Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio, Patras, GREECE 17-19 July, 2000

A NEW APPROACH TO ESTIMATE CONTROLLABLE AND RECOVERABLE REGIONS FOR SYSTEMS WITH STATE AND CONTROL CONSTRAINTS

C.A. YFOULIS *, A. MUIR , P.E. WELLSTEAD

Control Systems Centre, UMIST, P.O.Box 88, Manchester,
M60 1QD, U.K., p.wellstead@umist.ac.uk

Abstract. In this paper the problem of estimating controllable and recoverable regions for classes of nonlinear systems in the presence of uncertainties, state and control constraints is considered. A new computational technique is proposed based upon a ray-gridding idea in contrast to the usual gridding techniques. The new technique is also based on the positive invariance principle and the use of Piecewise Linear (PL) Lyapunov functions to generate polytopic approximations to the controllable/recoverable region with arbitrary accuracy. Various types of stabilising controllers achieving certain trade-offs between robustness, performance and safety, while respecting state and control constraints, can be easily generated. The technique allows the approximation of nonlinear systems via *piecewise linear uncertain* models which reduces the conservatism associated with *linear uncertain* models.

Key Words. Controllable region, recoverable region, PL-Lyapunov functions.

1 INTRODUCTION

The presence of control input bounds is unavoidable from a practical point of view due to the use of actuators that saturate. Moreover, state bounds are natural in many practical problems, since the state variables -which usually correspond to physical quantities of interest- are only allowed to take values within certain intervals. For such constrained systems *controllability* and *stabilisability* are fundamental issues. It is very useful to provide a characterisation of those states which can be controlled to the origin -the trivial equilibrium point- by means of admissible controls. Such states are called *controllable* [5, 6]. The *controllable region* is defined then as the set of all controllable states. In the presence of additional state limitations, all states that can be driven to the origin by means of bounded controls while respecting the state constraints are called *recoverable* [7]. All recoverable states constitute the *recoverable region.*

*This research was supported by the EPSRC under grant GR/K 36300 and studentship Ref. No. 97700206 for the first author. The support of the UMIST graduate research fund is also acknowledged.

The estimation of the controllable or the recoverable region can be valuable, especially when the designer wishes to take such limitations into account in the control design stage in order to find a control law that not only stabilises a system - and possibly achieves good performance- but also guarantees an almost maximal region of attraction. Moreover, if there exist tools to measure performance and safety degrees, the control design can aim at certain trade-offs between them.

There has been a significant activity in the past few years on the control of linear systems subjected to control bounds. Some analytical results characterising the controllable regions of anti-stable systems (all poles in the open right half of the complex plane) were given in [5, 6], in which references to other interesting results on global or semi-global stabilisation can also be found.

On the other hand, there has been a considerable research activity for discrete and continuous-time linear time invariant (LTI) systems subjected to both state and control constraints (see [4, 7] and references therein). Unfortunately, all these results are applicable to LTI systems only, except of the work of Blanchini [1, 2, 3] which can deal with linear uncertain systems (in a parametric uncertainty form). His method yields polytopic approximations of the recoverable region with arbitrary accuracy via the use of PL Lyapunov functions and the corresponding polytopic positively invariant (P.I.) sets. A set S is called *positively invariant* with respect to the trajectories of a dynamical system when all trajectories initiated in it never escape it. A different approach is presented in [7]. Our work is most closely related to Blanchini's ideas although a completely new computational technique is proposed. The proposed technique is named *ray-gridding* and is capable of getting progressively better estimates of the controllable or the recoverable region based on a gridding of the state-space in terms of rays and not points. An advantage is the ability to yield intermediate results which represent trade-offs between execution time and quality of approximation. These trade-offs, when seen from the control design viewpoint can be interpreted as trade-offs between performance and safety and can be used for more complete and systematic constrained stabilisation. Finally, not only linear uncertain but also PL uncertain systems (in a form consistent to the state-space partition imposed by the polytopic approximation) can be dealt with.

