MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 28, NO. 4, 2021
journal homepage: www.mais-journal.ru

THEORY OF COMPUTING

Notes on Recent Achievements in Proving Stability using KeYmaeraX

T. Baar[1], H. Schulte[1]

DOI: 10.18255/1818-1015-2021-4-326-336

[1]HTW Berlin, 75A Wilhelminenhofstraße, Berlin 12459, Germany.

MSC2020: 68N30
Research article
Full text in English

Received November 15, 2021
After revision December 1, 2021
Accepted December 8, 2021

KeYmaeraX is a Hoare-style theorem prover for hybrid systems. A hybrid system can be seen as an aggregation of both discrete and continuous variables, whose values can change abruptly or continuously, respectively. KeYmaeraX supports only variables having the primitive type bool or real.

Due to the mixture of discrete and continuous system elements, one promising application area for KeYmaeraX are closed-loop control systems. A closed-loop control system consists of a plant and a controller. While the plant is basically an aggregation of continuous variables whose values change over time accordingly to physical laws, the controller can be seen as an algorithm formulated in a classical programming language.

In this paper, we review some recent extensions of the proof calculus applied by KeYmaeraX that make formal proofs on the stability of dynamic systems more feasible. Based on an example, we first introduce to the topic and prove asymptotic stability of a given system in a hand-written mathematical style. This approach is then compared with a formal encoding of the problem and a formal proof established in KeYmaeraX. We also discuss open problems such as the formalization of asymptotic stability.

Keywords: cyber physical system; control theory; lyapunov function; imperative programming language

INFORMATION ABOUT THE AUTHORS

Thomas Baar correspondence author	orcid.org/0000-0002-8443-1558. E-mail: thomas.baar@htw-berlin.de Professor.
Horst Schulte	orcid.org/0000-0001-5851-3616. E-mail: horst.schulte@htw-berlin.de Professor.

For citation: T. Baar and H. Schulte, “Notes on Recent Achievements in Proving Stability using KeYmaeraX”, *Modeling and analysis of information systems*, vol. 28, no. 4, pp. 326-336, 2021.

Замечания о последних достижениях в доказательстве устойчивости с использованием KeYmaeraX

Т. Баар[1], Х. Шульте[1]

DOI: 10.18255/1818-1015-2021-4-326-336

[1]Университет прикладных технических и экономических наук г. Берлина, ул. Вилхелминенхофштрассе, 75A, г. Берлин 12459 Германия.

УДК 004.942
Научная статья
Полный текст на английском языке

Получена 15 ноября 2021 г.
После доработки 1 декабря 2021 г.
Принята к публикации 8 декабря 2021 г.

KeYmaeraX – это доказательство теорем в стиле Хоара для гибридных систем. Гибридную систему можно рассматривать как совокупность дискретных, так и непрерывных переменных, значения которых могут изменяться резко или непрерывно соответственно. KeYmaeraX поддерживает только переменные, имеющие примитивный тип bool или real.

Благодаря сочетанию дискретных и непрерывных элементов системы, одной из перспективных областей применения KeYmaeraX являются системы управления с замкнутым контуром. Система управления с замкнутым контуром состоит из установки и контроллера. В то время как установка в основном представляет собой совокупность непрерывных переменных, значения которых меняются со временем в соответствии с физическими законами, контроллер можно рассматривать как алгоритм, сформулированный на классическом языке программирования.

В этой статье мы рассмотрим некоторые недавние расширения исчисления доказательств, применяемые KeYmaeraX, которые делают формальные доказательства устойчивости динамических систем более выполнимыми. Основываясь на примере, мы сначала познакомимся с темой и докажем асимптотическую устойчивость данной системы.

Ключевые слова: киберфизическая система; теория управления; функция Ляпунова; императивный язык программирования

ИНФОРМАЦИЯ ОБ АВТОРАХ

Томас Баар автор для корреспонденции	orcid.org/0000-0002-8443-1558. E-mail: thomas.baar@htw-berlin.de Профессор.
Хорст Шульте	orcid.org/0000-0001-5851-3616. E-mail: horst.schulte@htw-berlin.de Профессор.

Для цитирования: T. Baar and H. Schulte, “Notes on Recent Achievements in Proving Stability using KeYmaeraX”, *Modeling and analysis of information systems*, vol. 28, no. 4, pp. 326-336, 2021.

1. Introduction

The verification of software has made huge progress over the last 20 years, but is still considered to be challenging [1]. Many software verification systems such as KeY, FramaC, VeriFast, Dafny and others rely on a classical Hoare-style calculus [2] for verifying pre-/post-conditions.

