
HOW DEDUCTION SYSTEMS CAN HELP YOU TO VERIFY STABILITY PROPERTIES

A PREPRINT

Mario Gleirscher¹, Rehab Massoud³, Dieter Hutter^{1,2,*}, Christoph Lüth^{1,2,*} 

April 17, 2024

ABSTRACT

Mathematical proofs are a cornerstone of control theory, and it is important to get them right. Deduction systems can help with this by mechanically checking the proofs. However, the structure and level of detail at which a proof is represented in a deduction system differ significantly from a proof read and written by mathematicians and engineers, hampering understanding and adoption of these systems.

This paper aims at helping to bridge the gap between machine-checked proofs and proofs in engineering and mathematics by presenting a machine-checked proof for stability using Lyapunov's theorem in a human-readable way. The structure of the proof is analyzed in detail, and potential benefits of such a proof are discussed, such as generalizability, reusability and increased trust in correctness.

1 Introduction

Stability assurance is essential for the safe and reliable performance of controlled systems, such as autonomous robots or vehicles. A prominent safety-related issue is to guarantee that a controller combined with a machine constitutes a stable controlled system. The correct behavior of the controller has to be guaranteed rigorously in that context. In many cases, Lyapunov's work on stability theory provides the mathematical foundation for the required verification, delivering a framework for organizing corresponding proofs.

Often, such proofs are tedious, error-prone, and rely on hidden assumptions, thereby creating the desire for a higher automation in finding and checking such proofs. Deduction systems (particularly, interactive tools called proof assistants) have a long history and are successfully applied in various domains, for example, for verifying chip designs [13] and for solving configuration or optimization problems [9, 4, 11, 2, 5]. As opposed to mathematicians arguing on a semantic level, deduction systems operate on a pure syntactical level applying a fixed set of rules to rewrite a given problem until it is subsumed by the given facts. Such a syntactic or fully formalized proof can be mechanically checked. Depending on the expressiveness of the underlying logic, deduction systems are able to find complex proofs automatically or they might need help in particular situations, for instance to select an appropriate application of some proof rule from a number of applicable alternatives, or to stop the system from pursuing irrelevant paths during proof search.

In the past decade, deduction systems have significantly improved [7], now able to tackle problems in control theory and engineering, including stability assurance. For instance, the KeYmaera X tool [6] provides specific proof support for applying Lyapunov's theorems [15] and for verifying the stability of switched systems [14]. For another example, the Coq prover [1] was used to prove La Salle's theorem [3] and the stability of the inverted pendulum on a cart [12].

^{*1} University of Bremen, 28359 Bremen, Germany.

^{†2} Cyber-Physical Systems, DFKI, 28359 Bremen, Germany.

^{‡3} EPFL, Systems Integration Lab, 1015 Lausanne, Switzerland. Most of the work was done before joining EPFL, while affiliated with ².

^{§*} Supported by the VeryHuman project (grant number 01IW20004) funded the German Federal Ministry of Education and Research (BMBF).

Using such tools to formalize stability proofs, one not only obtains more detailed rigorous proofs, but also gains more insight and understanding of the problem and its dependencies on the various parameters. For example, in [12], authors found and fixed errors in a previous manual stability proof [10].

However, using deduction systems for non-trivial applications typically requires human interaction during the proof search. Even if the proof is well-understood at the mathematical level, it is still a non-trivial task to find its formal, syntactically derived counterpart. Vice versa, it is non-trivial to interpret a syntactical proof in its semantic meaning, which is a prerequisite for assisting the deduction systems in the next steps of a proof attempt.

This paper helps bridging this gap between semantic, human-oriented and syntactic, formal proofs. We discuss a formal proof of the stability of a controlled system using Lyapunov’s stability theorem, using the inverted pendulum as a running example. Our specific contributions are:

- We extract the structure (e.g. key steps) of a mechanization of a fundamental control-theoretic proof [15] and present it in terms familiar to control theorists.
- We connect a usual approach for problem family characterization (i.e. identifying healthy combinations of system dynamics and Lyapunov function templates) with a deductive proof using well-formedness constraints on the parameters of the dynamics and the Lyapunov function template.
- We replicate and enhance the proof in [15] using these constraints as an additional side-condition, allowing both, the deductive system and its user, to navigate the proof more easily and intuitively.

The paper is organized as follows: Section 2 presents the formal background and the running example, before we apply Lyapunov’s direct method in Section 3 to characterise stable pendula. Section 4 explains the deductive proof. We then highlight in Section 5 the benefits of our approach, before we conclude the paper in Section 6.

2 Formal Preliminaries

After defining Lyapunov’s stability theorem and giving an overview of differential dynamic logic—the deduction system used in this work for constructing stability proofs—we introduce the inverted pendulum as a running example. Scalars $x \in \mathbb{R}$ are in italic and vectors $\mathbf{x} \in \mathbb{R}^n$ in bold.

