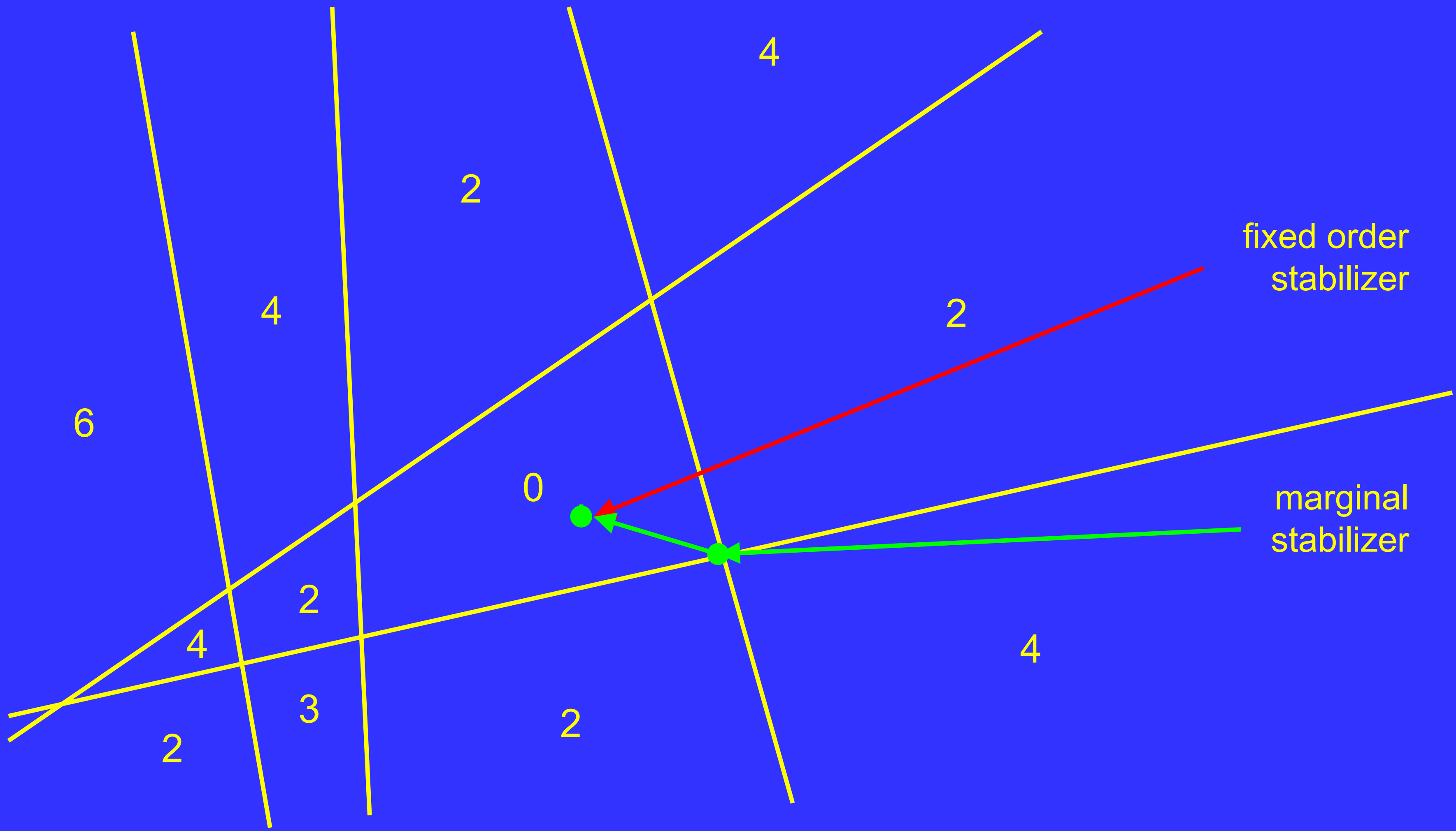
- The controller parameter vector $\theta^{(i)}$, if it exists, is a vertex of a polyhedral set of stabilizing controllers
- The n_θ rows of the matrix $\underline{\Psi}_i$ and the n_θ elements of Ψ_i and of υ_i define some of the hyperplanes of the boundary of a polyhedral set of stabilizing controllers

IEIIT-CNR

Marginal stabilizer



Fixed order stabilizer



IEIIT-CNR

Computation of stabilizing controller

- Once a marginal stabilizing controller is determined, we need to compute a fixed-order stabilizing controller
- Sensitivity method for the zeroes of $p(s,\theta)$ for perturbations of θ
- Suppose that θ is a marginal stabilizer so that $p(s,\theta)$ has k simple roots on the imaginary axis (and the remaining in the open left half plane)
- Study how the root $j\omega_i$ moves when perturbing θ by $\Delta\theta$
- Consider the analytic function $z_i = z_i\,(\Delta\theta)$