The paper is organised as follows: The basic underlying idea is described in section 2.1 for planar LTI systems and a technique that reduces the computational time significantly is outlined in section 2.2. The technique is implemented successfully to some planar examples from [7] in section 3. In section 4 all possible extensions of the technique to more useful system classes are discussed. Due to space limitations, only sketches of some proofs are given and the reader is referred to [8] for more details.

2 THE RAY-GRIDDING APPROACH

The *ray-gridding* approach has been conceived especially to provide a flexible framework in which one has the ability to experiment with different types of PL-Lyapunov functions, by adjusting parameters of their polytopic surfaces such as position and complexity.

Let us assume a planar single-input LTI system

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{b}\,u\ , \quad \mathbf{A} \in \mathbb{R}^{2\times 2}\ ,\ \mathbf{b} \in \mathbb{R}^{2\times 1} \quad (1)$$

Instead of gridding which operates pointwise, the ray-gridding operates in terms of rays.

Definition 1. By a **ray partition** in $\mathbb{R}^2$ we denote a set $\{r_i\ ,\ i = 1, 2, \ldots, n_r\}$ of rays, where $r_i = \{\mathbf{x} \in \mathbb{R}^2\ :\ \mathbf{x} = \lambda_i \cdot \mathbf{e}_i\ ,\ \lambda_i \geq 0\ ,\ \mathbf{e}_i \in \mathbb{R}^2\}$. The vectors $\mathbf{e}_i \neq \mathbf{0}$ which specify the rays are termed **ray vectors** and then any point on the ray r_i is uniquely determined by the non-negative scalar λ_i, referred to herein as its **scaling factor**. The number n_r of the rays in $\{r_i\}$ is called the **order** of the ray partition.

Definition 2. We also define the following special subclasses of the class of all ray partitions in $\mathbb{R}^2$:

- A ray partition is called **proper** when all its rays r_i are disjoint, or $r_1 \cap r_2 \cap \ldots \cap r_{n_r} = \{\mathbf{0}\}$, i.e. they only intersect at the origin. Otherwise it is called **improper**.
- A ray partition is called **unit** if and only if all its ray vectors are unit vectors, i.e. their magnitude is equal to 1.
- A ray partition is called **constrained** when all its scaling factors are bounded, i.e. there exist upper limits λ_i^+ such that $0 \leq \lambda_i \leq \lambda_i^+$.
- A constrained ray partition is called **normalised** if and only if all its scaling factors are bounded by 1, i.e. $\lambda_i^+ = 1\ \forall i$.
- A ray partition is termed **symmetric** if and only if for every ray r_i there exists another ray r_j such that $\forall \mathbf{x} \in r_i\ \exists \mathbf{x}' \in r_j\ :\ \mathbf{x}' = -\mathbf{x}$.

Remark 1. State and control constraints of the form $\mathbf{x}(t) \in \mathcal{X}$ and $\mathbf{u}(t) \in \mathcal{U}$, where $\mathcal{X}$,$\mathcal{U}$ are compact, convex and containing the origin in their interior, are assumed. In their presence, a ray partition becomes constrained, and it can be normalised by using as new ray vectors $\mathbf{e}_i'$ the intersection of the rays and the boundary of the state constraint set $\mathcal{X}$, i.e. $\mathbf{e}_i' = r_i \cap \partial\mathcal{X}$.

Definition 3. By a **controlled invariant polytope** P we refer to a polytope that can be made positively invariant by an appropriate choice of admissible controls, i.e. $\forall \mathbf{x} \in P \quad \exists u(\mathbf{x}) \in \mathcal{U}$ s.t. $\mathcal{D}^+ V_P(\mathbf{x}) < 0$. $V_P(\mathbf{x})$ denotes the PL-Lyapunov function induced by a polytope P and $\mathcal{D}^+ V_P(\mathbf{x})$ is the right Dini-derivative used for checking invariance for non-smooth functions (see [2]).