2.1 Lyapunov Stability Theory

Given system dynamics $\dot{\mathbf{x}}$ and, without loss of generality, an isolated equilibrium point $\mathbf{x}_e = \mathbf{0} \in \mathbb{R}^n$, asymptotic stability [8] of $\mathbf{x}_e$ amounts to establishing

$$\begin{aligned} \forall \epsilon > 0 \exists \delta: \|\mathbf{x}_0\| < \delta \Rightarrow \forall t \geq 0: \|\xi(t; \mathbf{x}_0)\| < \epsilon \quad \text{and} \\ \exists \delta > 0: \|\mathbf{x}_0\| < \delta \Rightarrow \lim_{t \rightarrow \infty} \|\xi(t; \mathbf{x}_0)\| = 0 \end{aligned}$$

of trajectories ξ emanating from $\mathbf{x}_0$ in a δ environment. *Lyapunov’s direct method* simplifies this approach to proving for $\dot{\mathbf{x}}$ and $\mathbf{x} \neq \mathbf{x}_e$ the existence of a function V with

$$V(\mathbf{x}) > 0 \quad (\text{positive definiteness of } V) \quad \text{and} \quad (1)$$

$$\dot{V}(\mathbf{x}) < 0 \quad (\text{negative definiteness of } \dot{V}), \quad (2)$$

where $\dot{V} \equiv \nabla V \cdot \dot{\mathbf{x}}$ is the total derivative of V along $\dot{\mathbf{x}}$. One challenge with the direct method is finding such a V .

2.2 Sequent Calculus and Differential Dynamic Logic

Sequent calculus is a deduction system going back to Gentzen and Hilbert. Basically, a sequent is given as $A_1, \dots, A_n \vdash B_1, \dots, B_m$ where A_i are the assumptions and B_j the conclusions, meaning that if all A_i hold, one of B_j will hold (where A_i, B_j are formulae). Formal proofs are represented as proofs trees, labeled at the nodes with sequents, and constructed using a handful of inference rules such as the *Cut* rule

$$\frac{\Gamma \vdash \Pi, A \quad A, \Sigma \vdash \Delta}{\Gamma, \Sigma \vdash \Pi, \Delta} \quad \text{Cut} \quad (3)$$

which allows us to drop (or cut) the formula A from the sequent. Other rules allow the manipulation of sequents (adding axioms, dropping assumptions) or formulae. Some rules can be applied schematically (such as rules introducing a connective), but in particular using the cut rules requires a semantic understanding of the proof.

We use *differential dynamic logic* (dL) to express properties and (hybrid) programs. dL allows writing down hybrid programs and properties in one language. Programs combine discrete and continuous state transitions; in this paper, we are only interested in the latter. A program is given by the usual constructs for assignments, choice and iteration, and differential equations describing the evolution of the variables. Properties are given by first-order formulae of real arithmetic, and the modal operators $\langle P \rangle \alpha$ and $[P] \alpha$, where P is a hybrid program and α a formula; $[P] \alpha$ that means that α always holds when and if P terminates, and $\langle P \rangle \alpha$ means that P will terminate in at least one state where α is true.

There are rules allowing us to manipulate terms according to the usual rules of first-order logic (quantifier elimination/introduction etc.), and rules for the modal operators, such as the *differential Cut* which illustrates the interplay between structural logic rules and differential equations:

$$\frac{\Gamma \vdash [\dot{\mathbf{x}} = f(\mathbf{x}) \& Q] C, \Delta \quad \Gamma \vdash [\dot{\mathbf{x}} = f(\mathbf{x}) \& Q \wedge C] P, \Delta}{\Gamma \vdash [\dot{\mathbf{x}} = f(\mathbf{x}) \& Q] P, \Delta} \text{dC}$$

Here, $\dot{\mathbf{x}} = f(\mathbf{x}) \& Q$ describes a program where $\mathbf{x}$ evolves as specified as long as Q holds; it terminates once Q does not hold. If we can show that C holds whenever the program terminates, and if we can further show P with C in the assumptions, we may drop C entirely. This rule is similar to the traditional cut rule (3) in that it allows to drop assumptions, but not derivable since it also deals with the evolution of $\mathbf{x}$ over time.

Auxiliary Notation Given some origin $\mathbf{x}_e$, a ball B_p is a closed set $\{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}_e\| \leq p\}$. With $\bar{B}_p$, B_p° , and ∂B_p , we denote B_p 's complement, interior, and boundary, respectively. For $p, q \in \mathbb{R}_{>0}$, ${}_p \langle P \rangle_q$ and ${}_p [P]_q$ denote that if program P is initialized in B_p (i.e. $\mathbf{x} \in B_p$) then P 's state can reach, respectively, will remain in B_q . If p is omitted, then p is assumed to be provided by the context. We abuse $\bar{p}$, p° , and ∂p to denote B_p 's complement, interior, and boundary. We use $=$ for definitions and equalities and $\equiv$ for abbreviations.

2.3 Linearly Controlled, Damped, Plane Inverted Pendula

The inverted pendulum captures a range of applications (e.g. in robotics) and is, thus, instructive for stability analysis. Consider a pendulum of mass m with a rigid rod of length l (the distance from its center of rotation and its center of mass). With the angle θ between rod and vertical axis, the upper equilibrium point is $\theta_e = 0$. The pendulum is subjected to friction (proportional to its angular velocity ω and a constant f) while being controlled for bringing it to and keeping it at $\mathbf{x}_e = [\theta_e, \omega_e]^\top = \mathbf{0}$. The dynamics are given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega \\ a\theta + b\omega + c \sin \theta \end{bmatrix} \quad (4)$$

where $a = \frac{k_1}{ml}$, $b = \frac{fl+k_2}{ml}$, and $c = -\frac{g}{l}$ with gravity g .

