ТРУДЫ ИНСТИТУТА МАТЕМАТИКИ И МЕХАНИКИ УрО РАН

Том 21 № 2 2015

УДК 517.983.54

EXAMPLES OF COMPUTED VIABILITY KERNELS

Nikolai D. Botkin, Varvara L. Turova

Conflict control systems with state constraints are under consideration. A numerical method of constructing viability kernels, i.e the largest subsets of state constraints where the system trajectories can be confined, is applied to several nontrivial examples. The method can be interpreted as the approximate construction of the Hausdorff limit of sets generated by a backward procedure based on differential games theory essentially developed in works by N. N. Krasovskii and A. I. Subbotin. Numerically, the method is implemented in the framework of computing level sets of approximate solutions to an appropriate Hamilton–Jacobi equation. The main objective of the paper is the demonstration of nontrivial examples of computing viability kernels.

Keywords: differential game, state constraint, viability kernel, backward procedure, Hamilton–Jacobi equation, viscosity solution.

Н. Д. Боткин, В. Л. Турова. Примеры численного построения ядер выживаемости.

Рассматриваются конфликтно-управляемые системы с ограничениями на фазовые состояния. Численный метод построения ядер выживаемости (наибольших подмножеств фазовых ограничений, где система может удерживаться неограниченное время) применяется к некоторым нетривиальным задачам. Метод может интерпретироваться как построение предельных множеств некоторой попятной процедуры, базирующейся на на теории дифференциальных игр, существенно развитой в работах Н. Н. Красовского и А. И. Субботина. Численная реализация метода выполнена на языке множеств уровня вязких решений подходящего уравнения Гамильтона-Якоби. Основной целью статьи является демонстрация нетривиальных примеров построения ядер выживаемости.

Ключевые слова: дифференциальная игра, фазовое ограничение, ядро выживаемости, попятная процедура, уравнение Гамильтона — Якоби, вязкое решение.

Dedicated to the memory and the 70th anniversary of the Academician of RAS
Andrei I. Subbotin

1. Introduction

In many technical problems, it is required to keep a control system within a state constraint in the presence of unpredictable disturbances. Viability theory (see [1]) can be considered as an approach to the design of appropriate control laws. Notice that a viable subset of the state constraint determines a feedback strategy which can keep the state vector within this subset. Such a feedback strategy can be designed using the extremal aiming procedure proposed in [2]. A method for finding viability kernels (maximal viable subsets of state constraints) is proposed in [3]. This method is based on the necessary and sufficient viability conditions formulated in terms of contingent cones and vectogramms. The monograph [4] considers a wide range of problems related to analysis and construction of viable sets and viability kernels. Grid methods for constructing viability kernels have been studied in works by V. N. Ushakov and his pupils (see e.g. [5]).

In papers [6–8], a method for finding viability kernels is described and theoretically analyzed. The approach developed there is based on the theory of differential games (see [2]). The viability kernel is constructed as the Hausdorff limit of sets generated by a backward procedure that produces the time-sections of the so-called maximal stable bridges described in [2]. In [8], it was sketched how this algorithm can be numerically implemented in terms of level sets of limiting solutions of an appropriate Hamilton–Jacobi equation arising from conflict control problems with state constraints.

The present paper describes a slightly modified grid algorithm for computing viability kernels and shows its application to nontrivial examples of conflict control problems.

2. Differential game and viability kernels

In this section, some definitions related to viability theory are recalled.

Consider a conflict control system with the autonomous dynamics

$$\dot{x} = f(x, u, v), \quad x \in \mathbb{R}^n, \quad u \in P \subset \mathbb{R}^p, \quad v \in Q \subset \mathbb{R}^q. \tag{1}$$

Here, x is the state vector; u and v are control parameters of the first and second players, respectively; and P and Q are compacts of the corresponding dimensions. In the following, it is assumed that all functions of x are defined on the whole $\mathbb{R}^n$ and have global properties. Thus, the right-hand side f is supposed to be globally bounded, continuous in (x, u, v), and Lipschitzian in x.

Bearing in mind the conflict control system (1), consider for any $v \in Q$ the differential inclusion

$$\dot{x} \in F_v(x) = \overline{\text{co}}\{f(x, P, v)\}. \tag{2}$$

Definition 1 (viability property [1]). *A set $K \subset \mathbb{R}^n$ is called viable if for any $x_* \in K$ and any $v \in Q$ there exists a solution $x(\cdot)$ to the differential inclusion* (2) *with the initial state* $x(0) = x_*$ *such that* $x(t) \in K,\ t \geq 0$.

Definition 2 (viability kernel [1]). *For a given compact set $G \subset \mathbb{R}^n$, define $Viab(G)$ as the largest subset of G with the viability property. This subset is called the viability kernel of G.*

Proposition 1 (see e.g. [1]). *For any compact set $G \subset \mathbb{R}^n$ there exists a unique viability kernel $Viab(G) \subset G$ if there exists at last one viable subset of G.*

3. Computation of viability kernels

The algorithm proposed in [8] for finding viability kernels utilizes ideas of dynamic programming. Namely, a Pontryagin-like step-by-step backward procedure (see [9]) yielding a sequence of sets is used. The new feature consists in the reduction of the time step at each backward iteration in such a way that the time step tends to zero, but the sum of all time steps goes to infinity. The algorithm looks as follows:

$$\begin{aligned} K_0 &= G, \\ K_{i+1} &= \bigcap_{v \in Q} \Big[\bigcup_{x \in K_i} (x - \delta_{i+1} F_v(x)) + \delta_{i+1}^2 \beta S \Big] \bigcap G. \end{aligned} \tag{3}$$

Here S is the unit ball in $\mathbb{R}^n$, $\beta = LM$, L is the Lipschitz constant of F_v, and $M = \sup\{|f| \colon f \in F_v(x), x \in \mathbb{R}^n, v \in Q\}$.

Theorem 1 (see [8] for the proof). *Let the sequence $\{K_i\}$ be defined by (3), where the sequence $\{\delta_i\}$ is such that $\delta_i \to 0$ and $\sum_{i=1}^{\infty} \delta_i = \infty$. If $K_i \neq \varnothing$ for any $i > 0$, then there exists the Hausdorff limit, $\lim\limits_{i \to \infty} K_i$, and*

$$\lim_{i \to \infty} K_i = Viab(G) \neq \varnothing.$$

Otherwise, if $K_i = \varnothing$ for some $i > 0$, there are no viable subsets of G.