Figure 1 shows a certain *ray partition* $\{r_i, i = 1, 2, 3, 4\}$. There is an infinite number of polytopes formed by joining vertices on the rays, each possessing four faces (edges) only. Let us consider all *convex* polytopes that can be generated by this ray partition of the state-space. Two of them and their convex hull are shown in Figure 1. One natural question to ask is then : “Is there a biggest such controlled invariant polytope and if there is one, how can it be calculated?” The following proposition states there is such a polytope:

Proposition 1. *Assume a constrained ray partition $\{r_i\}$ in $\mathbb{R}^2$ for a planar continuous-time LTI system $(\mathbf{A}, \mathbf{b})$. Then there exists a maximal controlled invariant polytope generated by it, which* *is the convex hull of all controlled invariant polytopes, or equivalently, the polytope specified by the maximum scaling factors along all rays.* □

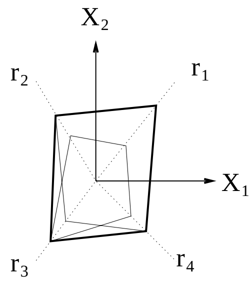

Figure 1: A symmetric ray partition with four rays, two polytopes and their convex hull.

2.1 The basic iterative procedure

The next important issue is how the maximal controlled invariant polytope can be calculated. This is equivalent to finding the largest scaling factors along all rays for which invariance holds. We propose a simple iterative algorithm for calculating it

Step 1 : Consider a ray partition $\{r_i \; i = 1, 2, \ldots, n_r\}$ and the corresponding ordered set of scaling factors λ_i that specify a randomly chosen initial polytope P_0. Transform the ray partition to a constrained and normalised ray partition with state constraints $\mathbf{x}(t) \in \mathcal{X}$ (see e.g. Figure 2). The aim of the iterative algorithm is to find the maximum scaling factors $\lambda_i^* \leq 1$ for which invariance is preserved. These will then specify the convex maximal polytope P_f.

Step 2 : Consider a ray r_i and the vector $\mathbf{v}_i = \lambda_i \cdot \mathbf{e}_i$ specified by the current scaling factor $\lambda_i \leq 1$ and the ray vector $\mathbf{e}_i$. The initial polytope P_0 specifies a value $\lambda_i^{(0)} \leq 1$ which is used as an initial guess. Let λ_i^-, λ_i^+ represent the lower and upper bounds respectively of the unknown maximal λ_i^* we seek. Using a repeated bisection procedure we modify both bounds until they converge to a common value (within a certain accuracy). This is done by using a linear program at every step of the iterative procedure to search for an admissible control that can satisfy the invariance conditions. The bounds are then modified at each step, depending on whether the linear program returns a solution or not. Initially $\lambda_i^- = 0$ and $\lambda_i^+ = 1$. In the first step, $\lambda_i = \lambda_i^{(0)}$ is checked and if a solution exists, then $\lambda_i^- = \lambda_i$, otherwise $\lambda_i^+ = \lambda_i$. Afterwards the procedure operates with $\lambda_i = (\lambda_i^- + \lambda_i^+)/2$. After a number of iterations, the upper and lower bounds will converge to a single number, which will be an estimate of the maximal λ_i^*. The number of iterations is specified by the accuracy selected.

Step 3 : When the repeated bisection procedure has been applied to all n_r rays of the ordered set, a new larger controlled invariant polytope is specified. However, this may not be the maximal one, since the new larger scaling factors λ_i found depend on the neighbouring scaling factors. Thus, an increase in one of them can affect its neighbouring ones and it may allow further increase. A simple way for dealing with this is to repeat the process in a number of cycles, until no further increase is possible. Therefore, Step 2 is repeated until no better results (according to the desired accuracy) are obtained. If some of the scaling factors become zero, this may be interpreted as a sign of infeasibility and a different ray partition with possibly a larger number of rays must be selected.

