SET-VALUED EULER'S METHOD WITH INTERPOLATED DISTANCE FUNCTIONS AND OPTIMAL CONTROL SOLVERS*

ROBERT BAIER†

Key words. set-valued Euler's method, reachable sets, distance functions, differential inclusions, optimal control

AMS subject classifications. 34A60, 93B03, 49M25, 65D18, 65L06, 49J15

EXTENDED ABSTRACT. Consider the nonlinear control system

$$x'(t) = f(t, x(t), u(t)) \quad (\text{a.e. } t \in [t_0, T]), \tag{1}$$
$$u(t) \in \mathrm{U} \quad (\text{a.e. } t \in [t_0, T]), \tag{2}$$
$$x(t_0) \in X_0, \tag{3}$$

where $U \subset \mathbf{R}^m$ denotes the control region and $X_0 \subset \mathbf{R}^n$ the starting set. If not only *one optimal* trajectory and corresponding optimal control have to be determined, but the end points of *all* admissible solutions of the control problem, the so-called *reachable set* $\mathcal{R}(T, t_0, X_0)$ at end time T has to be computed. Defining a set-valued map $F(t,x) = \bigcup_{u \in U} \{f(t,x,u)\}$, we can consider the equivalent differential inclusion

$$x'(t) \in \mathrm{F(t,x(t))} \quad (\text{a.e. } t \in [t_0, T]), \tag{4}$$
$$x(t_0) \in X_0. \tag{5}$$

Applications range from collision avoidance in safety systems for cars [6], short-term studies of weather forecasts or long-run studies of climate research models [5], motion planning [4] to population models as in fish harvesting.

While for linear control problems, the reachable set at a given end time T is always convex and efficient set-valued quadrature methods using support functions and supporting points can be applied (see e.g. [3]), reachable sets for nonlinear control problems are usually nonconvex and thus challenging to compute.

Besides the popular use of level-set methods in [9] and the solution of the Hamilton-Jacobi-Bellman PDE (see e.g. [4]), set-valued Runge-Kutta methods such as Euler's method for (4)–(5) are studied for approximating reachable sets. The Euler set iterates for a given step-size $h = (T - t_0)/N$ are

$$X_{j+1}^h = \bigcup_{\eta_j \in X_j^h} \Big(\eta_j + hF(t_j, \eta_j) \Big) \quad (j = 0, \ldots, N-1), \quad X_0^h = X_0. \tag{6}$$

The first implementation is due to [5] in which for each iterative step, the Euler iterates are discretized by grid points in state-space and gathered in parallel line segments.

To be able to guarantee the overall order of convergence 1 for Euler's method, each set X_j^h must be discretized up to an error $O(h^2)$, see [5, Section 4.3]. This leads

*This work has been supported by the European Union Seventh Framework Programme [FP7-PEOPLE-2010-ITN] under grant agreement 264735-SADCO.

†Chair of Applied Mathematics, Department of Mathematics, University of Bayreuth, 95440 Bayreuth, Germany (robert.baier@uni-bayreuth.de).

to computational complexity and high memory consumption for this implementation. One tool for reducing the computational load is the use of distance functions.

Distance functions appear widely, not only in proofs for numerical methods, but also in numerical algorithms in computational geometry (morphing, set operations, set reconstruction from cross-sections, ...) as well as in the popular level-set methods.

The set operations used in (6) can be easily expressed via distance functions:

$$\operatorname{dist}(g, X_{j+1}^h) = \inf_{\eta_j \in X_j^h} \operatorname{dist}\left(\frac{g-\eta_j}{h}, F(t_j, \eta_j)\right) \quad (g \in \mathbf{R}^n) \tag{7}$$

Using the signed distance function $b(g, X) = \operatorname{dist}(g, X) - \operatorname{dist}(g, X^c)$ which is more often regular than the Lipschitz-continuous distance function, only the inequality

$$b(g, X_{j+1}^h) \le \widetilde{b}(g, X_{j+1}^h) = \inf_{\eta_j \in X_j^h} b\left(\frac{g-\eta_j}{h}, F(t_j, \eta_j)\right) \tag{8}$$

holds. Nevertheless, the right-hand side of (8) can serve as a representing function, i.e. describes the union in (6) as a level set:

$$X_{j+1}^h = \{g \in \mathbf{R}^n \,:\, \widetilde{b}(g, X_{j+1}^h) \le 0\}$$

To avoid the state-space discretization of order $O(h^2)$ for the vectors g in a bounding box and for the sets X_j^h, the values of the (signed) distance function in (7) are calculated on a coarse grid with distance $O(h)$. These values serve as a basis for the spatial piecewise linear interpolation of the distance function on a fine grid with distance $O(h^2)$. The linear interpolation for the distance function is much quicker to evaluate than the exact formula (7) on a fine grid with order $O(h^2)$. First numerical tests still indicate order of convergence 1 and a considerable speedup in calculation.