4. Numerical implementation

The numerical method utilizes the idea of representation of the viability kernel as a level set of an appropriate function. Suppose that the state constraint, G, is given by the relation

$$G = \{x \in \mathbb{R}^n, g(x) \leq 0\}, \tag{4}$$

where g is a continuous function. It is required to construct a function Vrepresenting the viability set as follows:

$$Viab(G) = \{x \in \mathbb{R}^n, V(x) \leq 0\}. \tag{5}$$

Define the Hamiltonian of the game (1) by the formula

$$H(x,p) = \max_{v \in Q} \min_{u \in P} p \cdot f(x,u,v)$$

and consider the following Hamilton–Jacobi equation

$$H(x, \nabla \mathcal{V}) = 0. \tag{6}$$

A continuous function $\mathcal{V}$ is a viscosity solution of (6) if the following conditions hold.

(i) $\mathcal{V}(x) \geq g(x)$ for all $x \in \mathbb{R}^n$.

(ii) For any point $y_0 \in \mathbb{R}^n$ and any function $\varphi \in C^1$ such that $\mathcal{V} - \varphi$ attains a local maximum at y_0, the inequality $H\left(y_0, \nabla\varphi(y_0)\right) \leq 0$ holds.

(iii) For any point $y_0 \in \mathbb{R}^n$ satisfying the inequality $\mathcal{V}(y_0) > g(y_0)$ and any function $\varphi \in C^1$ such that $\mathcal{V} - \varphi$ attains a local minimum at y_0, the inequality $H\left(y_0, \nabla\varphi(y_0)\right) \geq 0$ holds.

The existence of a unique viscosity solution follows from the results of [10] (cf. Theorem 2.1) and [11]. Moreover, the function $\mathcal{V}$ satisfying conditions (i)–(iii) has the property

$$Viab(\{x \in \mathbb{R}^n, g(x) \leq c\}) = \{x \in \mathbb{R}^n, \mathcal{V}(x) \leq c\}.$$

Indeed, the condition (i) provides the embedding of the level sets of $\mathcal{V}$ into the corresponding level sets of g. Condition (ii) provides the u-stability property of the function $\mathcal{V}$. Condition (iii) expresses the v-stability property at points y_0 where $\mathcal{V}(y_0) > g(y_0)$ (i.e. in the interior of the state constraint). Such a modification of the conventional definition of viscosity solutions was introduced in the work [11] to describe conflict control problems with state constraints. See also [12] for the relation between viability sets and viscosity solutions.

Our aim is to construct grid approximations of viscosity solutions of equation (6) and propose a control procedure that guaranties an approximate non-increase of this grid approximation along trajectories of the conflict control system (1).

Let $h := (h_1, \ldots, h_n)$ be spatial grid steps, and $|h| := (\sum_{i=1}^n h_i)^{1/2}$. Use the notation

$$\phi^h(x_{i_1}, \ldots, x_{i_n}) := \phi(i_1 h_1, \ldots, i_n h_n)$$

so that the upper index "h" denotes the restrictions of ϕ to the grid.

Introduce an interpolation operator $\mathcal{L}_h$ mapping grid functions to continuous ones and satisfying the estimate

$$\|\mathcal{L}_h[\phi^h] - \phi\| \leq C|h|^2 \, \|D^2\phi\| \tag{7}$$

for all smooth function ϕ. Here, ϕ^h is, as usually, the restriction of ϕ to the grid, $\|\cdot\|$ the pointwise maximum norm, $D^2\phi$ the Hessian matrix of ϕ, and C a constant independent on the grid and function ϕ.

It should be noticed that the estimate (7) is typical for interpolation operators (see e.g. [13]) because interpolation operators reconstruct not only the values but also the gradients of interpolated functions.

Consider, for example, a multilinear interpolation operator. Introduce the functions $\omega_0 = 1 - \xi$ and $\omega_1 = \xi$. Assume that a point $x = (x_1, x_2, \ldots, x_n)$ lies in a grid cell whose origin is given by the multiindex $(ip_1, ip_2, \ldots, ip_n)$. The multilinear interpolation of a grid function ϕ^h reads:

$$\mathcal{L}_h[\phi^h](x) = \sum_{i_1=ip_1}^{ip_1+1} \cdots \sum_{i_n=ip_n}^{ip_n+1} \phi^h(x_{i_1}, \ldots, x_{i_n})\, \omega_{i_1-ip_1}\Big(\frac{x_1 - x_{ip_1}}{h_1}\Big) \ldots\, \omega_{i_n-ip_n}\Big(\frac{x_n - x_{ip_n}}{h_n}\Big).$$

Use the following backward grid procedure applied to continuous functions defined on $\mathbb{R}^n$:

$$\mathcal{V}_{\ell+1} = \Pi_g[\delta, h; \mathcal{V}_\ell], \quad \mathcal{V}_0 = g, \quad \ell = 0, 1, \ldots, \tag{8}$$

where the operator Π_g is defined as follows:

$$\Pi_g[\delta, h; \phi](x) = \max\{\Pi[\delta, h; \phi](x); g(x)\}, \quad x \in \mathbb{R}^n \tag{9}$$

with

$$\Pi[\delta, h; \phi](x) = \max_{v\in Q} \min_{u\in P} \mathcal{L}_h[\phi]\big(x + \delta f(x, u, v)\big), \quad x \in \mathbb{R}^n. \tag{10}$$

The notation $\mathcal{L}_h[\phi]$ assumes that the interpolation operator is computed using the values of the function ϕ at grid nodes. In other words, $\mathcal{L}_h[\phi] \equiv \mathcal{L}_h[\phi^h]$, where the upper index "h" denotes the restriction of the function ϕ to the grid. Notice that the procedure (8)–(10) is closed with respect to grid functions, i.e. we can compute the function $\mathcal{V}_{\ell+1}$ in the grid nodes using only grid values of the functions $\mathcal{V}_\ell$ and g. Thus, this procedure is numerically feasible.

The procedure (8)–(10) is related to the algorithm (3) in the following way. The shift f of the argument of the function $\mathcal{L}_h[\mathcal{V}_\ell^h]$ in (10) results in the opposite shift of level sets of ϕ. The minimum over u results in the union of level sets of ϕ, and the maximum over v results in the intersection of the level sets. Moreover, the maximum in (9) results in the intersection of the level sets with the state constraint set G. Thus, the procedure (8)–(10) implements the algorithm (3) in terms of level sets. The difference consists in the usage of a constant value of the backward time step δ.

To be more precisely, consider properties of the operators Π and Π_g.

(j) It holds $\Pi[\zeta] = \zeta$ for a constant function ζ.

(jj) The operator Π is monotone, i.e. if $\phi_1 \le \phi_2$ pointwise, then $\Pi[\delta, h; \phi_1] \le \Pi[\delta, h; \phi_2]$ pointwise. Obviously, the operator Π_g is also monotone in this sense.

(jjj) The operator Π has the consistency property, i.e.

$$\left| \frac{\Pi(\delta, h; \phi)(x) - \phi(x)}{\delta} - H(x, \nabla\phi) \right| \le C\|D^2\phi\|\Big(\delta + \frac{h^2}{\delta}\Big) \tag{11}$$

for any twice differentiable function bounded with its second derivative. Here C is a constant independent on ϕ, δ, and h.