Computation of stabilizing controller

- Use implicit function theorem to obtain $\partial z_i / \partial \theta_k$
- Want to find $\Delta\theta$ moving all imaginary zeroes inside the open left half plane
- This leads to the solution of a linear system of the kind

$$A\,\Delta\theta = b$$

where A and b are given data and

$$\Delta\theta = [\Delta\theta_0\ \Delta\theta_2\ \Delta\theta_4\ \dots]^T$$

- If A is full rank, we can immediately compute $\Delta\theta$
- Consider a parameter $\theta^{(j)} + \alpha\,\Delta\theta$, where $\alpha > 0$
- Obtain the optimal α using bisection

Algorithm 3

1. construct Ψ and υ for given $\eta^{(i)}$
2. for $j := 1,\ldots, N_{MI}(n_f,n_\theta)$ do
   begin
3. compute $\theta^{(j)}$ with matrix inversion
4. if $\theta^{(j)}$ is a marginal stabilizer then
   begin
5. compute $\Delta\theta$ with matrix inversion
6. if a stabilizing parameter $\theta^{(j)}+\alpha\Delta\theta$ is found then stop
   end
   end

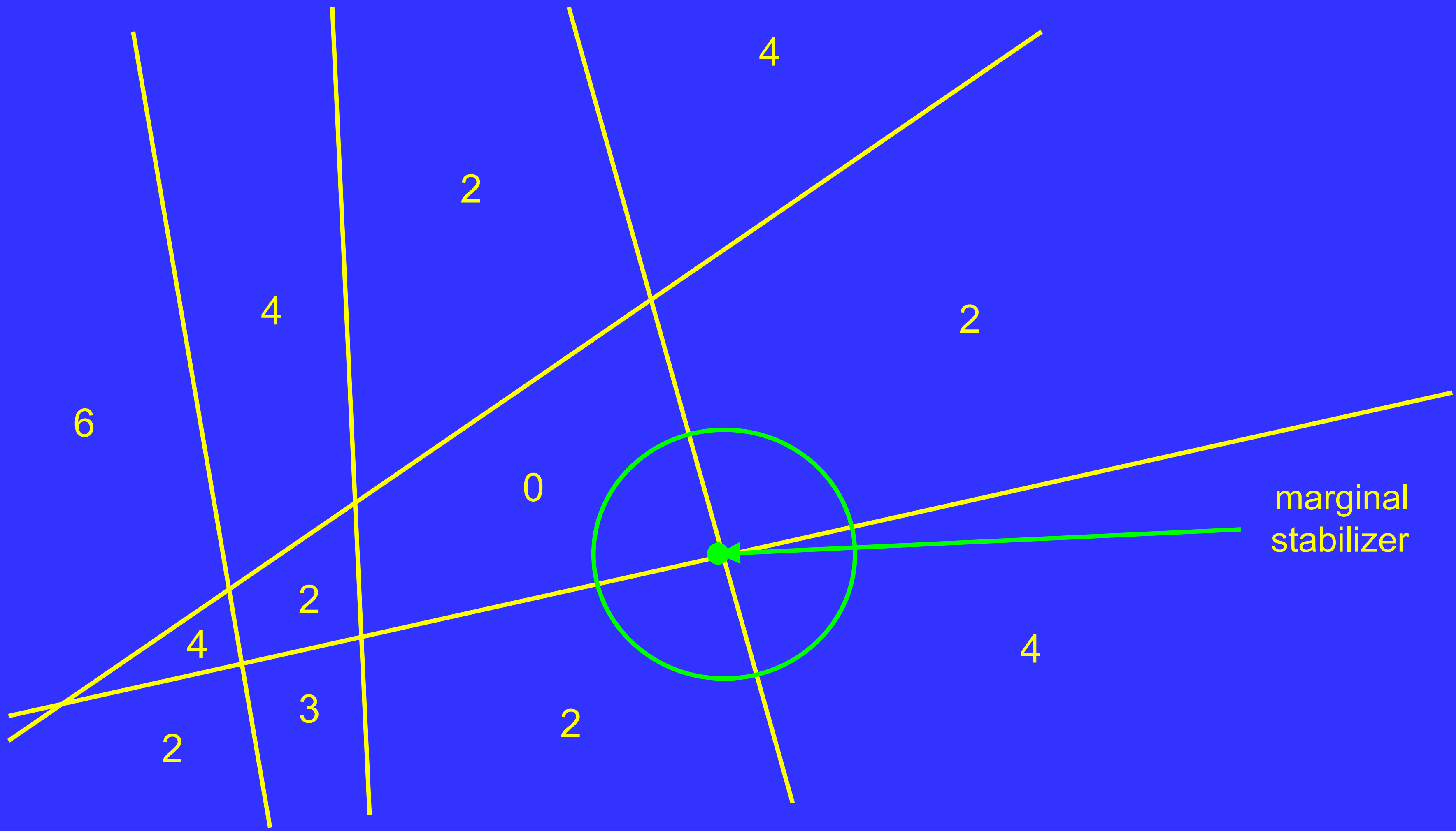
IEIIT-CNR

Another approach

- The sensitivity method requires two conditions
  - zeroes of $p(s,\theta)$ need to be simple
  - a full rank condition should be satisfied
- If these conditions are not met we use randomization
- Take a ball (for example Euclidean) of radius $r>0$, generate N samples within the ball until we find a fixed order stabilizer
- This procedure is guaranteed to converge because a stabilizer exists within the ball