Proposition 2. *The basic iterative procedure is well posed.* □

Proof: Only a sketch of the proof in [8] is given here. A well-posed iterative procedure is defined to be one for which the repeated bisection is valid, it converges after a finite number of steps to the optimal solution and operates on a necessary and sufficient basis. These can be easily proved for LTI systems, but can be also trivially extended to PL uncertain systems. ■

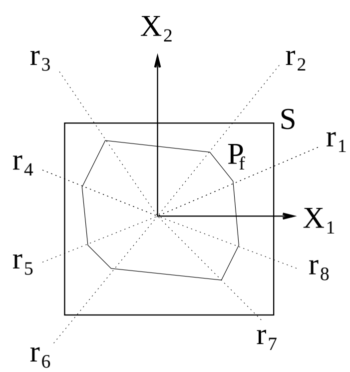

Figure 2: Hard state constraint set S and maximal (P_f) controlled invariant polytope

2.2 Ray-optimal control

Although the validity of the iterative procedure has been proven, experiments show that there are cases in which the execution time required may become significant, e.g. when a large number of cycles to complete the procedure is required or when the initial polytope is not a very favourable selection. This led us to the investigation of ray-optimal control policies which can reduce significantly the computational burden.

Definition 4. By a **ray-optimal** control law for points on a ray r_i we denote a control law $\mathbf{u} = \mathbf{u}(\lambda_i)$ that allows controlled invariance for the biggest possible λ_i. Thus, for a certain ray partition $\{r_i\}$, if the ray-optimal control is applied for all rays, a domain of attraction equal to the maximal controlled invariant polytope is obtained.

It can be shown that there exists a ray-optimal control, which can be specified independently of the other rays.

Proposition 3. *There exists a ray-optimal control for planar LTI systems which is equal to an extreme control value or a saturated linear state feedback for all points on the ray.* □

The ray-optimal control is specified for LTI systems only and can be found prior to running the iterative procedure. The main advantage gained from its use is that repeated bisection and use of linear programs is not necessary. The maximum scaling factor at any step can be found with a single calculation (since with fixed control parameterised expressions of the stability conditions in terms of the scaling factors can be formed, see [8]) and this results in high computational efficiency, as the examples in the next section reveal.

3 EXAMPLES

The previously presented results are illustrated in this section with two planar LTI examples

Example 1:

We consider the first example in [7] with dynamics

$$\dot{x}_1 = x_1 + u \quad , \quad \dot{x}_2 = u \quad , \quad |u| \leq 5 \ , \ |x_2| \leq 5 \tag{2}$$

The maximal controlled invariant polytopes obtained with 16, 32, 64, 128 rays (generated by corresponding symmetric and normalised ray partitions) are shown in Figure 3 (for 4 and 8 rays a solution does not exist). We note that increasingly better approximations are obtained when using a larger number of vertices. For more than 32 vertices only a small increase in the estimated area is obtained. The final approximation with 128 vertices appears to be a very good approximation of the region found in [7]. □

Example 2:

We continue with the second example in [7] with dynamics

$$\dot{x}_1 = 0.5x_1 - x_2 + u \ , \ \dot{x}_2 = x_1 + 0.5x_2 + u \tag{3}$$

and control limitations $|u| \leq 1$. The resulting maximal polytopes with 4,8,16,32,64,128 rays are shown in Figure 4. The controllable region is faithfully approximated when the number of rays is sufficiently increased. □

The areas of the calculated polytopes and the corresponding execution times are shown in Tables 3 and 3 (All examples in this paper were run in MATLAB^{TM} on a personal Laptop computer with a Pentium I, 266 Mhz CPU and 64 Mbs of memory). The acronyms SIP and RIP stand for the Standard and the Ray-optimal based Iterative Procedures, respectively. The number of cycles (iterations of the main loop) required to complete the procedure are given in parentheses. We observe significant computational time savings gained by the use of the ray-optimal control. Furthermore, in both examples a very small increase in the calculated area is achieved for more than 60 rays, thus one can stop there to get a representative picture of the controllable region.