We assume there to be a linear controller of the form $\mathbf{u} = \mathbf{k}^\top \mathbf{x}$ with tuning parameters $\mathbf{k} = [k_1 k_2]^\top$ and corresponding to a tangential force applied to the center of mass. If $\mathbf{u}$ is generated by a momentum around the center of rotation, we have $\mathbf{k} = \frac{\mathbf{k}_m}{l}$ where $\mathbf{k}_m$ contains the parameters of a torque controller $\mathbf{u}_m$. The discussion below is, thus, largely independent of whether $\mathbf{k}$ comes from linear proportional-derivative or linear-quadratic regulator design and identification, how $\mathbf{k}$ is interpreted, or whether $\mathbf{k}$ is chosen without considering performance criteria more specific than asymptotic stability.

3 Characterizing Stable Inverted Pendula

We illustrate, by example of the pendulum, (i) stability assurance of non-linear systems using Lyapunov's direct method and, inspired by [8], (ii) the characterization of stable problem families by deriving constraints on the parameters of a given dynamics $\dot{\mathbf{x}}$ and an appropriate Lyapunov function template V .

It is well-known [8] that quadratic forms $\mathbf{x}^\top \mathbf{P} \mathbf{x}$ can be used to modify a system's potential and kinetic energy function to meet necessary criteria for definiteness. Based on typical energy modeling of inverted pendula, we then obtain

$$V = \frac{ml^2}{2} (p_{11}\theta^2 + 2p_{12}\theta\omega + p_{22}\omega^2 - 2c(1 - \cos \theta)) \quad (5)$$

where $\mathbf{P} = [p_{ij}]_{i,j \in \{1,2\}}$ is a symmetric matrix to be defined in a way to establish (1) and (2). This V is an example of a Lyapunov function template that characterizes the stable family of pendula of the type described by (4).

From matrix analysis, we know that (1) can be achieved by $p_{11}p_{22} - p_{12}^2 > 0$ and $p_{11} > 0$. For (2), (5) and (4) yield

$$\dot{V} = \frac{ml^2}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + ml((g - gp_{22})\omega - gp_{12}\theta) \sin \theta \quad (6)$$

with symmetric

$$\mathbf{Q} = \begin{bmatrix} 2ap_{12} & p_{11} + bp_{12} + ap_{22} \\ p_{11} + bp_{12} + ap_{22} & 2(p_{12} + bp_{22}) \end{bmatrix}.$$

From $mlg(1 - \cos \theta)$ being positive definite, $f, g, l, m > 0$, and $a, b, c < 0$, the coefficients in the expansion of (6) suggest a simplification by $p_{22} = 1$. These observations allow us to under-approximate the stable family of inverted pendula by the following constraints on $\dot{\mathbf{x}}$ (via a, b) and $\mathbf{P}$:

$$p_{11} > p_{12}^2 \wedge p_{12} > 0 \wedge -\frac{p_{11}}{p_{12}} < b < -p_{12} \quad (7)$$

$$\text{and } a = -p_{11} - bp_{12}. \quad (8)$$

A proof using Lyapunov's direct method is possible if these constraints are satisfiable.⁵ $\mathbf{P}$ can then be chosen such that sign-indefinite terms in (6) are canceled. Depending on a, b , (8) determines, with p_{11}, p_{12} , the family of stable pendula.

Resolving (8) provides $p_{11} = -a - bp_{12}$. With p_{12} in (7), we can express the pendula family based on (5) and (6). For V to remain positive definite, we have to test $0 < p_{12} < \sqrt{p_{11}}$ from above, the latter of which is $p_{12} < \sqrt{-a - bp_{12}}$.

Model Validation Our plausibility checks for the development of (4) and (5) are informed by simulation and an understanding of the dynamics (a vector field) and the shape of V and $\dot{V}$ as developed in Fig. 1 (and Fig. 5).

4 Deductive Proofs of Asymptotic Stability

We now revisit the mechanization of a proof of Lyapunov's stability theorem [15] in KeYmaera X. This mechanization can be used for stability assurance of non-trigonometric systems. We add a proof parameter wf for well-formedness, which can make use of (7) and (8) and thereby reduce proof complexity while maintaining sound reasoning.

Lyapunov's stability theorem enables the use of the direct method (Sec. 2.1). Its proof [8, Theorem 4.1] uses an ϵ - δ -construction. It has been mechanized [15] in dL (Sec. 2.2) and applied to a non-trigonometric pendulum in Sec. 2.3.

Running Example We adopt the method in [15] for proving a similar variant where sine and cosine in (4) and (5) are replaced with their first- and second-order Taylor expansions $\sin \theta \approx \theta$ and $\cos \theta \approx 1 - \frac{1}{2}\theta^2$. The reason for such approximations are well-known limitations of quantifier elimination (QE) of trigonometric and general polynomials. As a result, we obtain a linear-quadratic problem comprising

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega \\ d\theta + b\omega \end{bmatrix} \quad \text{with } d = a + c, \quad (9)$$

represented as the purely continuous hybrid program

$$\text{IP} \equiv \{\dot{\theta} = \omega, \dot{\omega} = d\theta + b\omega\}$$

together with a non-trigonometric Lyapunov function

$$V = \frac{ml^2}{2} (-(d + bp_{12})\theta^2 + 2p_{12}\theta\omega + \omega^2) \quad (10)$$

such that the problem can be handled rather straightforwardly by the chosen tooling, KeYmaera X and the Wolfram Engine.