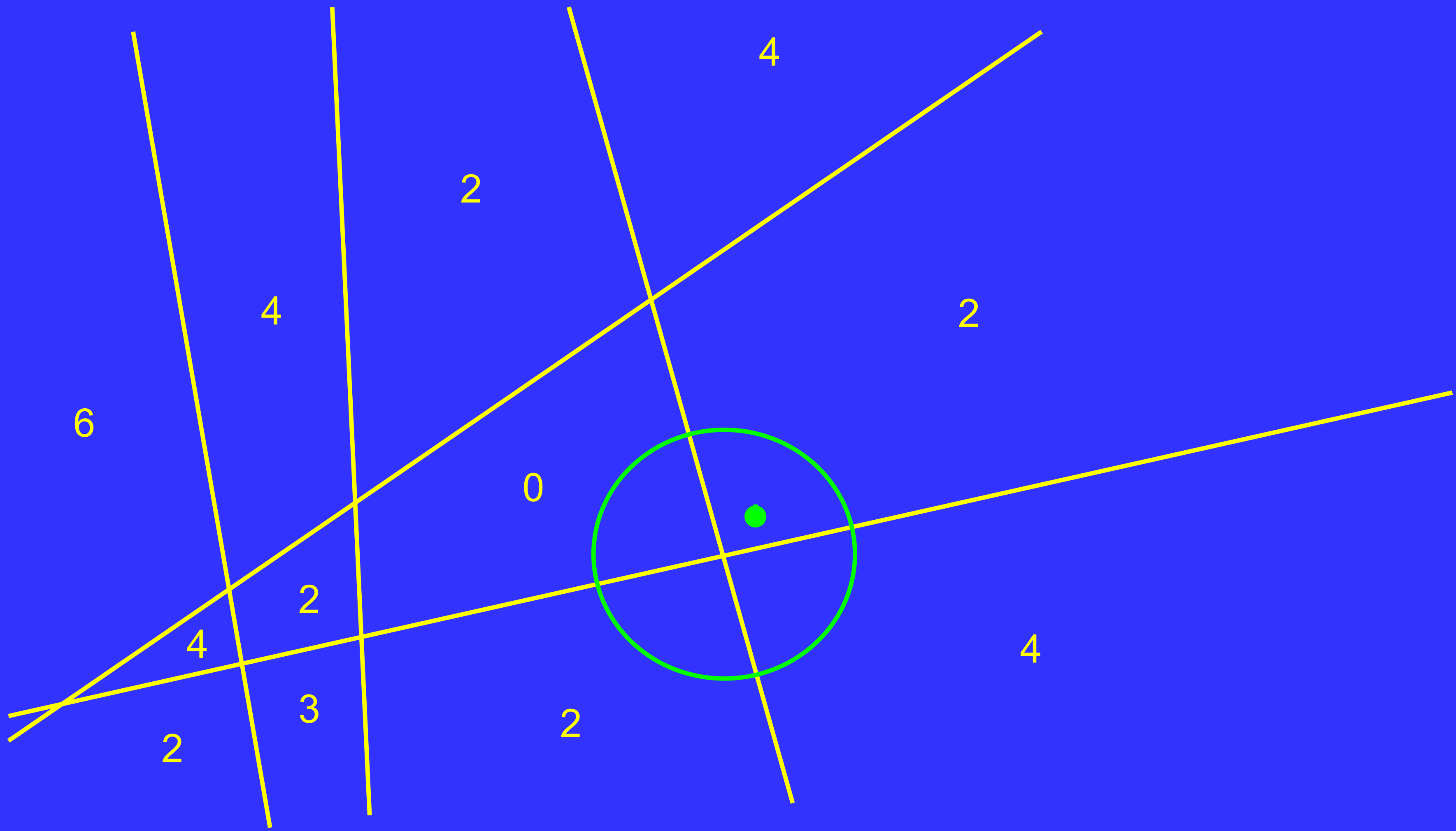
Randomization again…

IEIIT-CNR

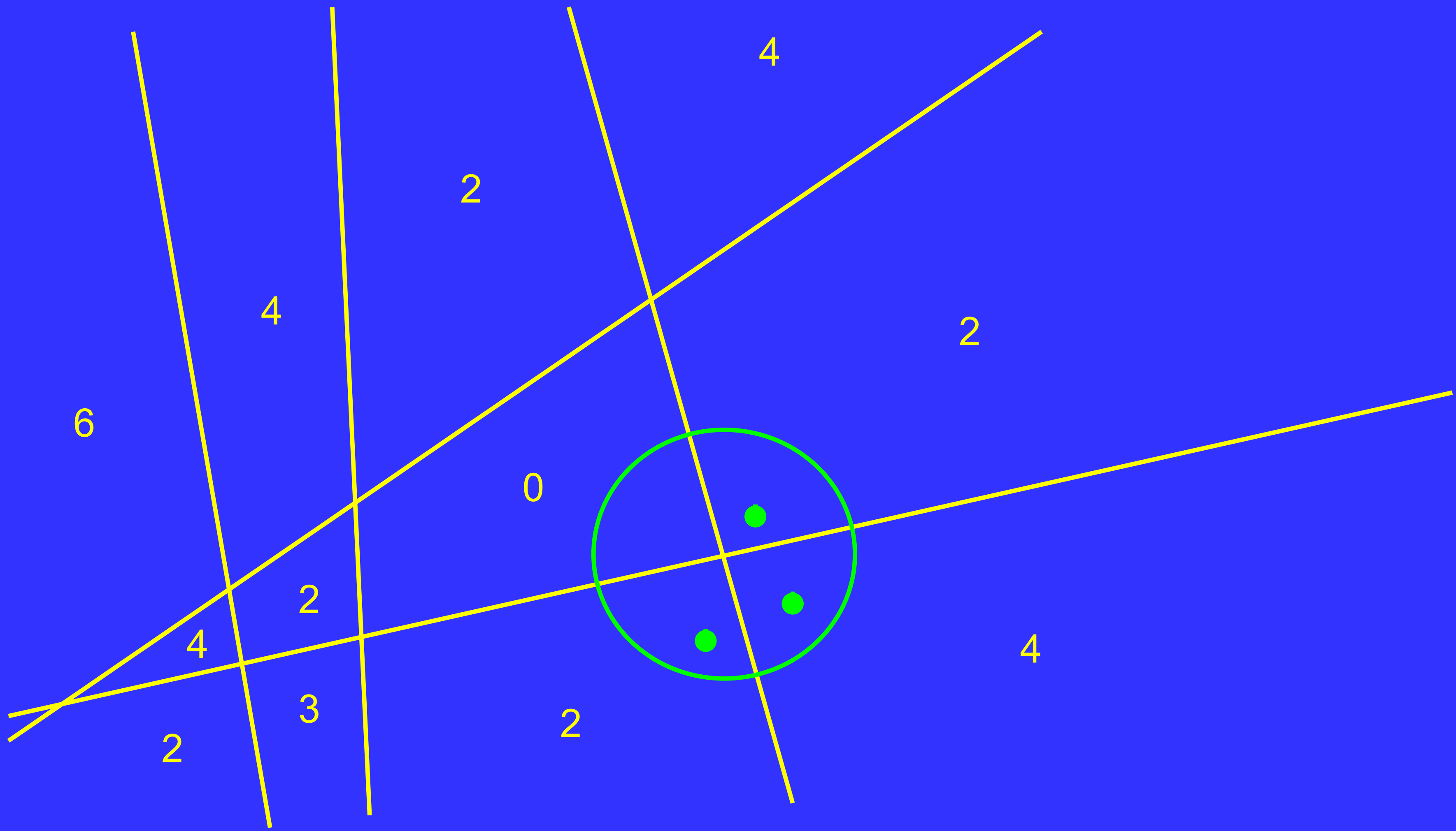


IEIIT-CNR

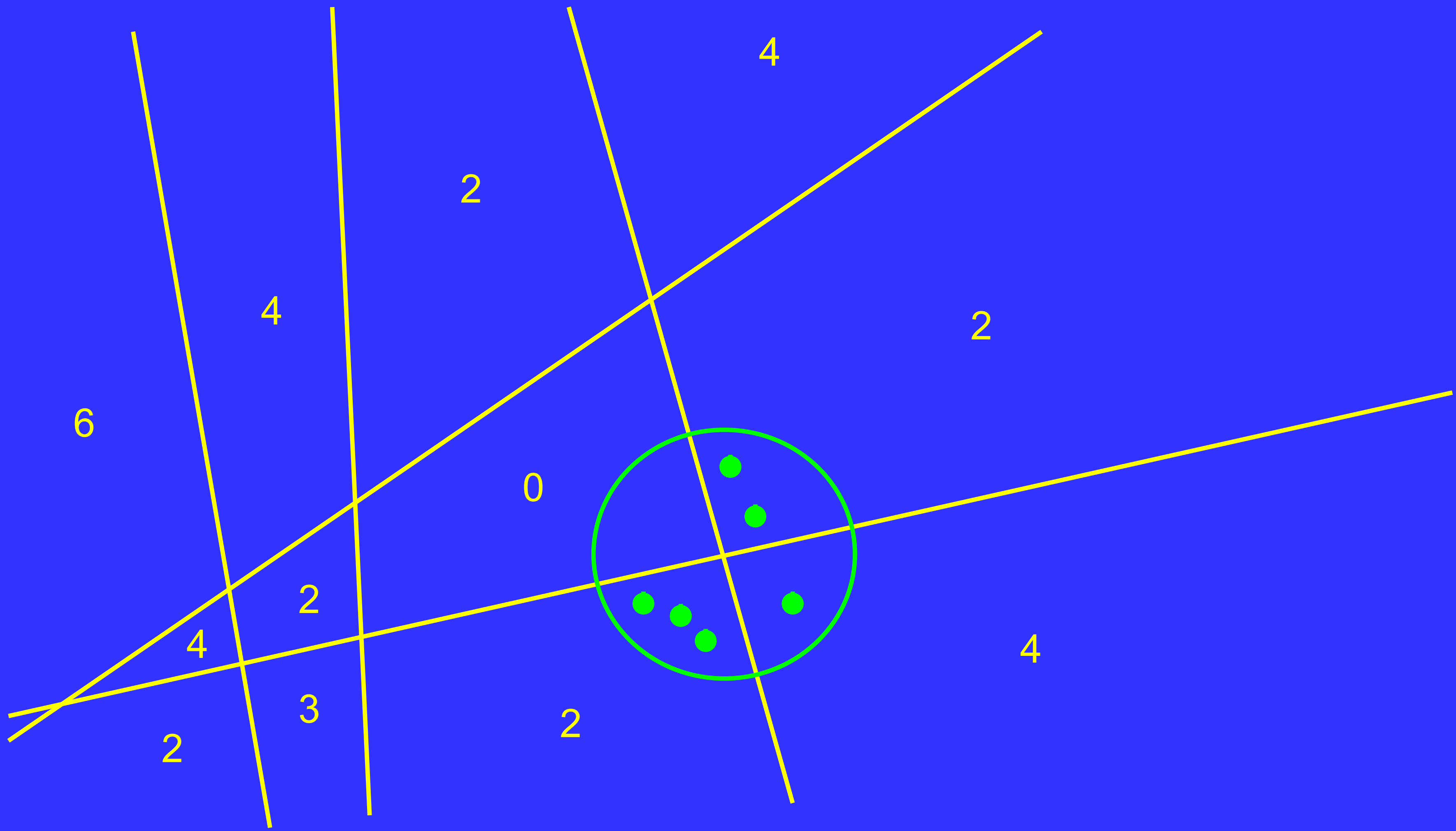
Randomization again…



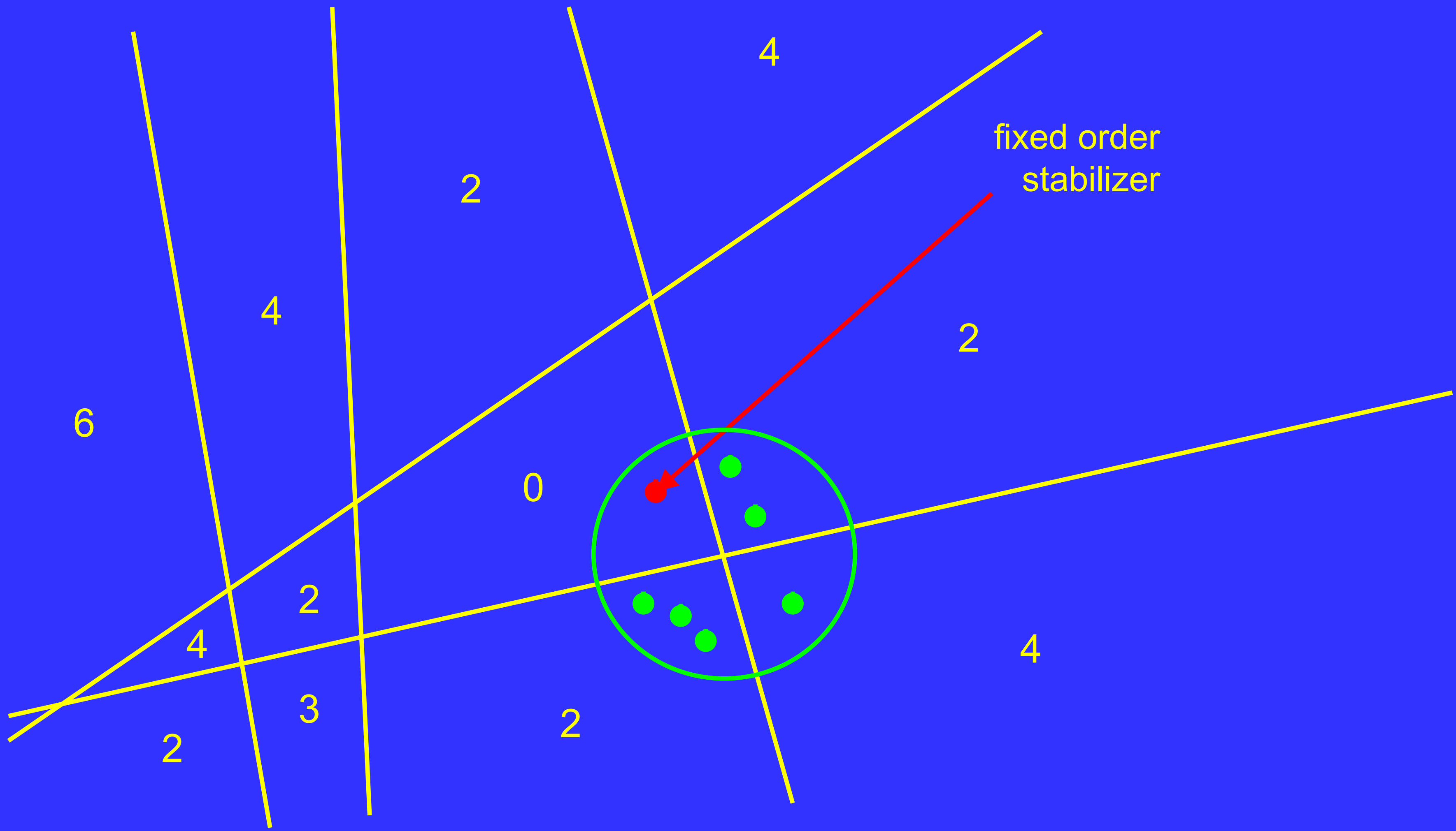
Randomization again…



Randomization again…



Fixed order stabilizer



Example: stabilization

- Unstable plant of order 4 and second order controller

- $N_P(s) = s\,(1 + 2\,s + s^2)$