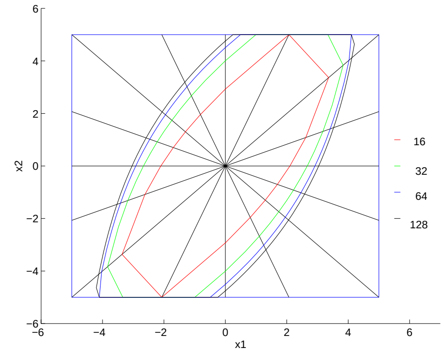

Figure 3: Maximal polytopes with 16,32,64,128 rays (Example 1 in [7])

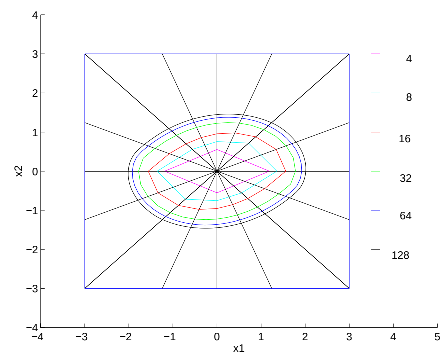

Figure 4: Maximal polytopes with 4,8,16,32,64,128 rays (Example 2 in [7])

Table 1. Example 1 execution time comparison.

# rays	SIP time (secs)	RIP time (secs)	total area
16	38.45 (3)	0.52 (13)	59.99
32	25.16 (4)	0.88 (18)	89.58
64	48.78 (4)	3.24 (34)	102.48
128	71.19 (3)	12.36 (67)	108.01

Table 2. Example 2 execution time comparison.

# rays	SIP time (secs)	RIP time (secs)	total area
4	6.81 (8)	0.00 (5)	2.63
8	18.18 (10)	0.16 (12)	5.39
16	75.69 (21)	0.66 (24)	8.55
32	339.27 (48)	2.31 (46)	13.29
64	1334.6 (96)	8.73 (91)	16.11
128	1990.6 (70)	31.08 (162)	18.10

sufficient, since the refinement method we used (doubling the number of rays uniformly) is by no means optimal. There are many other refinement types which could be useful for minimising the number of rays required for a good approximation.

4 EXTENSIONS OF THE RAY-GRIDDING APPROACH

Further research has shown that many of the ray-gridding approach results can be trivially extended to more general classes of control systems. The same underlying idea can be extended to cover nonlinearities and uncertainties in the dynamics without a significant increase in the computational burden. The technique can be also extended to higher dimensional systems.

4.1 Extension to PL uncertain systems

Let us assume a ray partition $\{r_i \ , \ i = 1, 2, \ldots, n_r\}$ of the state-space of interest in $\mathbb{R}^2$. The rays induce an associated state-space partition into conic sectors $S_i \ , \ i = 1, 2, \ldots, n_r$ which are the conic convex hulls of consecutive rays r_i, r_{i+1}. It is natural to consider this partition as the basis for the approximated PL dynamical system, i.e. the rays are common boundaries between different regions possessing different local dynamics

$$\dot{\mathbf{x}}_i = \mathbf{A}_i \cdot \mathbf{x} + \mathbf{B}_i \cdot \mathbf{u} \quad , \quad i = 1, 2, \ldots, n_r \tag{4}$$

and a *multiple* local linear model is obtained. All significant nonlinearities and uncertainties can be captured in a structured uncertainty parametric form

$$\mathbf{A} = \mathbf{A}^{(n)} + \sum_{j=1}^{n_A} \kappa_j \, \mathbf{E}_j^{(A)} \ , \ \mathbf{B} = \mathbf{B}^{(n)} + \sum_{k=1}^{n_B} \mu_k \, \mathbf{E}_k^{(B)} \tag{5}$$

where the uncertain coefficients take values in bounded intervals

$$\kappa_j \in \left[\kappa_j^-, \kappa_j^+\right] \quad , \quad \mu_k \in \left[\mu_k^-, \mu_k^+\right] \tag{6}$$

The same iterative procedure can be applied then to calculate the controllable/recoverable region. The difference is that an increased set of conditions is imposed due to the variety of different models, but it can be shown that the computational burden does not increase significantly [8][Chapter 6]. The ray-optimal control -for an LTI approximation of system (4)- can now be used to yield initial invariant polytopes only. These are preferred to random initial choices, because they can help in reducing the number of iterations required to complete the procedure.