The theorem prover KeYmaeraX and its underlying Differential Dynamic Logic [3] were successful in extending a traditional Hoare-style calculus with capabilities to reason also on the dynamics of continuous functions, which are specified by ordinary differential equations (ODEs). More precisely, KeYmaeraX allows to verify so-called *hybrid programs*[1] to be correct in the sense of fulfilling given pre-/post-condition contracts. The language of hybrid programs is a very simple traditional imperative while-language with *assignment*, *sequential composition*, *conditional execution*, and *iteration* as basic programming constructs. In addition, there is support for non-determinism (non-deterministic choice, non-deterministic iteration) and a very special construct bridging the gap to evolving functions called *evolving state*. More details on KeYmaeraX and its supported syntax can be found in tutorial [4]. In [5, 6], we have analyzed some deficiencies of the input syntax and present a number of introductory examples.

Control theory is an engineering discipline aiming at analyzing dynamic systems in general. A dynamic system is modeled by a set of state variables, which typically change their value continuously over time. In many cases, this change is described by ordinary differential equations (ODEs).

Well understood and widely applied are linear dynamic systems, whose dynamic can be described as $\dot{x} = Ax$, where x is a vector of *state variables*, A the so-called *system matrix*, and $\dot{x}$ denotes the *derivation of* x. The stability of linear systems is well-understood and proving it for a given concrete system usually requires few mathematical arguments: Common techniques are finding a *Lyapunov function* [7, 8] or analyzing the system matrix A together with *its eigenvalues*.

When modeling real world processes, pure linear systems are often not sufficient. However, the overall system can be modeled in many cases as a system switching through multiple linear subsystems [9]. The switching condition can be simple (as in our running examples shown below) or more elaborate. In any case, we can formulate such conditions using an imperative programming language.

While classical control theory is very successful in analyzing pure linear systems, its mathematical approaches do not work well when applied to composed system. If the composition scheme is restricted to few switching schemes, the mathematical analysis remains feasible but becomes cumbersome[9]. A more elegant solution for this problem could be to encode the composition rules for aggregating the resulting system in terms of an imperative programming language. To analyze such composed systems, one would obviously need a notion of the underlying semantics of the used programming language.

The theorem prover KeYmaeraX is able to capture both the continuous dynamics of systems as well as algorithms for switching the current mode. We will review in this paper some recent achievements that will make the formal verification of system stability more feasible. We also discuss some shortcomings of the dynamic logic and the proof calculus underlying KeYmaeraX.

This paper is organized as follows: Section 2 presents two linear systems and then two different compositions of it. Though the composition algorithms differ only very slightly, the two resulting systems differ dramatically. One of them remains stable, the other one becomes instable. In Section 2, we also give a mathematical proof for the stability of the first composed system. In Section 3, we provide an encoding of the running examples for KeYmaeraX and outline the formal proof within KeYmaeraX. We also report on yet unresolved obstacles such as the formulation of the verification goal (asymptotic stability of the overall system) within the Hoare-style logic supported by KeYmaeraX. In Section 4 we review related work and Section 5 concludes the paper.

[1] There is a graphical version of these programs (basically the Control Flow Graph) called *hybrid automaton*.

2. Running Example

Given is a periodic linear time invariant (LTI) dynamic system in state space form [10]

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 \\ -a & 0 \end{pmatrix} x(t) = A\,x(t)\,, \qquad x(t=0) = x_0 \tag{1}$$

with the state vector $x = (\,x_1\,,\,x_2\,)^T$, the system matrix A, the parameter $a \in \mathbb{R}_0$ and the initial condition as $x_0 \in \mathbb{R}^2$. Based on the theory of ordinary differential equations (ODEs) [11], the periodicity of the solution of (1) with

$$x(t) = x(t+T)\,, \quad T = \frac{2\pi}{\sqrt{a}} \tag{2}$$

is related to the eigenvalues of A denoted as $\lambda = \mathrm{eig}(A) \in \mathbb{C}^2$ with $\lambda_1 = -j\sqrt{a}$ and $\lambda_2 = +j\sqrt{a}$. Let us now consider two systems

$$Sys1 \equiv \dot{x}(t) = \underbrace{\begin{pmatrix} 0 & 1 \\ -a_1 & 0 \end{pmatrix}}_{A_1} x(t)\,, \qquad Sys2 \equiv \dot{x}(t) = \underbrace{\begin{pmatrix} 0 & 1 \\ -a_2 & 0 \end{pmatrix}}_{A_2} x(t)\,, \tag{3}$$