- $D_P(s) = 1 + s + s^2 + s^3 + s^4$
- $N_C(s) = X(s^2) + sY(s^2) = \theta_0 + \alpha_0\,s + \theta_2\,s^2$
- $D_C(s) = Z(s^2) + sV(s^2) = \beta_0 + \mu_0\,s + \beta_2\,s^2$

IEIIT-CNR

Tractable/intractable parameters

$\theta = [\theta_0 \;\; \theta_2]^T$ ***tractable parameters***

$\eta = [\alpha_0 \, \beta_0 \;\; \beta_2 \; \mu_0]^T$ ***intractable parameters***

- Set confidence and accuracy to

$$\varepsilon = 0.01 \text{ and } \delta = 0.01$$

and compute

$$N_1 = 459$$

IEIIT-CNR

Uniform random generation

- Randomly uniformly generate β_2 and μ_0 in the interval [0,5]
- Set $\beta_0 = 1$ to obtain a stable controller
- Since we may need a “non-minimum phase" controller, we randomly uniformly generate α_0 in [-5,5]

Running the algorithms

- We obtained intractable parameters

$$\eta = [3.6095\ 1\ 4.5845\ 3.3292\ ]^T$$

- We computed two critical frequencies

$$\Omega = \{0.4059,\ 1.2697\}$$

- Marginal stabilizing (tractable) controller parameters are obtained solving a linear equation

$$\theta = [-2.3082\ -0.3063\ ]^T$$

IEIIT-CNR

Stabilizing controller

- A marginal stabilizing controller is

$$N_C(s) = -2.3082 + 3.6095\, s - 0.3063\, s^2$$
$$D_C(s) = 1 + 3.3292\, s + 4.5845\, s^2$$