In addition to stability, performance requirements can be also easily dealt with by specifying a desired bound for the decay rate $\varepsilon > 0$ of the $-\varepsilon V(\mathbf{x})$, where V is our PL-Lyapunov function. This bound can be easily incorporated into the iterative procedure.

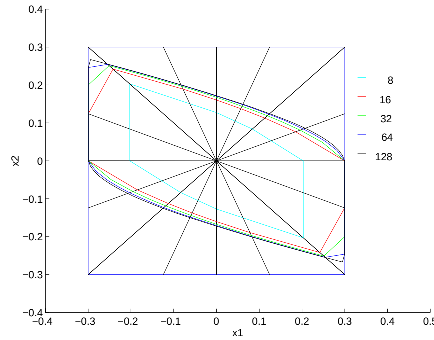

Figure 5: Maximal invariant polytopes with 8,16,32,64,128 rays for the magnetic levitation linear uncertain system ($\varepsilon = 0.01$).

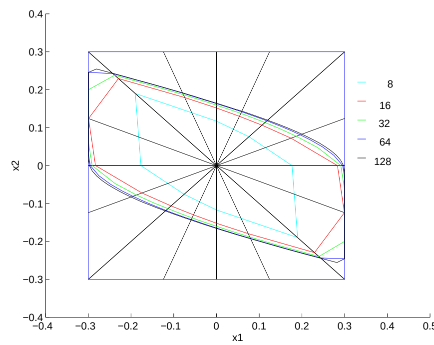

Figure 6: Maximal invariant polytopes with 8,16,32,64,128 rays for the magnetic levitation linear uncertain system ($\varepsilon = 2$).

Table 3. Execution times of the iterative procedure with $\varepsilon = 0.01$ and $\varepsilon = 2$ for the magnetic levitation linear uncertain system.

# rays	$\varepsilon = 0.01$ time	total area 1	$\varepsilon = 2$ time	total area 2
8	9.06 (5)	0.189	7.14 (4)	0.156
16	11.65 (4)	0.344	9.23 (3)	0.316
32	16.15 (3)	0.373	16.81 (3)	0.352
64	21.75 (2)	0.388	42.73 (4)	0.371
128	63 (3)	0.394	85.35 (4)	0.377

4.1.1 A magnetic levitation system

This system has been studied by Blanchini in [1]. It is a highly nonlinear system described by

$$M \cdot \ddot{x}_1 = -k \cdot \frac{u^2}{\left(x_1 + y_0\right)^2} - M \cdot g \tag{7}$$

where x_1 the vertical position, $x_2 = \dot{x}_1$ the vertical speed, u the magnet current , y and y_0 the ball-reference and reference-magnet distances respectively. The system has been modeled in [1] as *linear uncertain* in the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}(w_1(t), w_2(t)) \cdot \mathbf{x}(t) + \mathbf{b}(w_1(t), w_2(t)) \cdot u(t) \tag{8}$$

with

$$\mathbf{A} = \begin{pmatrix} 0 & 100 \\ w_1(t) & 0 \end{pmatrix} \quad , \quad \mathbf{b} = \begin{pmatrix} 0 \\ -w_2(t) \end{pmatrix} \tag{9}$$

and $w_1(t) \in [5, 8]$, $w_2(t) \in [11, 16]$. Strict state and control constraints are also assigned

$$|x_i| \leq 0.3 \quad i = 1, 2 \quad \text{and} \quad |u| \leq 0.6 \tag{10}$$

We proceeded with the uncertain system and different decay rates. For the case of 4 rays it was impossible to find a solution. For 8 rays solutions can be found and the maximum achievable decay rate is $\varepsilon \simeq 2$. The results of the iterative procedure for $\varepsilon = 0.01$ and $\varepsilon = 2$ are shown in Figures 5 and 6 respectively. The corresponding areas and execution times are collected in Table 4.1. Similar results to [1] are obtained.

