TIRA: Toolbox for Interval Reachability Analysis

User manual (work in progress)

Pierre-Jean Meyer, Alex Devonport, Murat Arcak

1 Overview

TIRA is a Matlab library gathering into a unified framework several methods for the computation of interval over-approximations of the finite time reachable set of a continuous- or discrete-time system described by

$$\dot{x} = f(t, x, p),$$
$$x^+ = F(t, x, p),$$

with time $t \in \mathbb{R}$, state $x \in \mathbb{R}^{n_x}$ and input $p \in \mathbb{R}^{n_p}$. The theory of the 5 main methods currently implemented in TIRA is described in HSCC'19 tool paper available at `https://arxiv.org/pdf/1902.05204.pdf`

This document provides the basic instructions for the main uses of TIRA:

- Test one or more reachability methods on the provided example systems.
- Define your own system and associated reachability problem, and call one of the implemented methods on it.
- Define your own reachability method and add it to the toolbox.

2 Installation and requirements

2.1 Requirements

The tool TIRA is implemented in Matlab. It is advised to have a Matlab version 2015b or superior (it has been tested on both Matlab versions 2015b and 2018a). TIRA requires the Matlab 'Optimization toolbox' (this toolbox is only used in function `Utilities\Sensitivity_falsification.m`).

Matlab internal tools also report possible use of the following toolboxes, although to the best of our knowledge they should not be required:

- Communications Toolbox
- Curve Fitting Toolbox
- Simulink Control Design
- Statistics and Machine Learning Toolbox
- Computer Vision System Toolbox

2.2 Installation instructions

TIRA can be downloaded from the public repository available at `https://gitlab.com/pj_meyer/TIRA`. To use it,

- either set Matlab's *current folder* to the main `TIRA` folder and work directly in this folder,
- or work from a remote folder by adding the downloaded `TIRA` folder and all its subfolders to the Matlab search path. For this, you can use the command `addpath(genpath('full_path_to_TIRA_folder'))`

3 Tool architecture

The overall architecture of the main `TIRA` folder and the toolbox is described below.

- `System_description.m` is where the user defines their own system (either continuous-time or discrete-time)
- `MAIN_CALL.m` is where the user defines the reachability problem to be solved (intervals of initial states and inputs, time range) and calls the hub function `OA_methods/TIRA.m`
- `Solver_parameters.m` allows the user to choose which reachability method to apply (if none is chosen, TIRA will automatically pick the most suitable one) and modify the internal solver parameters.
- folder `Input_files/` contains skeleton files that can be filled by the user to give further useful information about the considered system (e.g. information about the Jacobian matrices)
- folder `OA_methods/` contains the hub function `OA_methods/TIRA.m` as well as one function for each of the 5 reachability methods currently implemented in TIRA. Users do not need to modify anything in this folder.
  - `OA_methods/TIRA.m` receives as input the definition of the reachability problem. If a specific reachability method is requested (as input argument to the function, or by modifying `Solver_parameters.m`), this function checks if the requirements for this methods are satisfied by the system and the user-provided informations in folder `Input_files/`, and then calls the corresponding over-approximation method (or returns an error if the requirements are not satisfied). If no specific method is requested, this function iteratively checks the requirements of each over-approximation method, and calls the first one whose requirements are satisfied.
- folder `Utilities/` contains various functions that are called by the hub function `OA_methods/TIRA.m` or one of the reachability methods. Users do not need to modify anything in this folder.
- folder `Examples/` contains a copy of some of the main files (`System_description.m`, `MAIN_CALL.m`, `Solver_parameters.m` and all files from folder `Input_files/`) already filled with the reachability problems for 12 examples. See Section 4 to use this folder.

4 Tool use on one of the provided example

To run one of the 5 reachability methods on one of the provided example systems, follow the steps below.