Property (j) follows from the definition of the operator Π and the property $\mathcal{L}_h[\zeta] = \zeta$. Property (jj) follows from the monotonicity of the operators $\mathcal{L}_h$ and "$\max_v \min_u$". Property (jjj) is being proven using the expansion of the function ϕ and the estimate (7).

Remark 1. *If the spatial steps are chosen to be proportional to a power of the time step,* $h = C_h\delta^\gamma$, $\gamma \ge 1$, *inequality* (11) *yields the conventional, one-parameter form, of consistency* (*see e.g.* [10] *and* [11]).

Using the monotonicity of the operator Π_g, it is easy to see that the sequence of functions $\mathcal{V}_\ell$ is monotonically growing and bounded from above. Indeed, applying the operator Π_g to the obvious inequality $g = \mathcal{V}_0 \le \mathcal{V}_1$ yields the relation $\mathcal{V}_1 \le \mathcal{V}_2$ and so on. Moreover, the inequality $g \le C$ (the function g is assumed to be bounded) implies the relation $\mathcal{V}_1 = \Pi_g[\delta, h; g] \le \max\{C, g\} \le C$. Repeating this implies the required result.

Therefore, the sequence $\mathcal{V}_\ell$ converges pointwise to a limiting, generally speaking, upper semicontinuous function $\mathcal{V}_\delta$ satisfying the relation

$$\mathcal{V}_\delta = \Pi_g[\delta, C_h\delta^\gamma; \mathcal{V}_\delta] = \max\left\{\Pi[\delta, C_h\delta^\gamma; \mathcal{V}_\delta], g\right\}, \tag{12}$$

assuming that the relation $h = C_h\delta^\gamma$, $\gamma \geq 1$ holds.

Remark 2. *If the viscosity solution, $\mathcal{V}$, of equation* (6) *is a Lipschitz continuous function, and all functions $\mathcal{V}_\delta$ are uniformly Lipschitzian with respect to δ, then some adaptation* (*simplification*) *of techniques of the work* [11] *provides the estimate*

$$\sup_{x\in\mathbb{R}^n} |\mathcal{V}_\delta - \mathcal{V}| \leq C_1\sqrt{\delta}.$$

In the general case, results of the work [10] *provide the local uniform convergence of functions V_δ satisfying the relation*

$$V_\delta = \Pi[\delta, C_h\delta; V_\delta] \tag{13}$$

to the viscosity solution of equation (6) *without the condition "$\mathcal{V}(y_0) > g(y_0)$" in* (iii). *A minimal adaptations of the proof of Theorem* 2.1 *of* [10] *to the case of the presence of the condition "$\mathcal{V}(y_0) > g(y_0)$" in* (iii) *yields the local uniform convergence of the functions $\mathcal{V}_\delta$ to $\mathcal{V}$ as $\delta \to 0$.*

Proposition 2. *The following discrete u-stability property of the function $\mathcal{V}_\delta$ is true. If $x_\star$ is a grid node such that $\mathcal{V}_\delta(x_\star) \geq g(x_\star)$, then for any $v \in Q$ there exists $u \in P$ such that $\mathcal{L}_h[\mathcal{V}_\delta](x_\star + \delta f(x_\star, u, v)) \leq \mathcal{L}_h[\mathcal{V}_\delta](x_\star) = \mathcal{V}_\delta(x_\star)$.*

The validity of this proposition follows from the definitions of the operators Π and Π_g and the relation (12). □

It is worth to mention another way of constructing an operator satisfying the properties (j)–(jjj). Define

$$\hat{\Pi}[\delta, h; \phi](x) = \phi(x) + \delta \max_{v\in Q}\min_{u\in P}\sum_{i=1}^{n}(p_i^{\text{right}} f_i^+ + p_i^{\text{left}} f_i^-), \tag{14}$$

with f_i being the components of $f(x, u, v)$, and

$$a^+ = \max\{a, 0\}, \quad a^- = \min\{a, 0\},$$
$$p_i^{\text{right}} = \frac{[\phi(x_1, \ldots, x_i + h_i, \ldots, x_n) - \phi(x_1, \ldots, x_i, \ldots, x_n)]}{h_i},$$
$$p_i^{\text{left}} = \frac{[\phi(x_1, \ldots, x_i, \ldots, x_n) - \phi(x_1, \ldots, x_i - h_i, \ldots, x_n)]}{h_i}.$$

The new operator $\hat{\Pi}$ does not use any interpolation and can immediately be applied to a grid function to return a grid function. The proof of the properties (j)–(jjj) can be found in [14].

The new numerical scheme formally assumes the same form as (8), i.e.

$$\hat{\mathcal{V}}_{\ell+1} = \hat{\Pi}_g\big[\delta, h; \hat{\mathcal{V}}_\ell\big] := \max\Big\{\hat{\Pi}\big[\delta, h; \hat{\mathcal{V}}_\ell\big], g\Big\}, \quad \hat{\mathcal{V}}_0^h = g, \quad \ell = 0, 1, \ldots. \tag{15}$$

The application of the method (15) requires holding the relation $\delta/h \leq 1/(M\sqrt{n})$ for all ℓ, where M is the bound of F_v, see [11] and [14]. Nevertheless, numerical experiments show a very nice feature of the method(15): in real computations involving bounded solution domains, the noise coming from the boundary of the grid area is absent, and therefore the grid area should not be too much larger than its part where the solution is searched. The procedure (8) does not possess such a property, which requires larger grid areas. On the other hand, larger time steps δ are allowed, which can compensate the use of larger grid regions.

Obviously, the functional sequence $\hat{\mathcal{V}}_\ell$ is non-decreasing and bounded from above, and therefore converges to a limiting function $\hat{\mathcal{V}}_\delta$. Moreover, Remark 2 holds true if $\mathcal{V}_\delta$ is replaced by $\hat{\mathcal{V}}_\delta$.

Unfortunately, an analog of Proposition 2 (approximate u-stability) cannot be easily proven.

5. Control procedure

In this section, the computation of optimal controls for system (1) using the procedure of extremal aiming (see [2] and [15]) is described. Suppose that the right-hand side of (1) satisfies the following saddle point condition:

$$\max_{u\in P}\min_{v\in Q}\ s\cdot f(x,u,v)=\min_{v\in Q}\max_{u\in P}\ s\cdot f(x,u,v) \tag{16}$$

for all x and s from $\mathbb{R}^n$.

Remember that the procedure (8) gives the values of the functions $\mathcal{V}_\ell$ in the grid nodes. These values converge monotonically and relatively quickly to the grid values of the limiting function $\mathcal{V}_\delta$. Thus, it is reasonable to assume that the values of $\mathcal{V}_\delta$ are known exactly in the grid nodes, and therefore the function $\mathcal{L}_h[\mathcal{V}_\delta]$ is known exactly.

