RAMAS: Software Package for Reachability Analysis of Multi-Affine Systems

Authors: Marius Kloetzer and Calin Belta  
Center for Information and Systems Engineering  
Boston University

RAMAS can be used to study reachability and safety properties of systems with multi-affine dynamics, which are described by:

 (1)

where  is a polynomial in the indeterminates  with the property that the degree of  in any of the variables is less than or equal to 1. Stated differently,  has the form:

 (2)

with  for all  and using the convention that if , then .

Systems of type (1) are widely encountered in practice. For example, they describe the dynamics of biochemical networks when chemical reactions with unitary stoichiometric coefficients are modeled using mass action kinetics. Multi-affine dynamics are also found in other systems, including the celebrated Euler’s equations for angular velocity of rotation of rigid bodies, the equations of motion of translating and rotating rigid bodies with rotation parameterized by quaternions, Volterra, and Lotka-Volterra equations. Of course systems of type (1) include systems with linear and affine dynamics as well.

The algorithm is based on iterative partitioning of the state space using rectangular regions and on exploiting some interesting convexity properties of multi-affine functions on rectangles. The theory and algorithms were developed in:

C. Belta and L.C.G.J.M. Habets. Controlling a class of nonlinear systems on rectangles.*IEEE Transactions on Automatic Control*, 2006.

M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. In: Lecture Notes in Computer Science, Springer Berlin / Heidelberg, vol. 3927, pp. 348 – 362, 2006

**Download and usage:**

The Matlab scripts can be [downloaded here](https://calinbelta.com/wp-content/uploads/2023/12/RAMAS.zip).

INSTRUCTIONS: **Please cite the above two papers if you use our software.** The program takes as input the dimension *n*, a closed full dimensional rectangle *X* describing the region of interest in the Euclidean ***R****n*, the coefficients  of a multi-affine vector field  as in equation (2), and the initial and final sets  and , given as unions of open sub-rectangles of arbitrary order in *X*. The tool returns either a positive answer if there are no trajectories of the continuous system from  to , or a subset of  which is guaranteed to be safe with respect to .

**Case studies:**

Our trials show that the algorithm works even for . We present some examples for the planar case () so we can show illustrative pictures.

We first consider a nonlinear multi-affine vector field (Case Study 1). We then focus on three linear systems (*i.e.,*) (Case Studies 2, 3, 4), which are of course particular cases of multi-affine systems. The qualitative phase portraits for such planar linear systems are known, and reachability properties are almost intuitive. Applying our method to such systems gives us some idea on the conservativeness  
of our approach.

A graph of arrows and lines

Description automatically generated with medium confidence  
Figure 1. Case Study 1: multi-affine vector field , ,  on , initial set  (blue), final set  (yellow), and initial partition induced by initial and final sets.

**Case Study 1 (nonlinear multi-affine system):**

Consider ,  with , and . The initial set is , which can be written as the union of two zero-order open rectangles ,  and one first-order  
open rectangle . The final set is , which in the initial partition can be seen as the union of 6 zero-order open rectangles, 7 first-order open rectangles, and 2 second-order open rectangles. In figure 1, we plot the vector field  everywhere in  and the two curves  and . Note that the two curves intersect inside . Therefore, the refinement procedure will not terminate. At each iteration, the algorithm will produce strictly shrinking over-approximations of  in , which lead to strictly growing safe sub-regions in .

The results produced at different iterations are shown in Figure 2, where it can be seen that the safe region strictly increases with the number of refinement iterations.

|  |  |
| --- | --- |
| A yellow rectangle with green squares and blue lines  Description automatically generated | A yellow and green rectangular object with lines and dots  Description automatically generated |
| A yellow and green rectangle with black text  Description automatically generated | A green and yellow graph  Description automatically generated |

Figure 2. Case Study 1: iterations 1, 2, 3, and 10 from our safety verification algorithm.

The growing green area represents the safe sub-region  of .

|  |  |
| --- | --- |
| A graph of a function  Description automatically generated | A green grid with black arrows  Description automatically generated |
| (a) | (b) |
| A graph of a line graph  Description automatically generated | A graph of a graph with a curved arrow  Description automatically generated with medium confidence |
| (c) | (d) |
| A graph of a function  Description automatically generated | A diagram of a graph  Description automatically generated |
| (e) | (f) |

Figure 3.> Reachability analysis for linear vector fields: (a), (c), and (e) show vector fields for which the origin is a stable node, stable focus, and unstable node, respectively.

The equations of the straight lines are  and  The green regions are safe sets, while the white regions are over-approximations of reachable sets.

**Case Study 2 (linear system, stable node):**

Consider the planar linear system  with ,  in rectangle . The origin is a (globally asymptotically) stable node for the system. The vector field is plotted in Figure 3 (a), together with the lines  and  and the initial set . Figure 3 (b) shows an over-approximation of the set reached from  (white region) and a safe set (green region). The straight dashed lines show the directions of the eigenvectors. We also plotted the trajectories starting from the extremities of . Since the system is linear, it is known that the closed segment  will remain a closed segment while flowing along the vector field. Therefore, the set reached from  roughly looks like the area between the trajectories of the extremities of , as shown in Figure 3 (b). Our method however returns the white region in 4 iterations. More iterations will not shrink the white region dramatically – it will only remove small white chunks North-East from the origin.

**Case Study 3 (linear system, stable focus):**

The difference between this case and Case Study 2 is that the vector field (shown in Figure 3 (c)) is , , for which the origin is a stable focus. As it can be seen in Figure 3 (d), the conservativeness of our method is more obvious in this case. Also, in this case the refinement algorithm terminates in 3 iterations.

**Case Study 4 (linear system, unstable node):**

Consider the same rectangular region and the planar linear vector field , , for which the origin is an unstable node (saddle). The vector field is plotted in Figure 3 (e), together with the initial set  and the two lines  and , which intersect at the origin. The over-approximation of  calculated in 4 iterations by our method is shown as the white region in Figure 3 (f), together with the eigenvectors and some illustrative trajectories. It can be seen that in this case our results are not very conservative. Note that the refinement does not terminate – it continues in a small region North-West from the origin. However, the result does not change significantly with the number of iterations.

As a conclusion to Case Studies 2, 3, and 4, our method produces conservative results when the trajectories loop around an equilibrium