where the parameters a_i for $i = 1, 2$ satisfy the condition

$$0 < a_1 < 1 < a_2 \;\;. \tag{4}$$

For illustration, we consider the numerical examples of the form (3) satisfying (4) with

$$a_1 = \frac{1}{4}\,, \qquad a_2 = \frac{9}{4}\,.$$

Figure 1 and 2 show the particular solution and phase diagram for the initial value $x_0 = (\,0\,,\,3\,)^T$. The analytically determined period in (2) of

$$T_1 = \frac{2\pi}{\sqrt{a_1}} = 4\,\pi\,, \qquad T_2 = \frac{2\pi}{\sqrt{a_2}} = \frac{4\,\pi}{3}$$

corresponds to the numerical solution. Let us now examine the dynamics of a system which switches between the two systems (3) as a function of the current state x. For this purpose, the switching function

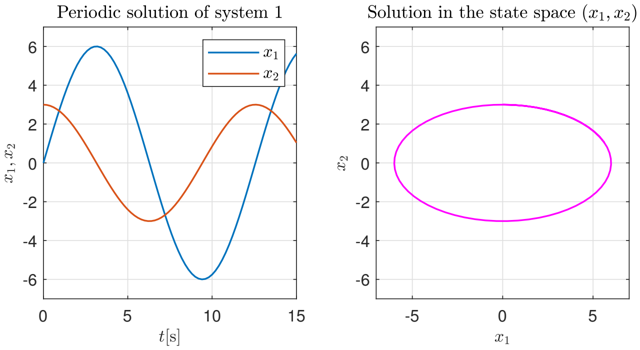

Fig. 1: Particular solution $x(t)$ and in the state space of $Sys1$ for $x_0 = (\,0\,,\,3\,)^T$

$$switch(x) = x_1\,x_2 \tag{5}$$

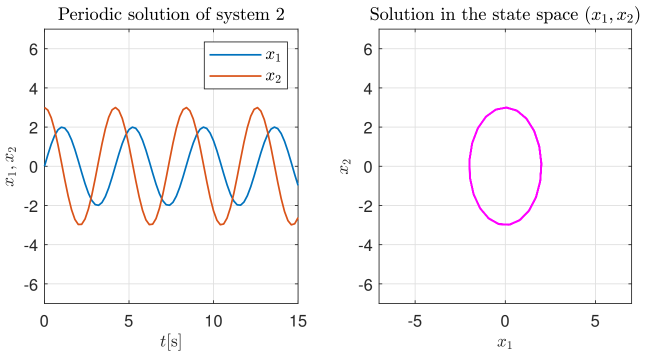

Fig. 2: Particular solution $x(t)$ and in the state space of $Sys2$ for $x_0 = (\,0\,,\,3\,)^T$

is defined to divide the state space into two non-convex subspaces

$$X_1 = \{\, x(t) \,|\, switch(x) < 0 \,\}\,, \qquad X_2 = \{\, x(t) \,|\, switch(x) \geq 0 \,\}\,. \tag{6}$$

Based on these definitions, two variants of a switching system can now be specified. The first switching system $SwSys1$ is given as

$$SwSys1 \equiv \dot{x}(t) = \begin{cases} A_1\, x(t)\,, & switch(x) < 0 \\ A_2\, x(t)\,, & switch(x) \geq 0 \end{cases} \,. \tag{7}$$

The second system $SwSys2$ inverts the switching condition from the first system and is given as

$$SwSys2 \equiv \dot{x}(t) = \begin{cases} A_1\, x(t)\,, & switch(x) \geq 0 \\ A_2\, x(t)\,, & switch(x) < 0 \end{cases} \,. \tag{8}$$

Analogous to the previous LTI systems, the switching systems have the same initial condition x_0. The particular solution of $SwSys1$ (7) is given in Figure 3. One can clearly see that this system is asymptotic stable for the given initial condition, i.e. $\lim\limits_{t \longrightarrow \infty} x(t) \longrightarrow 0$ for $x_0 = (\,0\,,\,3\,)^T$. By quadrant switching, two periodic systems are joined to form an asymptotically stable system.

The opposite effect occurs in the second switching system $SwSys2$ (8) if the switching condition is reversed. Here, the amplitudes of $x_1(t), x_2(t)$ fastly grow and exceed any bounds. Note that the diagrams of Figure 4 have on their axis much greater values than all the other diagrams.