Let $t_l=l\cdot\delta,\ l=0,1,\dots,$ be the time sampling of control process. Construct recursively a sequence, $\{\varepsilon_k\}$, as follows:

$$\varepsilon_0=|h|,\quad \bar{\varepsilon}^2_{k+1}=\varepsilon^2_k e^{L\delta}+R\delta^2+r(\delta),\quad \varepsilon_{k+1}=\bar{\varepsilon}_{k+1}+|h|,\quad k=0,1,\dots\ , \tag{17}$$

where the constant R and function $r(\delta)$ are defined below by formula (19).

Assume t_k and $x(t_k)$ being the current time instant and the state, respectively. Consider the neighborhood

$$\mathcal{U}_{\varepsilon_k}(x(t_k))=\{x\in\mathbb{R}^n:\ |x-x(t_k)|\le\varepsilon_k\} \tag{18}$$

of the current state. Search through all grid points $x^h\in\mathcal{U}_{\varepsilon_k}(x(t_k))$ and find a point x^h_k such that

$$\mathcal{V}_\delta(x^h_k)=\min_{x^h\in\mathcal{U}_{\varepsilon_k}(x(t_k))}\mathcal{V}_\delta(x^h).$$

The current control $u(t_k)$ of the first player applied on the time interval $[t_k,t_{k+1}]$ is computed from the condition of minimum projection of the system velocity $f\big(x(t_k),u,v\big)$ onto the direction of the vector $x(t_k)-x^h_k$, i.e.

$$u(t_k)=\arg\min_{u\in P}\max_{v\in Q}\big(x(t_k)-x^h_k\big)\cdot f\big(x(t_k),u,v\big).$$

Let $v(\xi)$ be an unknown control of the second player on the interval $[t_k,t_{k+1}]$. Introduce an auxiliary guide whose dynamics is described by the relation

$$y(t_{k+1})=y(t_k)+\delta f(y(t_k),u,v),$$

where the choice of controls u and v is at our disposal. Introduce an additional vector $v^0\in Q$ satisfying the relation

$$v^0=\arg\max_{v\in Q}\min_{u\in P}\big(x(t_k)-x^h_k\big)\cdot f\big(x(t_k),u,v\big).$$

Assume that the guide starts from the state $y(t_k)=x^h_k$. According to Proposition 2, for the control vector v^0 there exists a control vector $\bar{u}\in P$ such that

$$\mathcal{L}_h[\mathcal{V}_\delta](x^h_k+\delta f(x^h_k,\bar{u},v^0))\le\mathcal{L}_h[\mathcal{V}_\delta](x^h_k)=\mathcal{V}_\delta(x^h_k).$$

In other words,

$$\mathcal{L}_h[\mathcal{V}_\delta](y(t_{k+1})\le\mathcal{L}_h[\mathcal{V}_\delta](y(t_k)).$$

Estimate the distance $|x(t_{k+1})-y(t_{k+1})|$. It holds

$$x(t_{k+1})=x(t_k)+\int\limits_{t_k}^{t_k+\delta}f(x(\xi),u(t_k),v(\xi))d\xi,$$

$$y(t_{k+1})=y(t_k)+\int\limits_{t_k}^{t_k+\delta}f(y(t_k),\bar{u},v^0)d\xi.$$

Therefore,

$$\begin{aligned} x(t_{k+1}) - y(t_{k+1}) &= x(t_k) - y(t_k) \\ &+ \int\limits_{t_k}^{t_k+\delta} \big[f(x(t_k), u(t_k), v(\xi)) - f(x(t_k), \bar{u}, v^0)\big]\,d\xi \\ &+ \int\limits_{t_k}^{t_k+\delta} \big[f(x(\xi), u(t_k), v(\xi)) - f(x(t_k), u(t_k), v(\xi))\big]\,d\xi \\ &+ \int\limits_{t_k}^{t_k+\delta} \big[f(x(t_k), \bar{u}, v^0) - f(y(t_k), \bar{u}, v^0)\big]\,d\xi. \end{aligned}$$

Squaring the both sides of the above equation, accounting for the choice of $u(t_k)$ and v^0, and assuming that $x(t_k) - y(t_k)$ lies in a bounded domain for all k yield the estimate

$$|x(t_{k+1}) - y(t_{k+1})|^2 \le |x(t_k) - y(t_k)|^2 e^{L\delta} + R\delta^2 + r(\delta) \le \bar{\varepsilon}_{k+1}^2,$$

where

$$R = 12M^2 + MLD, \quad D = \sup_k |x(t_k) - y(t_k)|, \quad r(\delta) = 4LM^2\delta^3 + L^2M^2\delta^4. \tag{19}$$

Notice that, according to the relations (17), the grid cell containing $y(t_{k+1})$ entirely belongs to $U_{\varepsilon_{k+1}}(x(t_{k+1}))$, and therefore this grid cell has a node x_{k+1}^h such that

$$\mathcal{L}_h[\mathcal{V}_\delta](y(t_{k+1}) \le \mathcal{L}_h[\mathcal{V}_\delta](x_{k+1}^h) = \mathcal{V}_\delta(x_{k+1}^h).$$

Thus, the above described process can be repeated for the interval $[t_{k+1}, t_{k+2}]$. Finally, assuming that $|h| = C_h\delta\sqrt{\delta}$ and proving the estimate $\varepsilon_k^2 \le C\delta e^{Lt_k}$, where C is a constant depending on L, M, and D, yield the estimate $\mathcal{V}_\delta\big(x(t_k)\big) \le \mathcal{V}_\delta\big(x(t_0)\big) + G\sqrt{C}\delta^{1/2}e^{Lt_k/2}$ with G being the Lipschitz constant of the function $\mathcal{V}_\delta$.

Remark 3. *It should be noticed that the growth rate of the neighborhood U_{ε_k} with respect to k is large enough in theoretical considerations. In practical simulations, the value ε_k is defined as a minimum value such that*

$$\min_{x_h \in U_{\varepsilon_k}(x(t_k))} \mathcal{V}_\delta(x^h) \le \min_{x_h \in U_{\varepsilon_0}(x(t_0))} \mathcal{V}_\delta(x^h).$$

Numerical results show that ε_k defined according to this rule grows relatively slowly with k or remains constant.

