Algorithm for computing basins of attraction in Filippov systems

March 10, 2017

1 How to use the code

At the moment, the algorithm works for planar Filippov systems with a single discontinuity surface and it has been implemented in MATLAB environment. Here we will give some details on how to use the program for computing the basins of attraction in Filippov systems, as a particular example, we consider a discontinuous control problem.
The program consist of four files,

`fillipov.m`
`jacobians.m`
`vectorfields.m`
`Run_BAII.m`

and a folder LIB_RUN which contains a set of functions used in the main files mentioned before.

The main file that the user needs to adjust for computing the BA is `Run_BAII.m` which contains embedded functions for allocating the grid dynamically, and also for redefining the basin boundary. In the `Run_BAII.m` file the user define: a point with coordinates (x_1,x_2) in the state space where the program starts the computation of BA, a parameter related with the investigated area and a refinement parameter expressed as a vector in coordinates (x_1,x_2), setting the subdivision of the boundary cells in both coordinates. Extra parameters such as the minimum cell-width of the cells added at each iteration and the initial set of cells can also be set by the user. In the file `IMCELL.m` contained in the folder LIB_RUN, the user introduces the ODE solver and its properties (e.g., error tolerances) and the initial integration time. The integration time can be chosen representative large, since the algorithm is constantly updating this parameter according to the system dynamics.

In the vector field file `vectorfield.m` which is used by `filippovN.m`, the user adds the different vector fields, the function defining the discontinuity surface

and the normal to the discontinuity surface.
Let us show explicitly, how can be used the program. We consider the system

$$\dot{x} = \begin{pmatrix} -1 & 1 \\ 0 & 3 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \tag{1}$$

where the discontinuous control is given by $u = -10\,\mathrm{sgn}(x_1 + x_2)$. The control strategy is designed to steer the system dynamics to the zero equilibrium point, which lies on the discontinuity manifold. To use the program, let us rewrite the system within the Filippov formalism i.e.,

$$\dot{x} = \begin{cases} F_1(x), \ x \in R_1 \\ F_2(x), \ x \in R_2 \end{cases} \tag{2}$$

where the two vector fields are:

$$F_1(x) = \begin{pmatrix} -x_1 + x_2 \\ 3x_2 - 10 \end{pmatrix}, \quad F_2(x) = \begin{pmatrix} -x_1 + x_2 \\ 3x_2 + 10 \end{pmatrix}. \tag{3}$$

with the zero-level set, $\Sigma = \left\{ x \in \mathbb{R}^2 : x_1 + x_2 = 0 \right\}$. The user needs to be adjust the files as follow.

Run_BAII.m
In this file, the user has to specify the basin of attraction parameters

```
%parameters
Eq = [0 0]; %     %Initial point in desired region
num_iter_user = 8;              %Number of iterations defined by user
NC_B = [2 2];                   %Refinement
cell_dim = 3;                   %Cell-width dimension
cell_it = [3 3];                %Initial cell number
```

The algorithm displays the mapping sequence used to construct the BA of interest, as shown in Fig. 1, and the final result as depicted in Fig. 2.
The system dynamics information and the solver parameters have to be specified as follow.

IMCELL.m
In this file, the user has to specify the constant matrices and simulation time

```
% ODE solver
solver = 'ode45';
opts = odeset('InitialStep',1e-7,'RelTol',1e-5,'AbsTol',[1e-5 1e-5],'MaxStep',0.1);

% Name of the file with the two vectorfields
vfields = 'vectorfields';
```

```
% Name of the file with the two Jacobianss
jacobians = 'jacobians';

% Integration time
T = 2;
tspan = [0 T];

% Filippov parameters
Cp = 1;
params = [];
```

vectorfields.m
In this file the user has to specify the following

```
%Vector field in region 1 - H(x) > 0

F1 = [-y(1)+y(2);
      3*y(2)-10];

%Vector field in region 2 - H(x) < 0

F2 = [-y(1)+y(2);
      3*y(2)+10];

%Switching Manifold

H = y(1)+y(2);

%A vector normal to the switching manifold
dH = [1,1];
```

jacobians.m
In this file the user has to specify the following

```
%Jacobian in region R1 (H(x) > 0)
J1 = [-1 1;0 3];

%Jacobian in region R2 (H(x) < 0)
J2 = J1;

% grad(grad(H)) A vector normal to the discontinuity surface
d2H = zeros(size(J1));
```