- Set the `Examples` folder as Matlab's new *current folder*. (All files listed below are those in this `Examples` subfolder.)
- Open function `System_description.m` which lists and describes the 12 available system examples currently implemented in TIRA. Look at the `switch/case` number on variable `system_choice` corresponding to the system you want to use.
- Open file `MAIN_CALL.m` and in section "Choice of the example system" (top of the file), uncomment the line `system_choice = ...` corresponding to your system (and comment the others).
- In section "Definition of the reachability problem (for each example system)", look up the `switch/case` number of your system and modify the definition of the reachability problem (`x_low`, `x_up` for the interval of initial states, `p_low`, `p_up` for the interval of inputs, `t_init`, `t_final` for the time range of the reachability problem) as desired (or use the provided default values).
- Then, two choices are available.
  - Either run the script `MAIN_CALL.m` directly. This will let TIRA go through each available methods until it finds one whose requirements are satisfied, then apply it. It will not look for other methods after one is found.
  - If you want to request TIRA to apply a specific method (among the ones presented in Section 3 of the HSCC paper), open file `Solver_parameters.m`, and replace `parameters.OA_method = NaN;` by the number of the method you want to use (the file details which method corresponds to each number). This file can also be used to request a specific submethod or change the internal solver parameters (details are provided in the comments of the file). Then run the script `MAIN_CALL.m`. Since in this case you requested a specific method, TIRA will return an error if the chosen system does not satisfy the requirements for the chosen method.

This call will display in the Matlab console the method(s) tried by TIRA and their running times. If a method is successfully applied, it will also plot the resulting interval alongside an estimation of the actual reachable set (obtained from computing the successors of 1000 random initial states). This plotting is only enabled for systems whose state dimension is between 2 and 20.

If instead of the file `MAIN_CALL.m`, you follow the same procedure using file `MAIN_CALL_OA_comparison.m`, then all main over-approximation methods will be tried at once on the same problem. The methods which cannot be applied to the considered system and reachability problem are ignore, and the resulting over-approximations of all other methods are plotted in the same graph.

5 General tool use on user-provided system

A more general use of the tool TIRA involves the user providing their own system description and problem definition. While the use of TIRA in such case is similar to what is described in Section 4 above, there are some differences highlighted below.

- Set the main `TIRA` folder as Matlab's new *current folder*. (All files listed below are those in this `TIRA` folder.)
- Open function `System_description.m` and provide your own system description (this file can be used for both continuous-time and discrete-time systems).
- Provide additional information on the system in one or more of the 7 files in subfolder `Input_files` (e.g. known signs or bounds for the Jacobian matrices). This step is optional but recommended since most methods implemented in TIRA rely on at least one of these user-provided information. The list of requirements for each methods implemented in TIRA is provided in Section 3 of the HSCC paper. If the user does not know which information to provide or which reachability method to use, we advise to provide upper and lower bounds on the Jacobian matrices (in file `Input_files/UP_Jacobian_Bounds.m`) which is the most versatile and can be exploited in most reachability methods (although some of the methods have additional requirements that will be automatically checked for satisfaction by TIRA).
- Open file `MAIN_CALL.m` and define the reachability problem that you want to solve (`x_low`, `x_up` for the interval of initial states, `p_low`, `p_up` for the interval of inputs, `t_init`, `t_final` for the time range of the reachability problem). `t_final` is only needed for continuous-time systems.
- In `MAIN_CALL.m`, section "Call of the main over-approximation function" (shown below), uncomment the call of the `TIRA` function corresponding to the type of system you defined (continuous-time or discrete-time).

```
% Call for continuous-time systems over the time range [t_init,
    t_final]
% [succ_low,succ_up] = TIRA([t_init,t_final],x_low,x_up,p_low,p_up)
    ;

% Call for discrete-time systems for one step starting from time
    t_init
% [succ_low,succ_up] = TIRA(t_init,x_low,x_up,p_low,p_up);
```

- Then, the same two choices as in Section 4 are available: running the script `MAIN_CALL.m` directly to let the tool pick a method; or request a specific method by modifying the file `Solver_parameters.m`.

This call will display in the Matlab console the method(s) tried by TIRA and their running times. To obtain plots as in Section 4, copy the last two sections of `Examples/MAIN_CALL.m` to the end of `MAIN_CALL.m` (you also need to define beforehand variable `bool_discrete_time` to 0 if you have a continuous-time system or to 1 for a discrete-time system).

6 Adding your own reachability methods

The architecture of TIRA sketched in Section 3 was chosen to allow users to easily extend it with additional interval-based reachability methods. For this, we suggest to follow the procedure below.

- Define the main function of your over-approximation method in a separate file to be added in the folder `OA_methods/`. If this method needs to call any side functions, you can add them in folder `Utilities/`.
- In `Solver_parameters.m`, give a number to your method, and optionally add the default values for its internal parameters.
- In the hub function `OA_methods/TIRA.m`, add the call to this function taking inspiration from the format of the existing 5 methods. If this new reachability method has some requirements, make sure that all these requirements are checked within `OA_methods/TIRA.m` before the call of your function.