- Then, we computed the desired direction of the perturbation

$$\Delta\theta = [13.2513\ 15.3654]^T$$

- With the step parameter α = 0.1, we obtained

$$\theta + \alpha\,\Delta\theta = [\,-0.9831 \quad 1.2303\,]^T$$

- A stabilizing controller is given by

$$N_C(s) = -0.9831 + 3.6095\, s + 1.2303\, s^2$$
$$D_C(s) = 1 + 3.3292\, s + 4.5845\, s^2$$

Extensions: $\mathcal{H}_\infty$ performance

$\mathcal{H}_\infty$ performance

- $\mathcal{H}_\infty$ performance of sensitivity/complementary sensitivity

$$S(s) = \frac{1}{1 + P(s)\, C(s)}$$

$$T(s) = 1 - S(s)$$

- Consider also a (stable) weighting function

$$W(s) = N_W(s)/D_W(s)$$

Set of $\mathcal{H}_\infty$ controllers

- Suppose that an intractable parameter vector η is selected
- Theorem

The set θ of all tractable parameter vectors θ giving an $\mathcal{H}_\infty$ controller satisfying

$$\| W(s)\, S(s) \| \leq 1$$

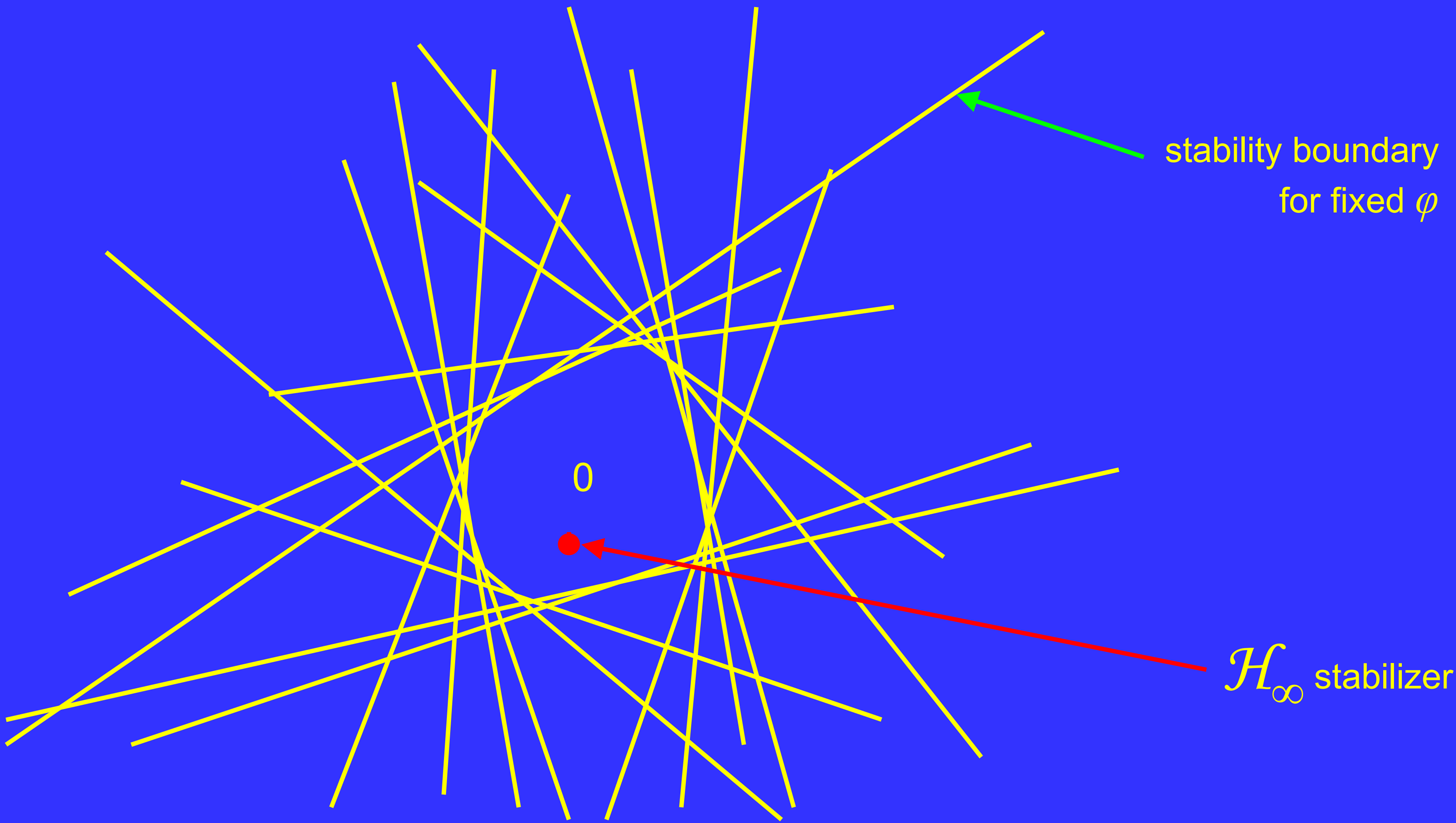
is either empty or is given by

$$\theta = \bigcap_{\varphi \in [0,2\pi)} \Gamma(\varphi)$$

where $\Gamma(\varphi)$ is union of finite number of polyhedral sets for fixed φ

IEIIT-CNR

(Convex) set of $\mathcal{H}_\infty$ controllers



IEIIT-CNR

Comments

- Proof makes use of transformation of $\mathcal{H}_\infty$ problem into a stability problem with the additional parameter φ
- The set θ is the union of *convex sets* (not necessarily polyhedral)
- This result is a minor extension of [Saeki, Aimoto, 00; Ho, 03; Blanchini, Lepsky, Miani, Viaro, 04]

Computation of critical frequencies

- Consider a closed-loop polynomial which depends on the additional parameter φ

$$p(s,\varphi) = p_0(s,\varphi) + p_1(s,\varphi)\, X(s^2)$$

- Setting $s = j\omega$ for fixed $\varphi \in [0, 2\pi)$, the critical frequencies are values of ω such that

$$p(j\omega,\varphi) = 0 \text{ for all } X(-\omega^2)$$

IEIIT-CNR

Computation of critical frequencies

- **Lemma**

  For fixed $\varphi \in [0, 2\pi)$, the critical frequencies are given by the solution of the equation

  $$f_0(\omega) + \sin\varphi\, f_s(\omega) + \cos\varphi\, f_c(\omega) = 0$$

  where $f_0(\omega), f_s(\omega)$ and $f_c(\omega)$ are given polynomials

- **Proof** based on lengthy but straightforward computations

Algorithm

Algorithm 4

```
1.  set ε > 0, n_φ, Δn_φ, and n^u
2.  while n_φ ≤ n^u do
     begin
3.      construct ψ and ν for given η^(i)
4.      for j := 1,…, N_MI(n_f, n_θ) do
         begin
5.          compute θ^(j) by matrix inversion
6.          if θ^(j) gives an ε-almost stabilizer then