⁵Unsatisfiable constraints deny conclusions about the stability of $\dot{\mathbf{x}}$. Because of the under-approximation, there might still be some V that could work. In particularly, definiteness analysis of $-\mathbf{Q}$ not further pursued here may allow constraint relaxation and lead to a more complete method.

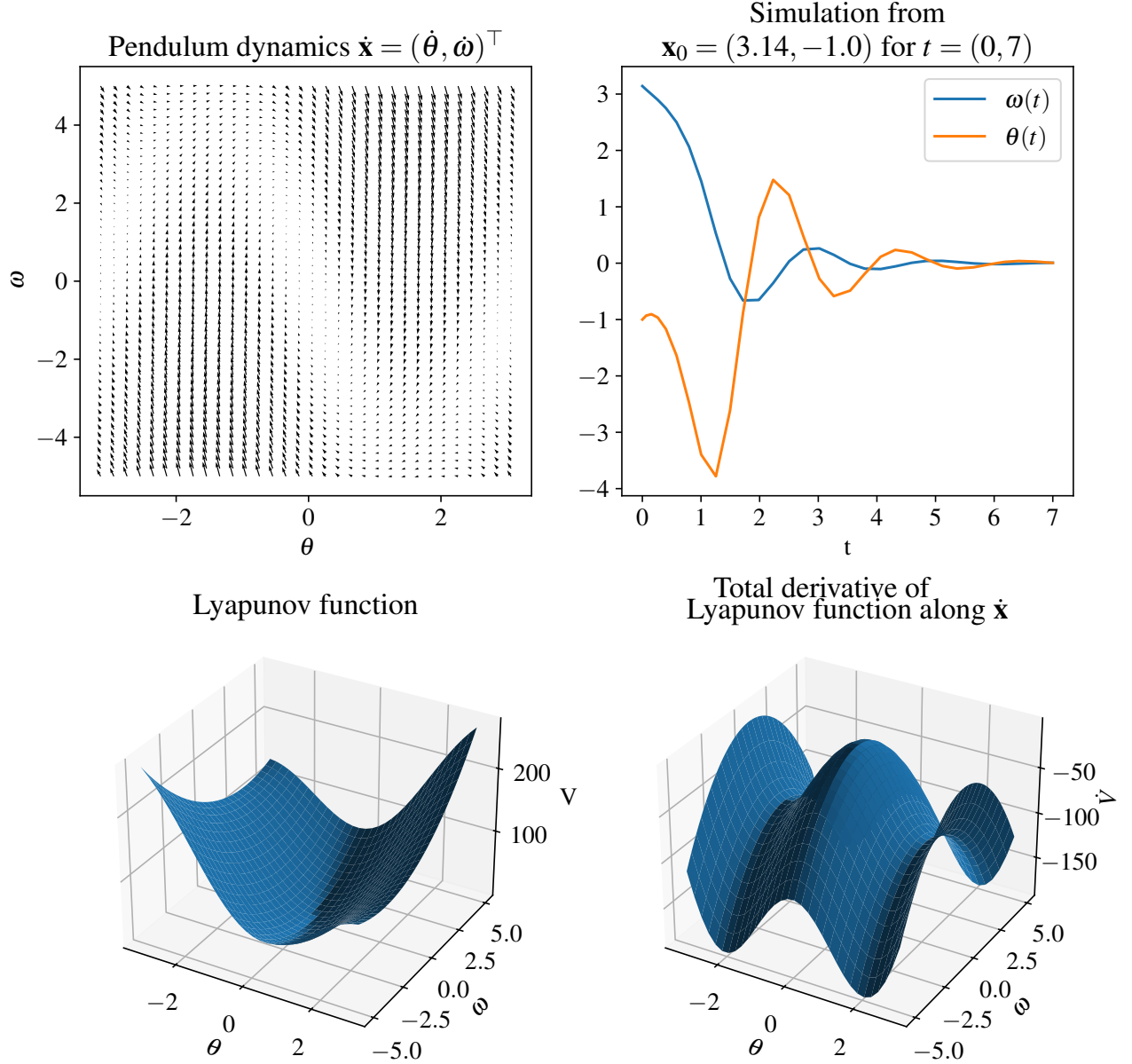


Figure 1: The trigonometric pendulum variant according to (4): the vector field of $\dot{\mathbf{x}}$ (upper left), a simulation (upper right), V (lower left), and its total derivative $\dot{V}$ (lower right)

4.1 Key Elements of the Proof

We parameterize the $d\mathcal{L}$ sequent-style proof in [15] by a program P , a Lyapunov function template V , and a well-formedness condition wf . In our example, we use $P \equiv \text{IP}$,

$$wf \equiv (7) \wedge \bigwedge_{i \in \{g, l, m\}} i > 0 \wedge \bigwedge_{j \in \{a, b, c, d\}} j < 0$$

and $V \equiv (10)$, but will continue with the parametric proof.