Remark 4. *The control procedure of the second player is being designed in a similar way as for the first player. The difference consists in the use of the function $\bar{\mathcal{V}}_\delta$ satisfying the relation, cf.* (12),

$$\bar{\mathcal{V}}_\delta = \bar{\Pi}_g[\delta, C_h\delta^\gamma; \bar{\mathcal{V}}_\delta] := \max\big\{\bar{\Pi}[\delta, C_h\delta^\gamma; \bar{\mathcal{V}}_\delta], g\big\},$$

where the operator $\bar{\Pi}$ is defined as, cf. (10),

$$\bar{\Pi}[\delta, h; \phi](x) = \min_{u\in P}\max_{v\in Q} \mathcal{L}_h[\phi]\big(x + \delta f(x,u,v)\big), \quad x \in \mathbb{R}^n.$$

The last operator has the approximate v-stability property (*cf. Proposition* 2), *which allows us to repeat the above arguments and design a control rule of the second player guaranteeing the estimate*

$$\bar{\mathcal{V}}_\delta\big(x(t_k)\big) \ge \bar{\mathcal{V}}_\delta\big(x(t_0)\big) - \bar{G}\sqrt{C}\delta^{1/2}e^{Lt_k/2},$$

where $\bar{G}$ is the Lipschitz constant of the function $\bar{\mathcal{V}}_\delta$.

It should be noted that numerical experiments show that the difference $\max_{x^h}|\bar{\mathcal{V}}_\delta(x^h) - \mathcal{V}_\delta(x^h)|$ *is very small* (*less then* δ^2) *if the condition* (16) *holds, and therefore the function $\mathcal{V}_\delta$ can be used for the control design of the second player.*

Remark 5. *Remember that the function $\hat{\mathcal{V}}_\delta$ constructed with the difference operator* (14) *does not evidently possess the approximate u-stability property* (*cf. Proposition* 2). *Therefore, the above described procedure of control design for the first player is not applicable in this case. Nevertheless, a method described in* [16] *can apparently be adapted, which demands an additional investigation.*

6. Examples

6.1. A pendulum with a movable suspension point

The first example serves to the purpose of comparing results obtained by the application of the grid algorithm (15) with an exact analytical solution found in [17].

A pendulum with the movable suspension point is under consideration. It is assumed that this point can move with the velocity u. The performance is disturbed by a horizontal force v applied to the suspended load. This force is caused, e.g., by wind gusts. Denote by x_1 the angle deviation from the vertical axis, and let x_2 be the velocity of the suspended load in the ground reference system. These variables satisfy the equations

$$\left\{\begin{array}{lcl} \dot{x}_1 & = & \beta(x_2 - u), \\ \dot{x}_2 & = & -\alpha x_1 + v, \end{array}\right. \tag{20}$$

where α and β are positive constant coefficients depending on the suspended mass and pendulum's length. The controls u and v are bounded as follows:

$$|u| \le \mu, \quad |v| \le \nu, \tag{21}$$

and the following state constraints are imposed:

$$|x_1| \le a, \quad |x_2| \le b. \tag{22}$$

The first player controlling the parameter u strives to maintain the state constraints. The second player dealing with the control parameter v tends to violate the state constraints. Assume that $\beta = 1$, otherwise this can be achieved by proper time scaling.

This problem was analytically solved in [17] for all values of the parameters a, b, μ, ν, and α. The most interesting case corresponds to the following relations between the parameters:

$$b \ge \mu \quad \text{and} \quad |\sqrt{\alpha}a - b| \le \mu - \nu/\sqrt{\alpha}.$$

In this case, the boundary of the viability kernel includes parts of all edges of the state constraint (see Fig. 1 to the right).

The viability kernel was analytically constructed using four families of ellipses of the form

$$\frac{1}{\alpha}(\alpha x_1 - v)^2 + (x_2 - u)^2 = C. \tag{23}$$

The families, I–IV, correspond to the following pairs of the controls u and v (see Fig. 1 to the left):

$$\text{I:} \ \{\mu, \nu\}; \quad \text{II:} \ \{-\mu, \nu\}; \quad \text{III:} \ \{-\mu, -\nu\}; \quad \text{IV:} \ \{\mu, -\nu\}. \tag{24}$$

In the analytical construction, the ellipses from the families I, II, III, and IV passing through the points p_1, p_2, p_3, and p_4, respectively, were used. The points d_1, d_2, d_3, and d_4 were obtained as common points of the corresponding ellipses and the edges of the state constraint. Notice that the conjunction at the points p_1, p_2, p_3 and p_4 is smooth.

In the numerical construction based on the application of the grid algorithm (15), the following values of the parameters are chosen: $a = 6, b = 4, \mu = 1, \nu = 0.3$, and $\alpha = 0.5$. The function g

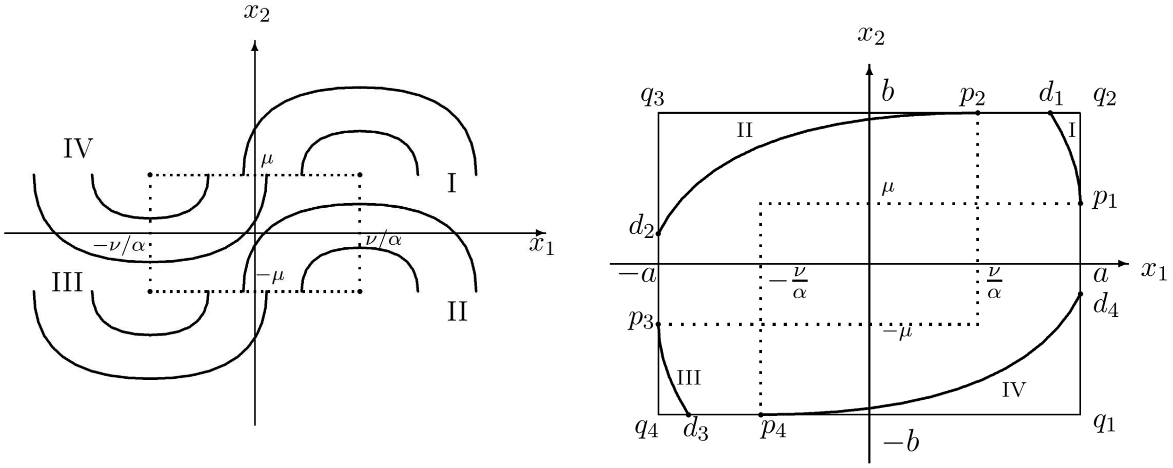

Fig. 1. Four families of ellipses (left) and the schematic structure of the viability kernel (right).