In order to understand the effect, it is worthwhile to analyze the phase diagrams for both $SwSys1$ and $SwSys2$ (cmp. Fig. 5). As one can see at the left part, system $SwSys1$ is enforced in the right-upper and left-lower quadrants (note that $switch(x) \geq 0$ holds) to have a greater value change for x_2 than for x_1. For example, when entering the right-upper quadrant, the value for x_1 is 0 and let v be the value for x_2. When leaving this quadrant, the value for x_2 is now 0 and x_1 has a value, but this is smaller than v. An analogous behavior we have in the right-lower and left-upper quadrant, where x_1 changes more than x_2. In summary, whenever the trajectory of $SWSys1$ crosses the diagram axes ($x_1 = 0$ or $x_2 = 0$), the non-zero value of the coordinate (alternating x_1, x_2) form a monotonically decreasing series.

For system $SwSys2$, the opposite is true: When crossing the axes, the non-zero component of the coordinate becomes larger. Thus, $SwSys2$ is not stable.

To sum up, combining two periodic systems $Sys1$, $Sys2$ to a switched system can result both in an asymptotic stable system $SwSys1$ or in an unstable system $SwSys2$. The difference in the definition of $SwSys1$ and $SwSys2$ is rather marginal. Therefore, it would be very helpful to have a verification tool able to check formally, whether the resulting system is stable or not.