The proof tree is split into three parts as illustrated in the Figures 3, 2, and 4 reflecting its structure. The tree is to be read from the root sequent at the bottom to the top. If a node in the tree has two branches, we call the *left*-hand branch the *show*-branch and the *right*-hand branch the *use*-branch.

$$\begin{array}{c}
\text{dW, FOL}_{\mathbb{R}, \text{ODE}} \quad \frac{*}{\dots \vdash [\text{P}]_{\bar{k}, \partial k} \quad \dots \vdash [\text{P} \& \Omega_k]_{\epsilon}} \quad \text{ODE} \\
\text{QE} \quad \frac{*}{wf, \exists \Omega_k \vdash \exists B_{\delta} \subset \Omega_k} \quad \frac{wf, \exists \Omega_k, \exists B_{\delta} \subset \Omega_k \vdash s}{wf, \epsilon > 0, \exists \Omega_k \vdash s} \quad \text{dC1} \\
\text{QE} \quad \frac{*}{wf, \epsilon > 0 \vdash \exists \Omega_k, s} \quad \text{M}\exists\text{R} \\
\text{Cut2} \quad \frac{wf, \epsilon > 0 \vdash \exists \Omega_k, s}{\wedge\text{R}, \rightarrow\text{R} \quad wf \vdash s} \quad \text{Figs. 3,4} \quad \text{Cut3} \\
\frac{\wedge\text{R}, \rightarrow\text{R} \quad wf \vdash s}{wf \vdash s \wedge (s \rightarrow a), s \wedge a} \quad \frac{*}{wf, s, s \rightarrow a \vdash s \wedge a} \quad \text{PL} \\
\text{Cut1} \quad \frac{wf \vdash s \wedge (s \rightarrow a), s \wedge a}{wf \vdash s \wedge a}
\end{array}$$

Figure 2: Part 1: Decomposing asymptotic stability into stability (s) and attractivity (a) in d $\mathcal{L}$ -sequent style extracted from [15]

$$\begin{array}{c}
\text{PL,QE} \frac{\text{wf}, \delta > 0, \mathbf{x} \in B_\delta, [\mathsf{P}]_1 \vdash b = \min_1 V, \langle \mathsf{P} \rangle_\epsilon}{\text{Cutd}}^* \\
\\
\text{PL,QE} \frac{\text{wf}, \delta > 0, \mathbf{x} \in B_\delta, [\mathsf{P}]_1 \vdash \langle \mathsf{P} \rangle_\epsilon}{\text{Simp}}^* \\
\quad \frac{\text{wf}, \exists \delta : {}_\delta[\mathsf{P}]_1 \vdash \exists \delta \forall \epsilon : {}_\delta \langle \mathsf{P} \rangle_\epsilon}{\text{L}\&\mathsf{R}, \forall \mathsf{R}} \\
\quad \frac{\text{wf}, \forall \epsilon \exists \delta : {}_\delta[\mathsf{P}]_\epsilon \vdash \exists \delta \forall \epsilon : {}_\delta \langle \mathsf{P} \rangle_\epsilon}{\text{L}(1)} \\
\quad \frac{\text{wf}, s \vdash r, a}{\text{Fig. 2}} \text{Cut3 (show } r)
\end{array}$$

Figure 3: Part 2: Proving reachability (r) of the equilibrium point $\mathbf{x}_e$ and any of its environments

In part 1 (Fig. 2), the root sequent $wf \vdash s \wedge a$ expresses asymptotic stability by concluding stability (s) and attractivity (a), formally,

$$\begin{aligned} s &\equiv \forall \epsilon \exists \delta : \delta [P]_{\epsilon} \\ &\equiv \forall \epsilon > 0 \exists \delta > 0 \forall \mathbf{x} \in B_{\delta} : [P]\mathbf{x} \in B_{\epsilon} \quad \text{and} \\ a &\equiv \exists \delta \forall \epsilon : \delta \langle P \rangle [P]_{\epsilon} \\ &\equiv \exists \delta > 0 \forall \mathbf{x} \in B_{\delta}, \epsilon > 0 : \langle P \rangle [P]\mathbf{x} \in B_{\epsilon}, \end{aligned}$$

from a well-formedness condition (*wf*) that system parameters (e.g. a, b, c) must satisfy.

Condition s specifies that P remains within any ϵ environment if P is initialized in the corresponding δ environment. Condition a holds if P will reach and remain in an arbitrarily small B_ϵ when starting from some B_δ .

Part 1 (Sec. 4.2) splits the proof into a stability proof and a proof that attractivity is implied by stability if the origin $\mathbf{x}_e$ is stable [15, Remark 3]. One can, however, see that s and a do not generally imply each other. The stability proof follows the idea that within an arbitrary B_ϵ , one can find a largest set $\Omega_k \subseteq \mathbb{R}^n$ of states with $V \leq k$ (i.e. a k -levelset) enclosed in B_ϵ and some B_δ enclosed in Ω_k . (1) and (2) are used as assumptions on the way to prove that any trajectory leaving B_ϵ must have started outside B_δ . Moreover, Part 2 (Sec. 4.3) handles the proving of reachability of $\mathbf{x}_e$ from $\mathbf{x}_0$ by showing strictly V -monotonic progress of $\mathbf{P}$ towards $\mathbf{x}_e$. Finally, Part 3 (Sec. 4.4) deals with proving stability (in formal verification also called *safety*) of $\mathbf{x}_e$ based on reachability of $\mathbf{x}_e$.