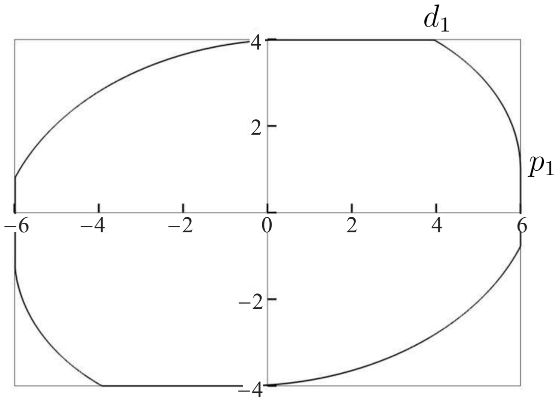

Fig. 2. The viability kernel computed numerically: $p_1 = (6, 1)$ and $d_1 = (3.95, 4)$. The exact values of p_1 and d_1 are $(6, 1)$ and $(3.94, 4)$, respectively.

defining the state constraint is of the form $g(x_1, x_2) = \max\{|x_1|/a, |x_2|/b\} - 1$. The time step δ is chosen as $\delta = 0.001$. The grid is obtained by dividing the region $[-7, 7]^2$ into 1000×1000 square cells. Computations using the algorithm (8) show that

$$\max_{\text{over grid}} |\mathcal{V}^h_{\ell+1} - \mathcal{V}^h_\ell| < 10^{-6} \text{ if } l \geq 8000,$$

which is the stop criterion. The runtime on a common laptop with a 4 core/8 thread processor is approximately 1 *min*. Figure 2 shows the viability kernel obtained as $Viab(G) = \{(x_1, x_2) : \mathcal{V}^h_\ell(x_1, x_2) \leq 0\}$. The computed coordinates of the points p_1 and d_1 are equal $(6, 1)$ and $(3.95, 4)$, respectively, whereas the analytically found values are $(6, 1)$ and $(3.94, 4)$, respectively. This proves the validity of the numerical algorithm (8). It should be noted that the algorithm (15) yields practically the same, visually indistinguishable, approximation of the viability kernel.

6.2. Acoustic version of the homicidal chauffeur game

The homicidal chauffeur game was firstly considered in [18]. In this game, the first player, a car whose minimum turn radius is bounded from below, strives to capture the second player, an inertia-less pedestrian, as soon as possible. The linear velocity of the car is assumed being constant, the magnitude of pedestrian's velocity is bounded by a given value. The car is controlled by changing the rate of turn of the linear velocity, whereas the pedestrian changes his velocity arbitrarily within

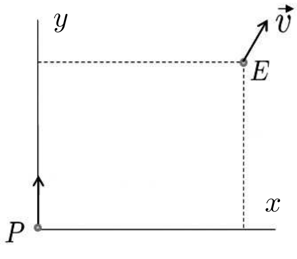

Fig. 3. Movable reference coordinate system in the homicidal chauffeur game.

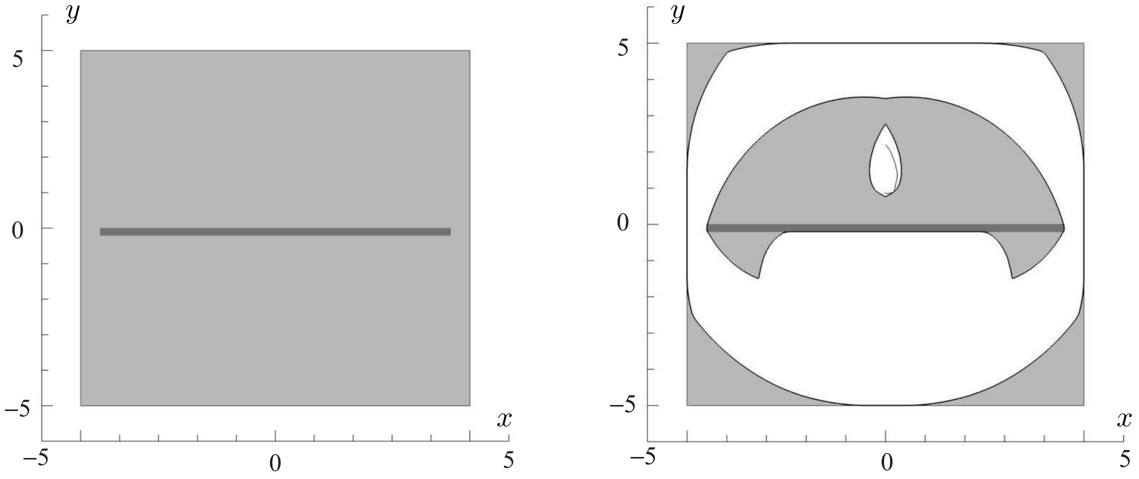

Fig. 4. To the left: the state constraint (light-grey). To the right: the viability kernel (white). The viability kernel consists of two disconnected parts. A trajectory corresponding to the optimal strategies of the players goes downwards and stops forever near the bottom of the small part.

the prescribed bounds. The original system consists of five differential equations, but it can be reduced (see [18]) to the following two-dimensional model:

$$\begin{aligned} &\dot{x} = -yu + v_1, \\ &\dot{y} = xu + v_2 - 1, \\ &|u| \le 1, \quad \sqrt{v_1^2 + v_2^2} \le \nu \end{aligned} \tag{25}$$

written in a movable reference frame with y-axis directed along the car velocity vector (see Fig. 3). The problem statement where the bound ν depends on the state (x, y) is known (due to Pierre Bernhard) as the acoustic version of the homicidal chauffeur game. The following case:

$$\nu(x, y) = \nu^* \min\left\{1, \sqrt{x^2 + y^2}/s\right\},$$

where ν^* and s are some positive constant parameters, was considered in [19] and [20].

Reformulate the problem as follows. Introduce the state constraint

$$G = [-c, c]^2 \setminus [-a, a] \times [-b, 0]$$

(see Fig. 4 to the left) given by the formula

$$g(x, y) := \max\left\{\max\{|x/c|, |y/c|\} - 1; \quad 1 - \max\{|x/a|, |2y/b + 1|\}\right\} \le 0.$$

The second player dealing with the control parameters v_1 and v_2 aims to maintain the state constraint, and the opposite player managing the control parameter u has the opposite objective.

Thus, one of the objectives of the first player is to bring the state vector to the set $\Gamma = [-a, a] \times [-b, 0]$, and the second player strives to avoid that. This establishes the relation with the original problem formulation used in [20] where Γ is considered as the target set of the first player.

The following values of the parameters are chosen in the numerical simulation based on the algorithm (8)–(10): $c = 5, a = 4.5, b = 0.2, \nu^* = 1.5,\ s = 0.9375$. The time step δ is chosen as $\delta = 0.001$. The grid is formed by dividing the region $[-7, 7]^2$ into 1000×1000 square cells, and 16000 iterations of the algorithm (8)–(10) are done.

Figure 4 shows the state constraint (to the left) and the computed viability kernel (to the right). The last consists of two disconnected parts. A trajectory started at a point of the small part and computed using optimal control strategies described in Section 5 goes downwards and remains forever near the bottom of the small part.

6.3. In-plane nonlinear double pendulum

This example demonstrates the treatment of a high-nonlinear system whose state vector lies in $\mathbb{R}^4$. In this case the algorithm (8)–(10) is more preferable because it allows us to choose larger time steps (see the remark at the end of Section 4). Figure 5 shows a sketch of the pendulum. The notations are self-explanatory. The control is the torque applied at the suspension point, and the disturbance is associated with wind. It is assumed that the force exerted by the wind on an linear element of the pendulum is computed as the product of the wind strength and the cosine of the angle between the element and wind directions.