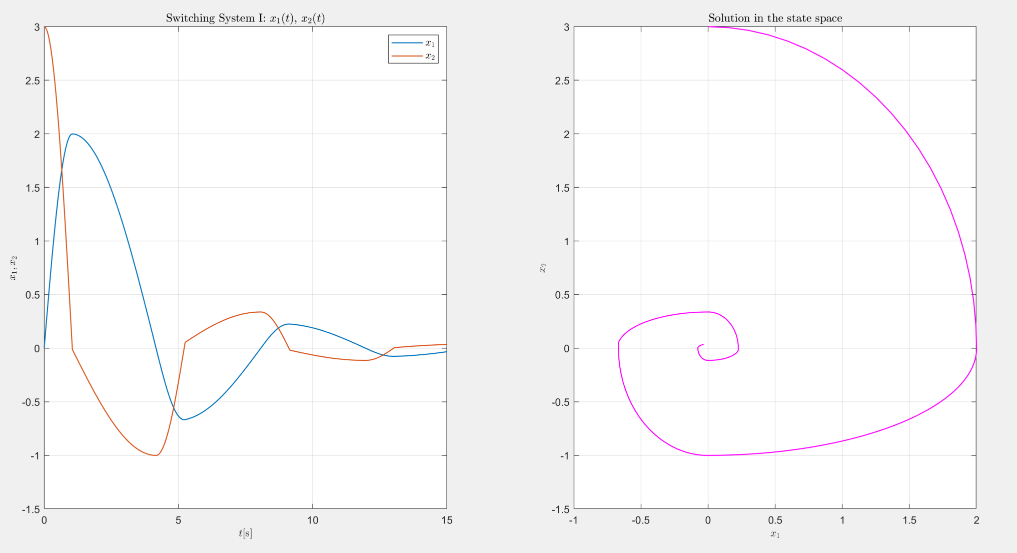

Fig. 3: Particular solution of $SwSys1$ (7) for $x_0 = (0\,,\,3)^T$

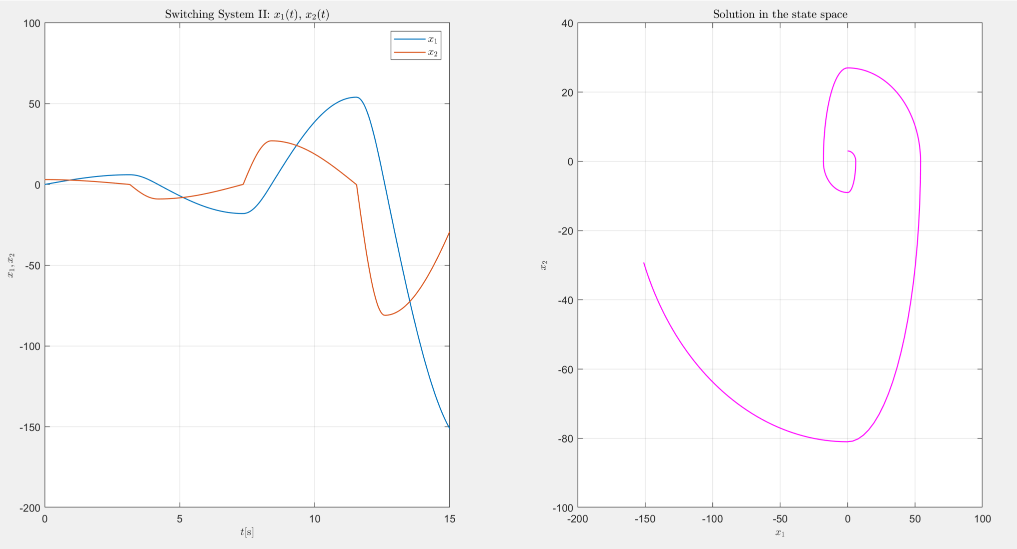

Fig. 4: Particular solution of $SwSys2$ (8) for $x_0 = (0\,,\,3)^T$

2.1. Lyapunov Stability Analysis

A system is called to be *stable*, iff the system will never leave an ϵ region around the origin (i.e. $|x| < \epsilon$) when started in an appropriate δ region ($|x| < \delta$). Note that δ can be freely chosen once ϵ has been fixed. Both δ and ϵ must be positive ($\epsilon > 0, \delta > 0$). A formalization of this definition is presented below in (16).

A system is called to be *asymptotic stable*, iff in addition $lim_{t\rightarrow\infty}x(t) = 0$ holds. Figure 6 shows a system, which is stable but not asymptotic stable.

We proceed with a mathematical proof for the stability of the first switching system $SwSys1$ using Lyapunov's direct method [7, 8, 12]. We start with considering one - heuristically chosen - Lyapunov function candidate for the original systems $Sys1, Sys2$

$$V(x) = x_1^2 + x_2^2 \qquad \forall x \neq 0 \tag{9}$$

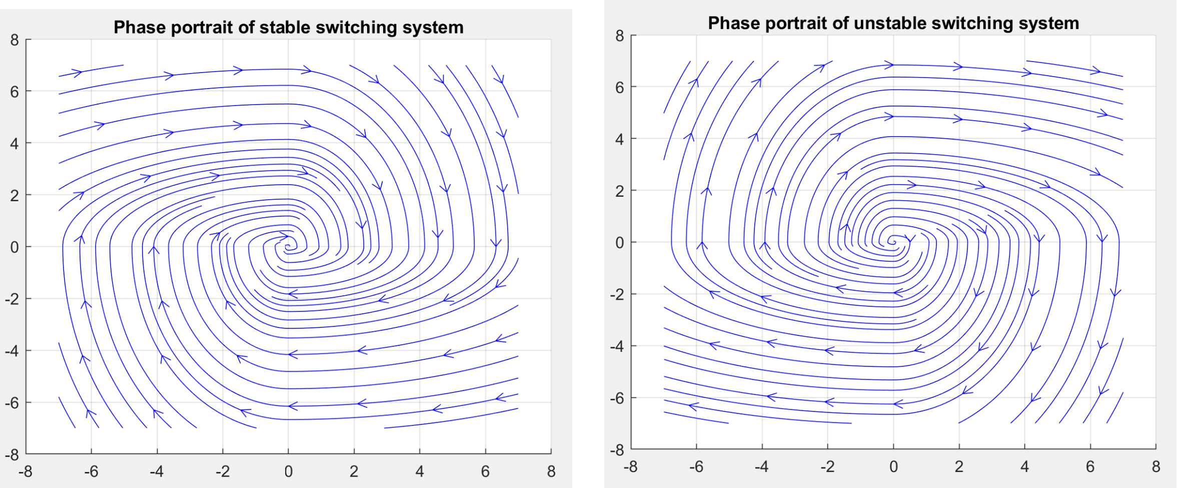

Fig. 5: Phase diagram for $SwSys1/SwSys2$

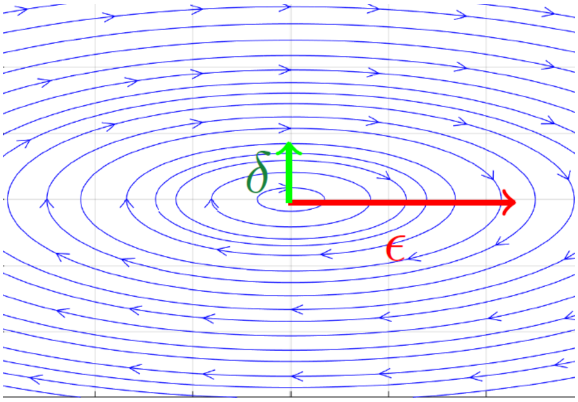

Fig. 6: Definition of Stability

The time derivative of the function $V(x)$ results in

$$\dot{V}(x) = 2\,x_1\,\dot{x}_1 + 2\,x_2\,\dot{x}_2\;. \tag{10}$$

Substituting $\dot{x}_1$ and $\dot{x}_2$ by the right hand side of the LTI systems in (3) with $0 < a_1 < 1 < a_2$ two Lyapunov function candidates are obtained for $Sys1, Sys2$:

$$\dot{V}_1(x) = 2\,x_1\,x_2\,(1-a_1)\,, \qquad \dot{V}_2(x) = 2\,x_1\,x_2\,(1-a_2) \tag{11}$$