The proof tree explained below highlights key steps (i.e. applying inference rules to intermediate verification conditions in the antecedents and succedents), particularly, cuts (3, i.e. deductive shortcuts using auxiliary conditions and requiring user interaction). For brevity, we omit steps non-essential for understanding the structure (e.g. weaken, unfolding) and occasionally hide conditions in the antecedents and succedents if they are irrelevant for the current step. The most relevant rules are explained in Sec. 2.2 and App. A.1.

4.2 Part 1: Decomposing Asymptotic Stability

The proof of the main theorem (the root sequent in Fig. 2) comprises several key steps shown separately below.

Cut1: Observed by [15, Remark 3, Cor. 5], the proof of asymptotic stability gets simpler if $\mathbf{x}_e$ is stable. We, hence, introduce the premise $s \wedge (s \rightarrow a)$ of modus ponens as an auxiliary condition. This cut allows us to separately prove s , $s \rightarrow a$, and via modus ponens, $s \wedge a$.

Cut2: Given $\epsilon > 0$, we search for the largest k -levelset

$$\Omega_k = \{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq k\}$$

enclosed in ball B_ϵ . Because of $V(\mathbf{x}(0)) \geq 0$ and $\dot{V}(\mathbf{x}(t)) \leq 0$ (derived from assumptions 1 and 2), any trajectory ξ starting in Ω_k remains in Ω_k . We use the cut condition

$$\exists \Omega_k \equiv \exists k > 0 \forall \mathbf{x}: \|\mathbf{x}\| = \epsilon \rightarrow V(\mathbf{x}) \geq k$$

defining this k . For the non-trigonometric pendulum, $\exists \Omega_k$ can be checked automatically from $wf \wedge \epsilon > 0$ by QE.

M $\exists$ R: To prove s from $\exists \Omega_k$, we now look for a ball B_δ with $0 < \delta < \epsilon$ inside Ω_k such that $\forall \mathbf{x} \in B_\delta: V(\mathbf{x}) < k$. To facilitate this, we cut the condition $\exists B_k \subset \Omega_k$ into the proof using the specific cut rule M $\exists$ R.

dC1: Next, by a differential cut of P, the conditions introduced by Cut2 and M $\exists$ R can now be used to perform two sub-proofs: First, that whenever P is at boundary ∂B_ϵ then it is either outside Ω_k or at $\partial \Omega_k$, formally,

$$[P](\mathbf{x} \in \partial B_\epsilon \rightarrow \mathbf{x} \in \bar{\Omega}_k \cup \partial \Omega_k)$$

(for tree brevity, $[P]_{\bar{k}, \partial k}$) and, second, that P, restricted to Ω_k , remains inside B_ϵ (indicated with $[P \& \Omega_k]_\epsilon$).

ODE: We resolve the unrestricted box modality $[P]_{\bar{k}}$ using differential weaken (dW) and first-order sequent inference (FOL $_{\mathbb{R}}$) and $[P]_{\partial k}$ by ordinary differential equation (ODE) solving. The restricted box modality $[P \& \Omega_k]_\epsilon$ on the right can be tackled directly with automated ODE solving.

The “*” leaf in each branch of the tree signifies that an axiom (i.e. a statement whose truth is justified outside dL) has been reached. As shown before, relating the top-most (i.e. most detailed) proof steps with axioms is made easier by automated inference, for example, QE and ODE solving.

4.3 Part 2: Proving Reachability of the Equilibrium Point

A key task in this part (Fig. 3) is to derive r from s by establishing a differential variant (**dV**), representing the strictly monotonic decrease of V along the trajectories of P.

$\forall, \exists$: We eliminate some quantifiers first. With **$\forall L(1)$** , we fix the assumption $\epsilon = 1$, that is, for some δ , P can remain within the unit circle B_1 . Then, **$\exists L$** eliminates $\exists$ in the antecedent. After $\exists$ -elimination in the succedent with **$\exists R$** , δ is provided by the antecedent.

Cut4: After simplifications (Simp), we cut in the condition

$$(b = \min_1 V) \equiv \exists b \forall \mathbf{x} \in B_1: V(\mathbf{x}) \geq b,$$

which characterises V ’s minimum b inside B_1 . By QE, we verify whether such a b , consistent with wf , exists.

dC_c: In the use-branch of Cut4, we introduce an auxiliary condition $[P]_{\bar{\epsilon}}$ (i.e. P remains outside B_ϵ), leading to the tautology $\langle P \rangle_\epsilon \vee [P]_{\bar{\epsilon}}$ in the show-branch of dC_c. In its use-branch, $[P]_1 \wedge [P]_{\bar{\epsilon}} \wedge b = \min_1 V$ is at our disposal: P never leaves B_1 , never reaches B_ϵ , and B_1 contains the b -levelset.

$\langle \& \rangle 2$: We show (i) using auxiliary condition $V(\mathbf{x}) \geq b^6$ on the left for any B_ϵ , that trajectories inside B_1 and outside B_ϵ remain outside the b -levelset, and (ii) with an auxiliary condition $V(\mathbf{x}) < b$ on the right that there is a trajectory reaching the b -levelset. We have to show that (i) and (ii) can only be true simultaneously if $\epsilon = 0$, meaning that $\mathbf{x}_\epsilon$ is attractive for P. ($[P]_{\bar{\epsilon}}$ can only be true for $\epsilon = 0$, hence, B_ϵ for an arbitrarily small $\epsilon > 0$ must be reachable.) While (i) can be shown from the antecedent by dW and propositional reasoning (PL), (ii) needs a reachability proof.

dV: The reachability proof can be tackled by proving a differential invariant that amounts to checking that $\dot{V} < 0$. The dV rule rephrases this check into

$$\exists p > 0 \forall \mathbf{x} \in \bar{B}_\epsilon \cap B_1: V(\mathbf{x}) \geq b \rightarrow -\dot{V}(\mathbf{x}) \geq p$$

that is, checking for monotonic progress $-\dot{V} \geq p > 0$ of V along P’s trajectories in $B_1 \cap \bar{B}_\epsilon$ and apart from minimum b .