The governing equations read

$$\dot{\theta}_1 = \frac{6}{ml^2}\frac{2\,p_1 - 3\cos(\theta_1 - \theta_2)\,p_2}{16 - 9\cos^2(\theta_1 - \theta_2)},$$

$$\dot{\theta}_2 = \frac{6}{ml^2}\frac{8\,p_2 - 3\cos(\theta_1 - \theta_2)\,p_1}{16 - 9\cos^2(\theta_1 - \theta_2)},$$

$$\dot{p}_1 = -\frac{1}{2}ml^2\left[\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + 3\frac{g}{l}\sin\theta_1\right] + u + vl^2\left[\frac{1}{2}\cos^2\theta_1 + \cos\theta_1\cos\theta_2\right],$$

$$\dot{p}_2 = -\frac{1}{2}ml^2\left[-\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + \frac{g}{l}\sin\theta_2\right] + vl^2\frac{1}{2}\cos^2\theta_2.$$

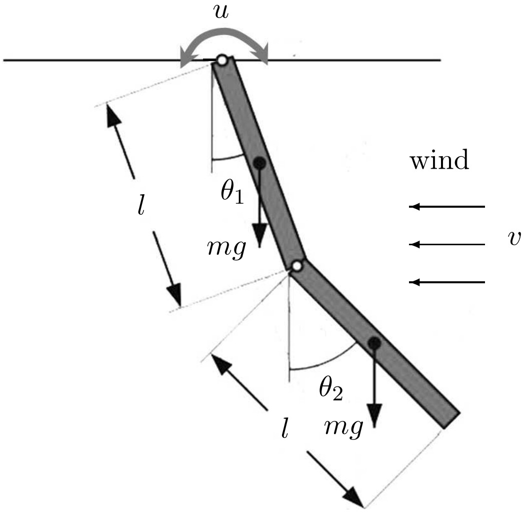

Fig. 5. Sketch of an in-plane nonlinear double pendulum. The control parameter is the torque applied at the suspension point, and the disturbance is associated with wind.

Here, p_1 and p_2 are the so-called generalized momenta defined by the relation

$$p_i = \frac{\partial L}{\partial \dot{\theta}_i}, \quad i = 1, 2,$$

where

$$L = \frac{1}{6} m l^2 \left[4\dot{\theta}_1^2 + \dot{\theta}_2^2 + 3\dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) \right] + \frac{1}{2} mgl (3\cos\theta_1 + \cos\theta_2)$$

is the Lagrangian of the system (kinetic energy minus potential energy).

Assume that the control and the disturbance are bounded as follows:

$$|u| \leq 1, \quad |v| \leq 0.1,$$

and the state constraint $G \in \mathbb{R}^4$ is defined by the inequality $g(p_1, p_2, \theta_1, \theta_2) \leq 0$, where

$$g(p_1, p_2, \theta_1, \theta_2) = \max\left\{ \sqrt{p_1^2 + p_2^2}, |\theta_1|/2.6, |\theta_2|/2.6 \right\} - 1.$$

Therefore, the central p_1, p_2 cross-section of the state constraint is the unique circle (see Fig. 6, to the left), and the central θ_1, θ_2 cross-section is 2.6 × the unit square (see Fig. 6, to the right).

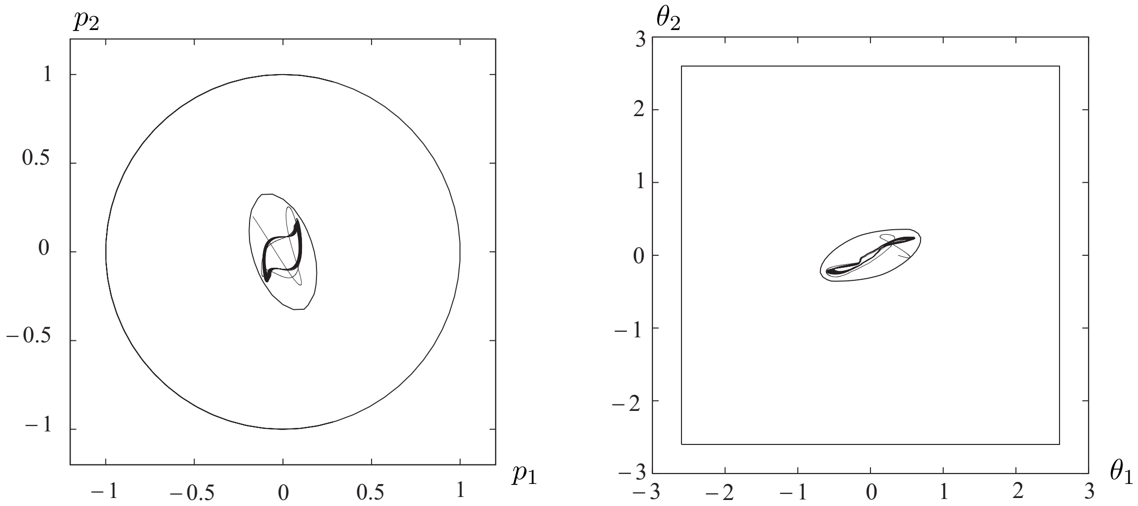

Fig. 6. To the left: the central p_1, p_2 cross-section of the state constraint and the viability kernel; the p_1, p_2 projection of the trajectory computed using optimal strategies of the players. To the right: the same but for the central θ_1, θ_2 cross-section and θ_1, θ_2 projection.

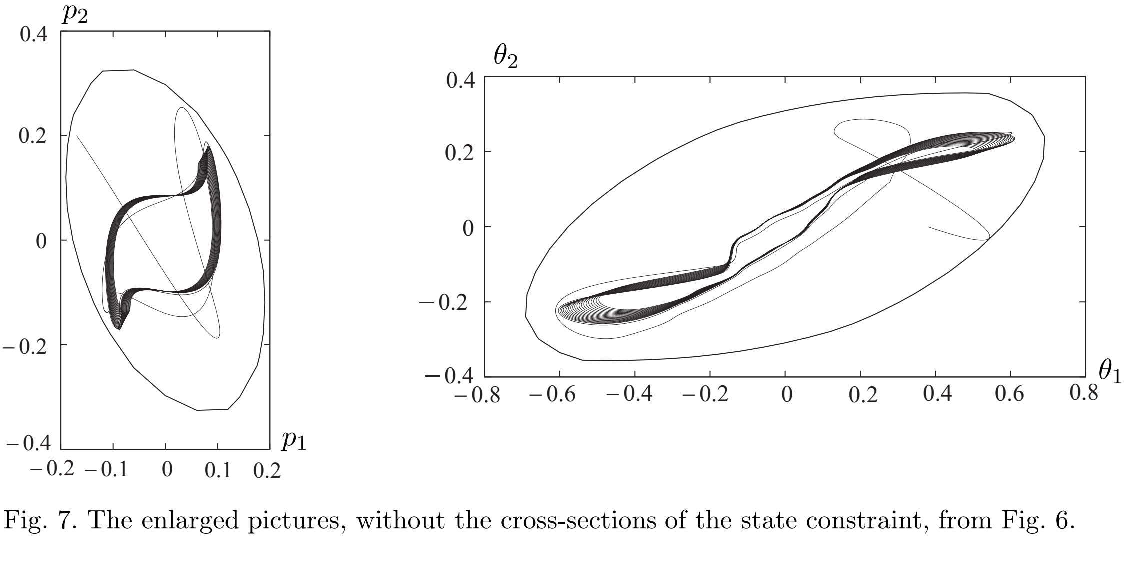

Fig. 7. The enlarged pictures, without the cross-sections of the state constraint, from Fig. 6.