Each separate function $\dot{V}_i(x)$ does not satisfy the requirement of a Lyapunov function with $\dot{V}_i(x) < 0.\ \forall x$. This is valid only segment-wise and with the introduced switching function (5) and switching condition (6) of x it follows that V_{SwSys1} is a Lyapunov function with

$$\dot{V_{SwSys1}} = \begin{cases} 2\,x_1\,x_2\,(1-a_1) < 0\,, & switch(x) \leq 0 \\ 2\,x_1\,x_2\,(1-a_2) < 0\,, & switch(x) \geq 0 \end{cases} \tag{12}$$

where $\dot{V_{SwSys1}} < 0$ holds for all $x \neq 0$. The switching condition in (12) corresponds to the one in system $SwSys1$ (7) which therefore proves the asymptotic stability.

3. Running Example for KeYmaeraX

In this section, we will present an elegant and succinct specification of both the linear and the switched system presented in the previous section using the input formalism of KeYmaeraX. In KeYmaeraX, a dynamic system is described in form of a program α.

As a first attempt, the linear system $Sys1$ might look as a program could as follows:

$$\alpha_{Sys1} \equiv \{x' = A_1 x\} \tag{13}$$

Unfortunately, KeYmaeraX does not allow the usage of state matrix A_1 since all constants and variables have to have a primitive type (real, bool). Also, the state vector x cannot be directly used and must be split into its components x_1, x_2. Thus, we have to rewrite our first attempt (13) as

$$\alpha_{Sys1} \equiv \{x1' = x2, x2' = -a_1 * x1\} \tag{14}$$

The system $Sys1$ can be described in KeYmaeraX by just one evolving state. For the system $SwSys1$, we combine the two evolving states representing $Sys1$ and $Sys2$ using the non-deterministic choice construct (operator ++) and enclose this by a non-deterministic iteration (operator ∗):

$$\begin{aligned} \alpha_{SwSys1} \equiv \quad & (\\ & \qquad \{x1' = x2, x2' = -a_1 * x1 \;\&\; switch(x) \leq 0\} \\ & ++ \\ & \qquad \{x1' = x2, x2' = -a_2 * x1 \;\&\; switch(x) \geq 0\} \\ &)* \end{aligned} \tag{15}$$

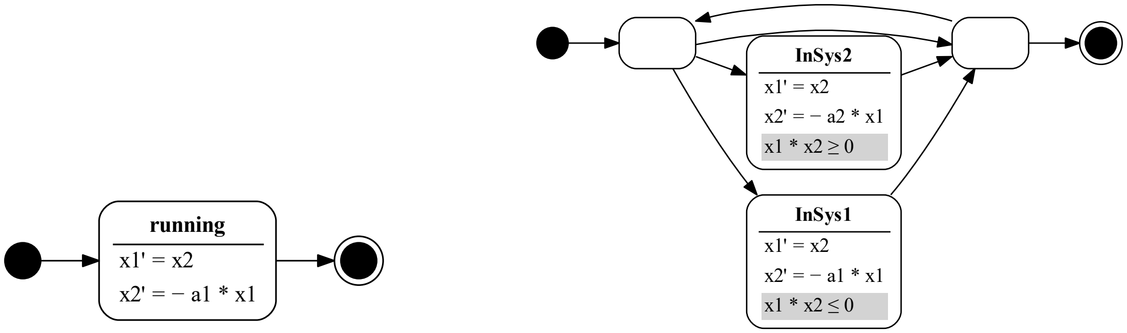

Fig. 7: System descriptions of $Sys1$, $SwSys1$ in KeYmaeraX (as hybrid automaton)

The graphical version of the programs (13) and (15) are shown in Figure 7.

3.1. Proving stability using KeYmaeraX

In terms of differential dynamic logic, the stability of system $SwSys1$ formulates as:

$$\forall \epsilon \, (0 < \epsilon \longrightarrow \exists \delta \, (0 < \delta \wedge x_1^2 + x_2^2 < \delta \longrightarrow [\alpha_{SwSys1}] \, x_1^2 + x_2^2 < \epsilon)) \tag{16}$$

Establishing a formal proof using KeYmaeraX for such a property of system $SwSys1$ remained for quite a long time rather a challenge. This was due to the fact that the proof's fundamental argument - the Lyapunov function $V(x) = x_1^2 + x_2^2$ - could not be encoded directly when establishing the proof.