EXAMPLE 1 ([5, Example 5.2.5]). *Let $U = [-1, 1]$ be the control set and $x_0 = (-5, -5)$ the starting point. We calculate the reachable set at time $T = 2.5$ of*

$$x'(t) \in \begin{pmatrix} x_2 \\ -x_1 + x_2(1.4 - 0.14x_2^2) \end{pmatrix} + \begin{pmatrix} 0 \\ 4 \end{pmatrix} U \quad (t \in [0, 2.5])$$

Table 1 shows that there is an enormous improvement of the CPU times for the parallized method (2 × Intel Xeon E5640, 2.66 GHz, 24 GB RAM) without loosing accuracy (see [8, Section 5.2]). In Fig. 1 the convergence of the black sets X_N^h to the gray reference solution from [7] with $N = 80$ show graphical order of convergence 1.

N	$\mathcal{O}(h^2)$-grid	$\mathcal{O}(h)$-grid with lin. interpolation (LIP)	N	$\mathcal{O}(h^2)$-grid	$\mathcal{O}(h)$-grid with LIP
5	$\le$ 1s	$\le$ 1s	35	16h 8min 7s	8min 8s
10	$\le$ 1s	$\le$ 1s	40	40h 50min 29s	12min 7s
15	39s	4s	50	–	3h 4min 33s
20	5min 46s	10s	60	–	7h 11min 7s
25	50min 0s	1min 50s	70	–	14h 53min 28s
30	4h 3min 25s	4min 3s	80	–	22h 28min 36s

TABLE 1
Comparison of CPU times for Example 1 and the two approaches

In [1, 2], an approach based on distance functions calculated via optimal control solvers was studied. By formulating an appropriate optimal control problem (OCP)

$$\min_{\text{s.t. (1)–(3)}} \frac{1}{2}\|g - x(T)\|_2^2, \tag{9}$$

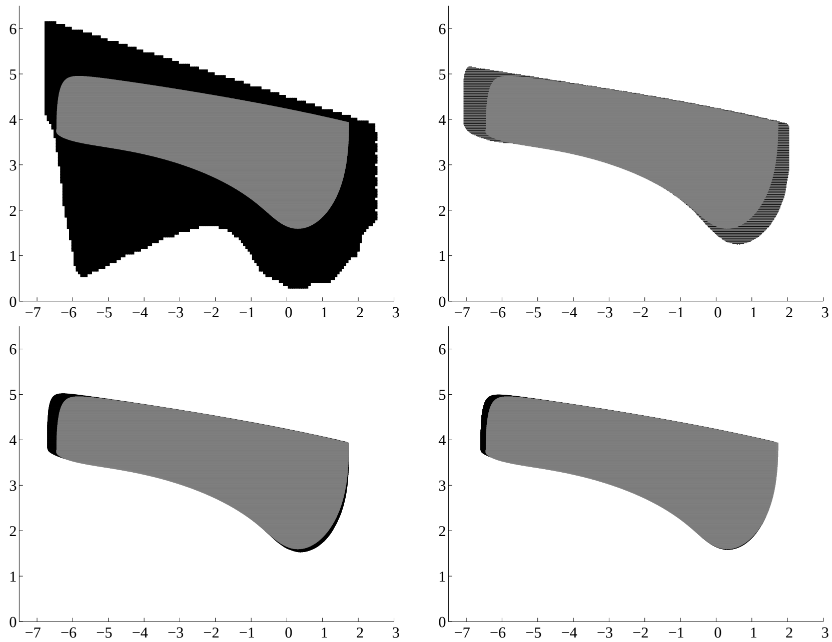

FIG. 1. *reachable sets for Example 1 with* $N = 10, 20$ *(first row) and* $N = 40, 60$ *(second row)*

the distance function $\operatorname{dist}(g, \mathcal{R}(T, t_0, X_0))$ from an arbitrary grid point $g \in \mathbf{R}^n$ to the reachable set appears in the optimal value. By varying the grid point within a bounding box G, we can approximate the reachable set via the set representation

$$\mathcal{R}(T, t_0, X_0) = \mathbf{R}^n \setminus \left\{ \bigcup_{g \in G} \operatorname{int} B_{r(g)}(g) \right\} \quad \text{with} \quad r(g) = \operatorname{dist}(g, \mathcal{R}(T, t_0, X_0)) \tag{10}$$

with the complement of open balls. The solution of Euler's discretization for the OCP (9) in [1, 2] enables the calculation of the distance function to the discrete reachable set. Hence, OCP solvers based on nonlinear programming yield the values of $\operatorname{dist}(g, X_N^h)$ for grid points $g \in G$ and approximate the Euler's iterate X_N^h via (10).

EXAMPLE 2 ([1, Example 3]). *Let* $U = [-1, 1]$ *be the control set and* $x_0 = (2, 2)$ *the starting point. We calculate the reachable set at time* $T = 1$ *of*

$$x'(t) \in 8Ax(t) - 16a_{12} \begin{pmatrix} x_2 \\ -x_1 \end{pmatrix} U \quad (t \in [0, 1])$$

with $A = \begin{pmatrix} \sigma^2 - 1 & \sigma\sqrt{1-\sigma^2} \\ -\sigma\sqrt{1-\sigma^2} & \sigma^2 - 1 \end{pmatrix}$, $a_{12} = \sigma\sqrt{1-\sigma^2}$ *and* $\sigma = 0.9$.