            begin
7.              compute Δθ by matrix inversion
8.              if a stabilizing parameter θ^(j) +αΔθ is found then stop
            end
         end
9.      n_φ := n_φ + Δn_φ
     end
```

Comments

- Comments similar to those made for Algorithm 3 can be stated here
- Algorithm 4 is based on combination of matrix inversions and sensitivity methods
- Main difference with Algorithm 3 if the presence of the additional parameter φ
- Grid points for this parameter are needed
- This increases the number of matrix inversions, which is a function of the number of grid points

Extensions: stabilization of interval plants

Interval plants

- SISO strictly proper *interval* plant

$$P(s,q,r) = \frac{N_P(s,q)}{D_P(s,r)}$$

where $N_P(s,q)$ and $D_P(s,r)$ are interval polynomials

- Fixed order controller $C(s)$

IEIIT-CNR

One-parameter stabilization problems

- Transform the fixed-order stabilization problem with interval plant into a number of one-parameter stabilization problems of the form

$$p(s,\lambda) = N_{i_1}(s)\left(X(s^2) + sY(s^2)\right) + D_{i_2,i_3}(s,\lambda)\left(Z(s^2) + sV(s^2)\right)$$

and

$$D_{i_2,i_3}(s,\lambda) \;= \lambda\, D_{i_2}(s) + \left(1\text{-}\ \lambda\, D_{i_2}(s)\right)$$

where $N_{i_1}(s)$, $D_{i_2}(s)$, $D_{i_2}(s)$ are fixed polynomials associated to the interval plant and $\lambda \in [0,1]$ is a parameter

IEIIT-CNR

Comments

- Number of one-parameter stabilization problems is 32
- Consider a closed-loop polynomial which depends on the additional parameter λ

$$p(s, \lambda) = p_0(s, \lambda) + p_1(s, \lambda)\, X(s^2)$$

- Critical frequencies can be computed solving a quadratic equation in λ
- Approach similar to $\mathcal{H}_\infty$ stabilization, details are tedious
- Polynomial-time algorithm can be designed

Conclusions

IEIIT-CNR

Conclusions

- A number of years ago, closed-form meant writing down a “good looking” equation on a piece of paper
- Due to the increasing computational power, the notion of closed-form solution has changed substantially
- Various experts (rightfully) convinced us that a new notion of closed-form is to re-write (if possible) a control problem as convex optimization

IEIIT-CNR

Conclusions

- Unfortunately, many important control problems are not convex
- In such cases, we can either introduce relaxation or use randomization
- Perhaps the control community should pay more attention to the latter option accepting a different and weaker notion of problem solution