However, Tan and Platzer report in their very recent paper [13], how the proof calculus has been recently conservatively extended, i.e. new proof rules have been derived and can be used now directly within proof tactics. One of the new proof rules looks as follows:

$$\mathsf{Lyap}_{>}\ \frac{\vdash f(0) = 0 \land v(0) = 0 \qquad \vdash \exists \gamma > 0 \forall x (|x|^2 \leq \gamma^2 \rightarrow v > 0 \land v' < 0)}{\vdash AStab(x' = f(x))}$$

The formal proof for the stability of the switched system $SwSys1$ has been published in [14] and is available online[2].

Please note, that (16) only formalizes stability but not asymptotic stability. To formulate asymptotic stability, one would need to encode a situation that is far in the future ($lim_{t\rightarrow\infty}$) as asymptotic stability means that there will be a point in time, after which the system will always remain within an ϵ region. How can we catch this point in time in a formula defining asymptotic stability? In order to formalize asymptotic stability, one has to choose - probably - a different form than the usual invariant $\phi \longrightarrow [\alpha]\psi$ that has been successfully applied for formalizing stability. This problem has to remain as an open question here. Note, however, that also the conclusion of [14] considers the formalization of asymptotic stability within KeYmaeraX as an unsolved problem and future work.

4. Related Work

In his landmark paper published in 1892 (see [7] for a French and [8] for an English translation), A.M. Lyapunov identifies and describes mathematical tools for analyzing dynamic systems. A Lyapunov function for a given system is an energy measure that has to be decreasing/non-increasing as the system evolves over time. Once an appropriate Lyapunov function is found, it can witness the stability of the system.

Much research has gone into finding suitable Lyapunov functions automatically. Some numerical approaches [15], [16] are based on sum-of-square programming techniques while other approaches exploit Gröbner basis [17], Lie derivatives [18], or constraint solving techniques [19]. Switched systems [20], [9], [21] can vary considerably in their switching mechanics (see also [14] for an overview). A stability proof for switched system often requires to find more sophisticated Lyapunov functions taken all different system modes into account. In [22], a more relaxed notion of stability is presented together with a verification methods based on model checking.

5. Conclusion

In this paper, we report on experiences we gained when merging verification techniques from two engineering disciplines: control theory and software engineering.

Control theory has developed numerous techniques to verify certain properties of modeled systems. One of the most important properties is stability. One fundamental verification technique is finding a Lyapunov function as an upper bound for the system's state change. When the Lyapunov function decreases monotonously, the system changes will become smaller and smaller over time and the system converts towards a stable point. Another often used verification technique for linear system is the analysis of the eigenvalues of the system matrix.

It becomes more and more popular in control theory to combine rather trivial systems by some glue code (written in an imperative programming language) in order to form more complex systems. Here, the behavior of the overall system depends also from the semantics of the used imperative programming language.

The traditional, math-based verification techniques usually fail to verify such composed systems, because they do not have a notion of programming constructs. KeYmaeraX is a theorem prover that covers both areas: continuously and abruptly evolving systems. Very recently, KeYmaeraX was extended by dedicated verification support for stability properties of dynamic systems. Thus, it is now possible to formally verify

[2] see `https://github.com/LS-Lab/KeYmaeraX-projects/blob/master/stability/switchedsystems.kyx`

the stability also of such systems, which are composed of subsystems and which switch between these subsystems. The switching decision can be any algorithm encoded by a simple imperative programming language.

From a practical point of view, the most urgent future work is to find an elegant encoding of the notion of asymptotic stability within the formalism supported by KeYmaeraX.

References

[1] P. Baudin, F. Bobot, D. Bühler, L. Correnson, F. Kirchner, N. Kosmatov, A. Maroneze, V. Perrelle, V. Prevosto, J. Signoles, and N. Williams, "The dogged pursuit of bug-free C programs: the Frama-C software analysis platform", *Communications of the ACM (CACM)*, vol. 64, no. 8, pp. 56–68, Aug. 2021.

[2] C. A. R. Hoare, "An Axiomatic Basis for Computer Programming", *Commun. ACM*, vol. 12, no. 10, pp. 576–580, 1969.

[3] A. Platzer, *Logical Foundations of Cyber-Physical Systems.* Springer, 2018, ISBN: 978-3-319-63587-3.