Cut5: Inside B_1 , $b = \min_1 V$ implies $V \geq b$. So, we perform a cut with

$$(\exists p: -\dot{V} \geq p) \equiv \exists p > 0 \forall \mathbf{x} \in \bar{B}_\epsilon \cap B_1: -\dot{V}(\mathbf{x}) \geq p$$

to (i) check the differential variant via QE in the context of wf in the antecedent and (ii) use it for proving $\langle P \rangle_\epsilon$.

⁶Here, we do not need to make use of the assumption $V(\mathbf{x}_\epsilon) = 0$.

$$\begin{array}{c}
\text{PL} \frac{\text{dW,PL} \frac{\dots, \delta[P]_\epsilon \vdash [P \& \neg [P]_\epsilon]_\delta [P]_\epsilon \quad \dots \vdash [P \& \neg [P]_\epsilon \wedge \delta[P]_\epsilon]_\delta}{\langle P \rangle_\delta \vdash [P \& \neg [P]_\epsilon]_\delta} \quad \text{dC2}}{\frac{wf, \langle P \rangle_\delta \vdash \langle P \rangle_\delta}{wf, \langle P \rangle_\delta \vdash \langle P \rangle [P]_\epsilon} \quad \text{dC2}} \\
\frac{wf, \langle P \rangle_\delta \vdash \langle P \rangle [P]_\epsilon \quad \forall L(\delta)}{wf, \exists \delta \forall \epsilon: \delta \langle P \rangle_\epsilon \vdash \langle P \rangle [P]_\epsilon} \quad \text{dC2} \\
\frac{wf, r \vdash a}{\text{Fig. 2}} \quad \text{Cut3 (use } r)
\end{array}$$

Figure 4: Part 3: Proving stability of the equilibrium point $\mathbf{x}_e$ based on its reachability

4.4 Part 3: Proving Stability of the Equilibrium Point

In Fig. 4, we continue with the proof of $s \rightarrow a$, the observed implication between stability and attractivity under the condition of a stable $\mathbf{x}_e$. We can, only then, establish attractivity as a special case of stability.

Cut3: Attractivity requires that from B_δ (to be proven to exist in the stability proof), an arbitrarily small B_ϵ can be reached, formally,

$$r \equiv \exists \delta \forall \epsilon: \delta \langle P \rangle_\epsilon \equiv \exists \delta > 0 \forall \mathbf{x} \in B_\delta, \epsilon > 0: \langle P \rangle \mathbf{x} \in B_\epsilon.$$

This auxiliary condition is introduced by Cut3. Unlike where $M\exists R$ is applied, B_δ can now also include B_ϵ .

E(r, a): r and a are merely expanded here. Note that asymptotic behavior implies that once P reaches B_ϵ from B_δ , it stays there. Hence, note the $[P]_\epsilon$ in the succedent.

$\forall L(\delta)$: After eliminating $\exists \delta$, we fix $\epsilon = \delta$ in the antecedent to connect the reachability condition r with the $\langle \cdot \rangle$ -modality of the attractivity condition a , eliminating $\forall \epsilon$ on the left and reducing our focus to the proof of $\langle P \rangle_\delta$, that is, that we can reach an arbitrarily small $B_\delta \subseteq B_\epsilon$ for any ϵ in the succedent.

$\langle \& \rangle 1$: We decompose $\langle P \rangle [P]_\epsilon$ into a δ -reaching $\langle \cdot \rangle$ -modality (the show-branch establishes the aforementioned connection) and an ϵ -safe $[\cdot]$ -modality. The use-branch of $\langle \& \rangle 1$ produces the contra-positive of $\langle P \rangle [P]_\epsilon$, namely, $[P \& \neg [P]_\epsilon]_\delta$. That is, evolution in states not ϵ -invariant must occur outside B_δ , even outside Ω_k . (Because the antecedent provides reachability of $B_\delta \subset \Omega_k$, P remains in B_ϵ .)

dC2: To the use-branch of $\langle \& \rangle 1$, we apply a differential cut with the auxiliary condition $\delta [P]_\epsilon$, that is, ϵ -safety for some positive δ , to be established as an invariant in the show-branch. Both dC2-branches can be closed by weakening (dW). On the left, we show that from $\delta [P]_\epsilon$, formula $[P \& \neg [P]_\epsilon]$ is vacuously true (discharged by PL). On the right, we then use the fact that there can be no state in the intersection of $\neg [P]_\epsilon \wedge \delta [P]_\epsilon$, concluding that P must be outside B_δ in order to leave an arbitrarily small B_ϵ .

5 Discussion

In the previous section, we have given an in-depth review of a deductive stability proof using the Lyapunov method, which has been formalized in the Keymaera X prover. It has hopefully become clear that the level of detail at which such proofs are conducted goes beyond the level usually employed by mathematicians or engineers.

