Verification of Continuous Dynamical Systems by Timed Automata

Christoffer Sloth and Rafael Wisniewski
Department of Computer Science and Section for Automation & Control, Aalborg University, Denmark

MT-LAB
A VKR CENTRE OF EXCELLENCE

Introduction

The verification of properties is important for most systems, to ensure that they operate in consistence with the specification, and operate safely when implemented.
The verification of discrete systems such as finite automata is mature, but methods for verification of continuous and hybrid systems are still limited. The limiting factor in the verification of both continuous and hybrid systems is related to the over-approximation of the continuous dynamics of the models [1]. Additionally, some methods require explicit solutions of the differential equations that describe the continuous dynamics, which are usually unknown.
We propose a method for abstracting continuous systems by timed automata using a partition of the state space generated from Lyapunov functions. It is chosen to abstract continuous systems by timed automata, since tools for efficient verification of such models exist. Additionally, Lyapunov functions are chosen for generating the partition, since their sub-level sets are positive invariant; hence, it is possible to determine a priori if the abstraction is sound or complete.

Method

The proposed method is intended for the verification of autonomous continuous systems $\Gamma = (X, f)$, with state space $X \subseteq \mathbb{R}^n$ and dynamics described by ordinary differential equations

$$\dot{x} = f(x). \tag{1}$$

The abstraction is based on partitioning the state spaces using invariant sets, which are generated by sub-level sets of Lyapunov functions. The set-differences of the positive invariant sets are slices as shown in Fig. 1.

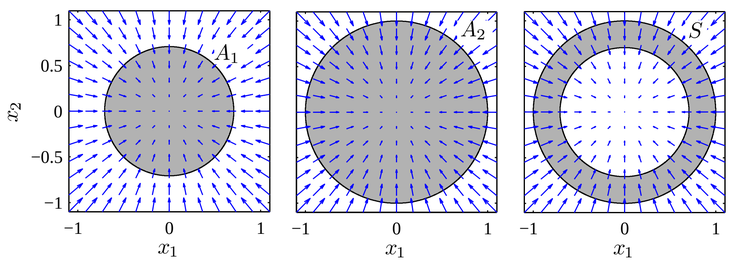

Figure 1: Phase plot of a dynamical system (blue arrows). The slice S is the set-difference of A_1 and A_2, i.e. $S = \mathrm{cl}(A_2 \backslash A_1)$.

Cells are generated by intersecting slices as shown in Fig. 2. From the figure it is seen that the intersection of the slices forms more than one closed component (gray area in Fig. 2), but each of the closed components is said to be a cell.
The timed automaton abstracting the continuous system Γ is generated by associating each cell of the partition with a location of the timed automaton. The time information for the timed automaton is determined

Method (continued)

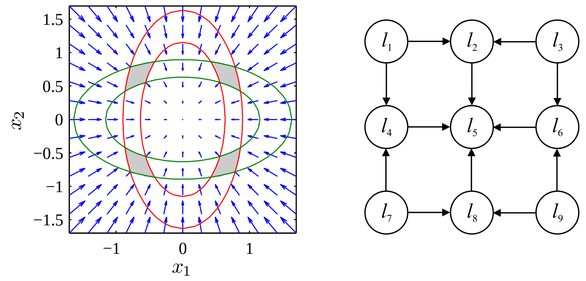

Figure 2: Partition of a state space using two different Lyapunov functions (red and green). Each cell is associated to a location of the timed automaton.

based on the time it takes to traverse the slices generating the cells, as in [2]. This time is calculated based on the derivative of the Lyapunov function; i.e., the solutions to the differential equations are not utilized.
A class of abstractions can be composed into multiple separate timed automata. This allows parallelization of the verification process, which is beneficial for systems of high dimensionality. The principle of the compositionality is illustrated in Fig. 3, where a partition generated by two Lyapunov functions and partitions generated using one of the Lyapunov functions are plotted.

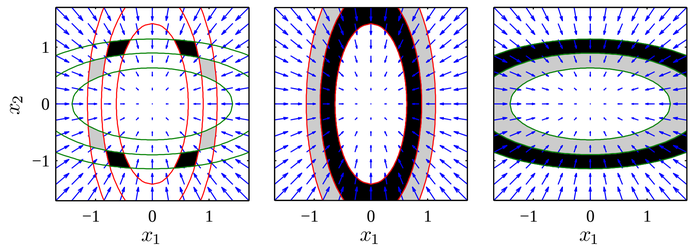

Figure 3: The left subplot shows a partition generated by two Lyapunov functions (red and green). The middle and right subplot show partitions generated using only one Lyapunov function. Black areas are unsafe cells and gray areas are initial cells.

The timed automaton generated by the partition to the left is safe if one of the timed automata generated by the middle or right partition is safe.

Conditions for the Partitioning

Conditions for the partition generating a timed automaton have been set up to guarantee soundness, completeness, and refinability. These properties are important according to the following:

1. If a sound abstraction $\mathcal{A}$ is safe then Γ is also safe.
2. If a complete abstraction $\mathcal{A}$ is safe (unsafe) then Γ is also safe (unsafe).
3. A complete and refinable abstraction $\mathcal{A}$ can approximate the reachable states of Γ arbitrarily close.

Results

Sound and complete abstractions of a simple dynamical system are conducted, and their reachable sets are compared. We consider the system

$$\dot{x} = Ax$$

with

$$A = \begin{bmatrix} -3 & -1 \\ -2 & -5 \end{bmatrix} \text{ and } X_0 = [1.5, 2] \times [-9.5, -10].$$

In both the sound and the complete abstraction, the state space is partitioned utilizing two different Lyapunov functions i.e. two slice-families, consisting of 10 sub-level sets each. From this, both abstractions result in a timed automaton with 361 locations.

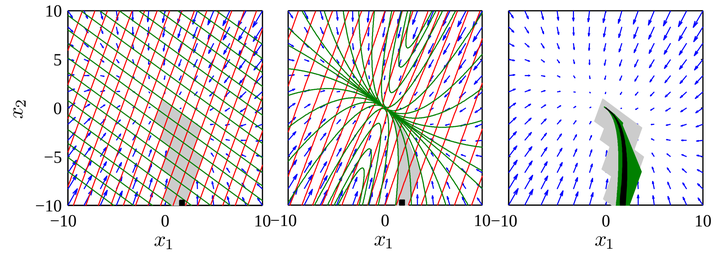

Figure 4: The right subplot is a sound abstraction and the middle a complete abstraction. The gray areas are the approximated reachable sets.

The reachable set (black) is approximated closer by the complete abstraction (green), than by the sound abstraction (gray).

Conclusion

The proposed abstraction is based on partitioning the state space of the dynamical systems by set-differences of positively invariant sets.

- Conditions for soundness completeness, and refinability are derived.
- The abstraction can be obtained as a parallel composition of multiple timed automata under certain conditions.
- An a priori upper bound on the over-approximation of the reachable set can be determined for complete abstractions.
- Sound (complete) and refinable abstractions for hyperbolic Morse-Smale (linear) systems exist.

The method should be extended to allow controller design e.g. by generating a timed game, for which automatic controller synthesis is possible.

References

[1] H. Gueguena, J. Zaytoon, *On the formal verification of hybrid systems*, (2004).
[2] Maler, O., Batt, G.: *Approximating continuous systems by timed automata*, (2008).