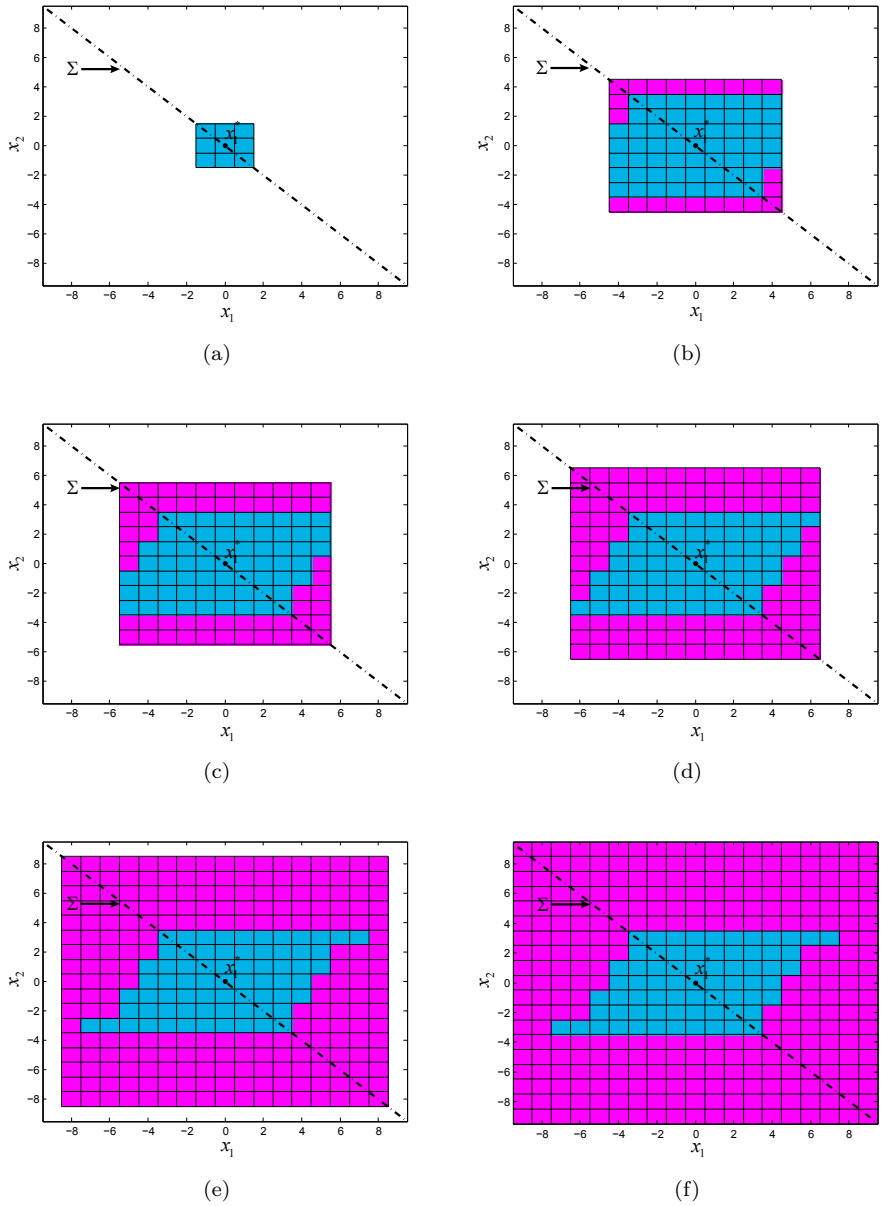

Figure 1: Sequence of the dynamic construction of the cell state space. The black lines are used to represent the cell state space, so that each square represents a cell. The black dash-line corresponds to Σ. The initial region (blue cells) used in the first iteration is shown in (a). The 3^{nd}, 4^{th}, 5^{th} and 6^{th} iterations of the algorithm are shown in (b)-(e), respectively. A first (coarse) approximation of the basins in the 7^{th} iteration is shown in (f).

Cells colored in blue belong to the basin of attraction of the equilibrium $x_1^* = (0, 0)$, while initial conditions in the magenta cells are mapped outside the basin of attraction. The total number of cells investigated by the ESCM at the end of the first stage is 361, as shown in Fig. 1 (f). Nevertheless, it is worth noting that, at each step of the algorithm just a small group of regular cells is investigated, since starting from a small set of cells around the origin, extra layers of cells are only added and examined iteratively. Once the entire basin has been mapped a refinement stage is performed in order to get more accurate results, in which the boundary the boundary cells are split, and the ESCM is applied again as described above.

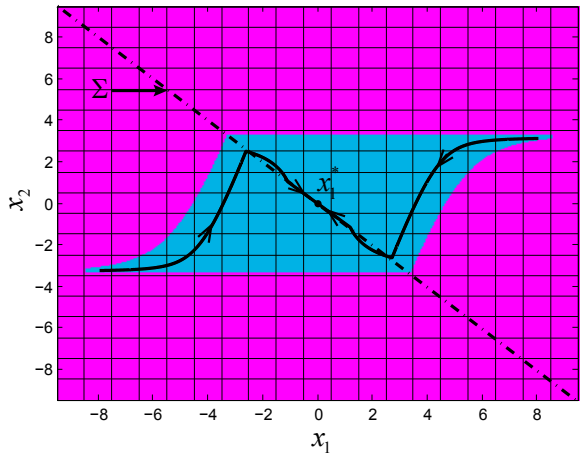

Figure 2: Basins of attraction for the sliding control problem (1) using ESCM, for an initial resolution of 19×19 grid of cells, initial cell size of $\delta_1 = 1$ (initial stages) and $\delta_2 = 0.083$ for the refinement stage. Black curves stand for trajectories inside of BA of x_1^*, for initial conditions $(x_1, x_2) = (8, 3.3)$ and $(x_1, x_2) = (-8, -3.3)$ while the dash-line indicates the switching manifold.