It was first noticed in [7] that the calculated reachable sets in [1] based on results by a local optimizer only represent subsets of the discrete reachable sets. Since the approach introduced here is realizing the full Euler steps, the results in Fig. 2 are correct, but calculated considerably slower as the non-parallelized algorithm in [1]. The difference for $N = 15, 20$ *between the realization on a full* $\mathcal{O}(h^2)$*-grid (in gray) and the linear interpolation on a* $\mathcal{O}(h)$*-grid (in black) is hardly visible in Fig. 2.*

The numerical test examples in [6] indicate the superior performance, even in the case in which more than four state variables appear in the control system and only a lower-dimensional projection of the reachable set has to be computed.

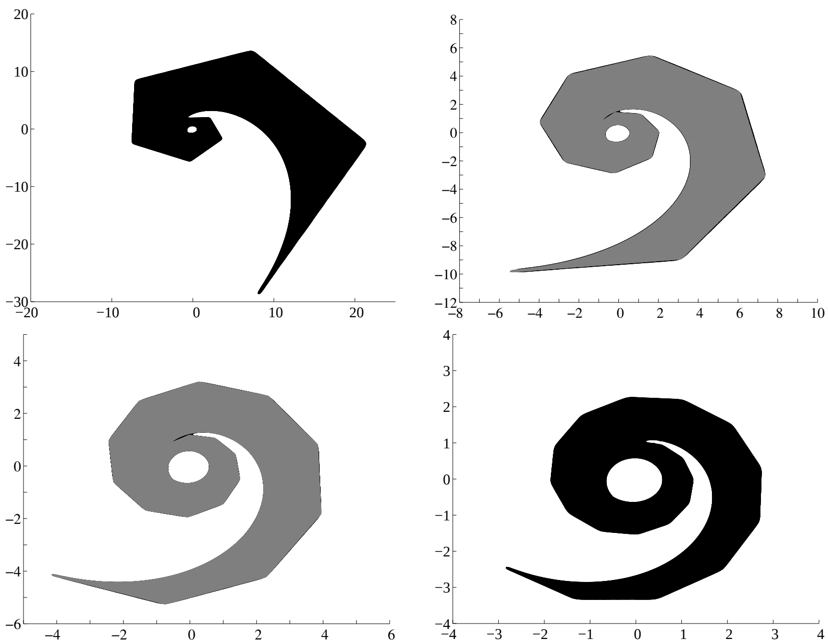

FIG. 2. *reachable sets for Example 2 with $N = 10, 15$ (first row) and $N = 20, 25$ (second row)*

The author would like to acknowledge the help by Eduard Leipi who realized the approach with the linear interpolation of distance functions in [8].

REFERENCES

[1] R. BAIER, M. GERDTS, *A computational method for non-convex reachable sets using optimal control*, in Proceedings of the European Control Conference (ECC) 2009, Budapest (Hungary), August 23–26, EUCA, 2009, pp. 97–102.

[2] R. BAIER, M. GERDTS, I. XAUSA, *Approximation of reachable sets using optimal control algorithms*, Technical report, University of Bayreuth, 36 pages (2011), submitted. `http://num.math.uni-bayreuth.de/de/publications/2011/baier_gerdts_xausa_approx_reach_sets_2011/`

[3] R. BAIER, F. LEMPIO, *Computing Aumann's integral*, in Modeling Techniques for Uncertain Systems, Proceedings of a Conference held in Sopron, Hungary, July 6–10, 1992, A. B. Kurzhanski and V. M. Veliov, eds., Progress in Systems and Control Theory, Vol. 18, Birkhäuser, Basel, 1994, pp. 71–92.

[4] O. BOKANOWSKI, N. FORCADEL, H. ZIDANI, *Reachability and minimal times for state constrained nonlinear problems without any controllability assumption*, SIAM J. Control Optim., 48, no. 7 (2010), pp. 4292–4316.

[5] I. A. CHAHMA, *Set-valued discrete approximation of state-constrained differential inclusions*, Bayreuth. Math. Schr. 67 (2003), pp. 3–162.

[6] M. GERDTS, I. XAUSA, *Collision avoidance using reachable sets and parametric sensitivity analysis*, Technical report, Bundeswehruniversität München, 11 pages (2011), submitted.

[7] L. GRÜNE, T. JAHN, *Computing reachable sets via barrier methods on SIMD architectures*, to appear in Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), September 10–14, 2012, Vienna, Austria.

[8] E. LEIPI, *Lineare Interpolation von Distanzfunktionen zur effizienteren Berechnung erreichbarer Mengen [Linear Interpolation of Distance Function to Compute Reachable Sets More Efficiently]*, Diploma Thesis, Department of Mathematics, University of Bayreuth, February 2012, 105 pages.

[9] J. A. SETHIAN, *Level set methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 1996.