In the numerical computations, the time step δ is chosen as $\delta = 0.01$. The grid is formed by dividing the region $[-1.2, 1.2]^2 \times [-3, 3]^2$ into $40^2 \times 100^2$ cells, and 8000 iterations of the algorithm (8)–(10) are done.

Fig. 6 (to the left) shows the central p_1, p_2 cross-section of the viability kernel and the p_1, p_2 projection of a trajectory corresponding to the optimal control laws described in Section 5. The trajectory remains in the viability kernel arbitrary long.

Fig. 6 (to the right) shows the central θ_1, θ_2 cross-section of the viability kernel and the θ_1, θ_2 projection of the trajectory.

Fig. 7 shows the enlarged pictures, without the cross-sections of the state constraint, from Fig. 6.

7. Conclusions

In this work, viability kernels for some nontrivial problems are computed using numerical procedures developed by the authors in the framework of grid methods for solving differential games. Theoretically, these numerical procedures are suitable for arbitrary dimensions of the state vector. Currently, the maximal number of processors used in the computations is up to 32, which allows us to handle nonlinear problems of the dimension less or equal to four. The usage of supercomputing systems with large number of processors and the utilization of sparse representations of grid functions will enable us to advance our capabilities towards higher dimensionality. This should support the treatment of e.g. aircraft applications related to essentially nonlinear takeoff and landing problems with complex state constraints.

Acknowledgements

This work was supported in part by the DFG grant TU427/2-1.

REFERENCES

1. **Aubin J.P.** A survey of viability theory // SIAM J. Control Optim. 1990. Vol. 28, no. 4. P. 749–788.
2. **Krasovskii N.N. and Subbotin A.I.** Game-theoretical control problems. New York: Springer-Verlag, 1988. 517 p.
3. **Quincampoix M.** Frontieres de domaines d'invariance et de viabilite pour des inclusions differentielles avec contraintes // Comptes Rendus de l'Académie des Sciences - Series I - Mathematics. 1990. Vol. 311. P. 411–416.
4. **Aubin J.-P., Bayen A.M. and Saint-Pierre P.** Viability theory: New directions. Berlin; Heidelberg: Springer-Verlag, 2011. 830 p.
5. **Neznakhin A. A. and Ushakov V. N.** A discrete method for constructing an approximate viability kernel of a differential inclusion // Comput. Math. Math. Phys. 2001. Vol. 41, no. 6. P. 846–859.
6. **Botkin N.D.** Asymptotic behavior of solution in differential games. Viability domains of differential inclusions // Dokl. Akad. Nauk. 1992. Vol. 325, no. 1. P. 16–19.
7. **Botkin N.D.** Construction of viable sets for autonomous controlled systems. Preprint No. 430. Würzburg: University of Würzburg, 1993. 14 p. (Preprint Ser. SPP Anwendungsbezogene Optimierung und Steuerung (Applied optimization and control)).
8. **Botkin N.D. and Turova V.L.** Numerical construction of viable sets for autonomous conflict control systems // Mathematics. 2014. Vol. 2. P. 68–82.
9. **Pontryagin L.S.** Linear differential games. 2 // Soviet Math. Dokl. 1967. Vol. 8. P. 910–912.
10. **Barles G. and Souganidis P.E.** Convergence of approximation schemes for fully nonlinear second order equations // Asymptot. Anal. 1991. Vol. 4, no. 3. P. 271–283.
11. **Botkin N. D., Hoffmann K.-H., Mayer N. and Turova V. L.** Approximation schemes for solving disturbed control problems with non-terminal time and state constraints // Analysis. 2011. Vol. 31, no. 4. P. 355–379.
12. **Bardi M. and Goatin P.** Invariant sets for controlled degenerate diffusions: a viscosity solutions approach // Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming / eds. W.H. Fleming, W.M. McEneaney, G.G. Yin, and Q. Zhang. Boston: Birkhäuser, 1999. P. 191–208. (Systems Control Found. Appl.)

13. **Mößner B. and Reif U.** Error bounds for polynomial tensor product interpolation // Computing. 2009. Vol. 86, no. 2-3. P. 185–197.
14. **Botkin N. D., Hoffmann K.-H. and Turova V. L.** Stable numerical schemes for solving Hamilton–Jacobi–Bellman–Isaacs equations // SIAM J. Sci. Comp. 2011. Vol. 33, no. 2. P. 992–1007.
15. **Subbotin A.I. and Chentsov A.G.** Optimization of guaranteed result in control problems.Moscow: Nauka, 1981 (in Russian). 288 p.
16. **Subbotina N.N. and Tokmantsev T.B.** On the efficiency of optimal grid synthesis in optimal control problems with fixed terminal time // Differ. Equ. 2009. Vol. 45, no. 11. P. 1686–1697.
17. **Botkin N.D. and Ryazantseva E.A.** Structure of viability kernels for some linear differential games // J. Optim. Theory Appl. 2011. Vol. 147, no. 1. P. 42–57.
18. **Isaacs R.** Differential games. New York: John Wiley, 1965. 408 p.
19. **Cardaliaguet P., Quincampoix M. and Saint-Pierre P.** Set valued numerical analysis for optimal control and differential games // Stochastic and Differential Games: Theory and Numerical Methods / eds. M. Bardi, T. Parthasarathy, and T. E. S. Raghavan. Boston: Birkhäuser, 1999. P. 177–247. (Ann. Internat. Soc. Dynam. Games, 4).
20. **Patsko V. S. and Turova V. L.** Level sets of the value function in differential games with the homicidal chauffeur dynamics // Game Theory Review. 2001. Vol. 3, no. 1. P. 67–112.

Received February 14, 2015

Nikolai D. Botkin
Technische Universität München, Fakultät für Mathematik
Boltzmannstr. 3, 85747 Garching, Germany
e-mail: botkin@ma.tum.de

Varvara L. Turova
Technische Universität München, Fakultät für Mathematik
Boltzmannstr. 3, 85747 Garching, Germany
e-mail: turova@ma.tum.de