4.2 Extension to $\mathbb{R}^n, n > 2$

The planar case is special. All polytopes on the plane are simple and simplicial, but this is not the case in higher dimensions. However, it is beneficial to continue working with simplicial polytopes and the ray-gridding technique can be then applied to yield approximations. It is important to mention that in $\mathbb{R}^n, n > 2$, the main difference is that the faces that specify the stability conditions are not fixed. One solution to this problem is to apply convexification at selected stages of the iterative procedure in order to assist the procedure to yield better estimates. Details of how this is achieved are given in [8][Chapter 6], in which higher order examples are also presented. However, depending also on the frequency of the convexification, the computational load may be high for increasing values of n, as it is well known that for convex hull computations with $n > 3$ no $O(n \log n)$ algorithms are available.

4.3 Constrained stabilisation

Although this paper has been concentrated on the controllable/recoverable region estimation problem, the approach outlined can provide a number of different control laws. In addition to the largest scaling factors, the iterative procedure returns the control values (on the vertices of the resulting P.I. set) found by the linear programs (or specified by the ray-optimal control). It can be shown [8][Chapter 5] that these values can be used to generate a variety of different control laws with different performance characteristics. The ray-gridding approach can then be seen as a useful tool for achieving desirable trade-offs between robustness, performance and safety in the presence of uncertainties.

5 Conclusions

This paper has presented a new approach for calculating polytopic approximations of the controllable and recoverable regions of classes of nonlinear systems. The resulting polytopes have all their vertices on selected rays. By refining the ray partition better estimates can be progressively obtained. It has been shown by means of some examples that the computational complexity is kept low for planar systems and that the technique can deal with nonlinearities and uncertainties. For high order systems the computational load rises quickly, but it is believed that the technique is still less computationally expensive than gridding. Another good feature of the technique is that it can provide a variety of stabilising control laws that achieve certain trade-offs between robustness, performance and safety.

References

[1] Blanchini F., Carabelli S. Robust Stabilisation via computer-generated Lyapunov functions: an application to a magnetic levitation system, Proc. 33rd IEEE Conference on Decision and Control, 1994, pp. 1105-1106.

[2] Blanchini F., Miani S. Constrained stabilisation of continuous-time linear systems, Systems and Control Letters, Vol. 28, 1996, pp. 95-102.

[3] Blanchini F., Miani S., Constrained stabilisation via smooth Lyapunov functions, Systems and Control Letters, Vol. 35, 1998, pp. 155-163.

[4] Gilbert E.G., Tan K.T. Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets, IEEE TAC, Vol. 36, No. 9, 1991, pp. 1008-1020.

[5] Hu T., Qiu L., Lin Z. Stabilization of LTI systems with planar anti-stable dynamics using saturated linear feedback, Proc. 37th IEEE Conference on Decision and Control, 1998, pp. 389-394.

[6] Hu T., Qiu L. Controllable regions of linear systems with bounded inputs, Systems and Control Letters, Vol. 33, 1998, pp. 55-61.

[7] Stephan J., Bodson M., Lehoczky J. Calculation of recoverable sets for systems with input and state constraints, Optimal Control Applications and Methods, Vol. 19, 1998, pp.247-269.

[8] Yfoulis C.A. Stabilisation Studies for Piecewise Linear Control Systems, Ph.D. Thesis, Control Systems Centre, UMIST, Manchester, U.K., Jan 2000.