[4] J.-D. Quesel, S. Mitsch, S. Loos, N. Aréchiga, and A. Platzer, "How to Model and Prove Hybrid Systems with KeYmaera: A Tutorial on Safety", *STTT*, vol. 18, no. 1, pp. 67–91, 2016. DOI: 10.1007/s10009-015-0367-0.

[5] T. Baar and S. Staroletov, "A Control Flow Graph Based Approach to Make the Verification of Cyber-Physical Systems Using KeYmaera Easier", *Modeling and Analysis of Information Systems. 2018;25(5)*, pp. 465–480, 2018.

[6] T. Baar, "A Metamodel-Based Approach for Adding Modularization to KeYmaera's Input Syntax", in *PSI, Novosibirsk*, Springer, 2019, pp. 125–139. DOI: 10.1007/978-3-030-37487-7_11.

[7] A. Liapounoff, "Problème général de la stabilité du mouvement", *Annales de la faculté des sciences de Toulouse*, vol. 9, no. 2, pp. 203–474, 1907, French translation of original work published in 1892. [Online]. Available: http://www.numdam.org/item?id=AFST_1907_2_9__203_0.

[8] A. M. Lyapunov, "The general problem of the stability of motion", *International Journal of Control*, vol. 55, no. 3, pp. 531–773, 1992, English translation of original work published in 1892.

[9] D. Liberzon, *Switching in Systems and Control.* Birkhäuser, 2003. DOI: 10.1007/978-1-4612-0017-8.

[10] R. E. Kalman, "On the general theory of control systems", in *Proc. 1st World Congress of the International Federation of Automatic Control*, 1960, pp. 481–493.

[11] V. I. Arnol'd, *Gewöhnliche Differentialgleichungen.* Springer-Verlag, 1979.

[12] J. L. Salle and S. Lefschetz, *Die Stabilitätstheorie von Ljapunov.* Mannheim: BI Hochschultaschenbücher, 1974.

[13] Y. K. Tan and A. Platzer, "Deductive Stability Proofs for Ordinary Differential Equations", in *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings, Part II*, J. F. Groote and K. G. Larsen, Eds., ser. Lecture Notes in Computer Science, vol. 12652, Springer, 2021, pp. 181–199. DOI: 10.1007/978-3-030-72013-1_10.

[14] Y. K. Tan and A. Platzer, "Switched Systems as Hybrid Programs", in *7th IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2021, Brussels, Belgium, July 7-9, 2021*, R. M. Jungers, N. Ozay, and A. Abate, Eds., ser. IFAC-PapersOnLine, vol. 54, Elsevier, 2021, pp. 247–252. DOI: 10.1016/j.ifacol.2021.08.506.

[15] U. Topcu, A. K. Packard, and P. J. Seiler, "Local stability analysis using simulations and sum-of-squares programming", *Autom.*, vol. 44, no. 10, pp. 2669–2675, 2008. DOI: 10.1016/j.automatica.2008.03.010.

[16] M. Anghel, F. Milano, and A. Papachristodoulou, “Algorithmic Construction of Lyapunov Functions for Power System Stability Analysis”, *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 60-I, no. 9, pp. 2533–2546, 2013. DOI: 10.1109/TCSI.2013.2246233.

[17] K. Forsman, “Construction of Lyapunov functions using Grobner bases”, in *Proceedings of the 30th IEEE Conference on Decision and Control*, 1991, pp. 798–799. DOI: 10.1109/CDC.1991.261424.

[18] J. Liu, N. Zhan, and H. Zhao, “Automatically Discovering Relaxed Lyapunov Functions for Polynomial Dynamical Systems”, *Math. Comput. Sci.*, vol. 6, no. 4, pp. 395–408, 2012. DOI: 10.1007/s11786-012-0133-6.

[19] S. Sankaranarayanan, H. B. Sipma, and Z. Manna, “Constructing invariants for hybrid systems”, in *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2004, pp. 539–554.

[20] M. S. Branicky, “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems”, *IEEE Trans. Autom. Control.*, vol. 43, no. 4, pp. 475–482, 1998. DOI: 10.1109/9.664150.

[21] Z. Sun and S. S. Ge, *Stability Theory of Switched Dynamical Systems*. Springer, Communications and Control Engineering, 2011. DOI: 10.1007/978-0-85729-256-8.

[22] A. Podelski and S. Wagner, “Model Checking of Hybrid Systems: From Reachability Towards Stability”, in *Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006, Proceedings*, J. P. Hespanha and A. Tiwari, Eds., ser. Lecture Notes in Computer Science, vol. 3927, Springer, 2006, pp. 507–521. DOI: 10.1007/11730637_38.