An important benefit of working at this level, and of the mechanized proof check enabled by this effort, is that we can re-run the proof after changes, ensuring these changes do not invalidate the proof, and that we can be certain that all side conditions and assumptions are made explicit. For example, in the presented proof, the family of Lyapunov functions comes as a handy test for the associated controller design: we can change the controller, possibly plug in a changed Lyapunov function, and re-run the proof. We can be sure if our side conditions (e.g. constraints on the parameters) are too weak, the mechanized proof will likely fail.

In Sec. 3, we derived constraints on the problem parameters $(a, b, \dots, \mathbf{P})$ to enrich the well-formedness condition wf . The strength of wf can thus be better adjusted and, thus, aid both, the proof assistant and its user, in understanding and closing the relevant parts of a stability proof instance.

Sec. 4 further indicates opportunities for new features on the prover side to make deductive systems and mechanized proof more readily available to the practising control engineer. For example, a proof tree visualization (similar to that in Keymaera X) is essential in understanding the proof structure. In Keymaera X, proofs are conducted and stored in the Bellerophone tactical language. With more complex proofs requiring a range of cuts, it could, for example, be beneficial to navigate several tree branches side-by-side in addition to navigating at the Bellerophone level.

The presentation of proofs at an appropriate level, as illustrated in Sec. 4 for the stability proof, is crucial in following the successive application of proof rules, including their role in completing the proof. This allows proof authors and

users (e.g. researchers, practitioners, certification auditors) to validate whether a proven theorem faithfully represents the application problem, for example, a stability assurance problem. Particularly important for these roles will be the validity of the assumptions included in the well-formedness condition as well as the approach to identifying auxiliary conditions for the various forms of cuts as applied in the example proof in Sec. 4. An understanding of both these aspects will support the ability of engineers to re-run, modify, and transfer existing proofs into different settings.

6 Conclusion

Deduction systems in general and proof assistants in particular facilitate obtaining stronger formal, detailed and mechanized mathematical proofs; that are more qualified to serve as guarantees of safety and other critical properties. In this paper, we address one of the main barriers hindering wider adoption of deduction systems in control engineering research and practice; namely the understandability of the proofs and the complexity of the derivation steps. We explained how proof assistants can map manual proofs into mechanized ones by carrying on two stability proofs: a control-theoretic mathematical proof, and a one produced using the automated deduction tool KeYmaera X. The steps to guide the deduction system to complete the same proof mechanically was explained, and we highlighted the insights gained from doing the proof using the proof assistant KeYmaera X. The steps presented serve as a guide for how to carry on such a proof using a deduction system. This guidance can enable wider use and reuse of the proofs and deduction systems by control engineers, in ways that contribute to more safe and formally verified controllers.

References

- [1] Yves Bertot and Pierre Castéran. *Interactive theorem proving and program development: Coq’Art: the calculus of inductive constructions*. Springer, 2013.
- [2] Brandon Bohrer, Yong Kiam Tan, Stefan Mitsch, Andrew Sogokon, and André Platzer. A formal safety net for waypoint following in ground robots. *IEEE Robot. Autom. Lett.*, 4(3):2910–2917, 2019.
- [3] Cyril Cohen and Damien Rouhling. A formal proof in Coq of La Salle’s invariance principle. In *ITP*, volume 10499 of *LNCS*, pages 148–163. Springer, 2017.
- [4] Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Trans. Robot.*, 2023.
- [5] Ankush Desai, Tommaso Dreossi, and Sanjit A Seshia. Combining model checking and runtime verification for safe robotics. In *RV*, volume 10548 of *LNCS*, pages 172–189. Springer, 2017.
- [6] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völpl, and André Platzer. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In Amy P. Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 527–538. Springer, 2015.
- [7] Mario Gleirscher, Jaco van de Pol, and James Woodcock. A manifesto for applicable formal methods. *Softw. Syst. Model.*, (22):1737–1749, 2023.
- [8] Hassan Khalil. *Nonlinear Control*. Pearson, Boston, 2015.
- [9] Sarah M. Loos, David W. Renshaw, and André Platzer. Formal verification of distributed aircraft controllers. In Calin Belta and Franjo Ivancic, editors, *HSCC*, pages 125–130. ACM, 2013.
- [10] Rogelio Lozano, Isabelle Fantoni, and Dan J. Block. Stabilization of the inverted pendulum around its homoclinic orbit. *Systems & Control Letters*, 40(3):197–204, 2000.
- [11] Stefan Mitsch and André Platzer. ModelPlex: Verified runtime validation of verified cyber-physical system models. In Borzoo Bonakdarpour and Scott A. Smolka, editors, *RV*, volume 8734 of *LNCS*, pages 199–214. Springer, 2014.
- [12] Damien Rouhling. A formal proof in Coq of a control function for the inverted pendulum. In *CPP, 7th ACM SIGPLAN Int Conf*, pages 28–41. ACM, 2018.
- [13] Natarajan Shankar. Automated deduction for verification. *ACM Comput. Surv.*, 41(4), oct 2009.
- [14] Yong Kiam Tan, Stefan Mitsch, and André Platzer. Verifying switched system stability with logic. In Ezio Bartocci and Sylvie Putot, editors, *HSCC*, pages 2:1–2:22. ACM, 2022.
- [15] Yong Kiam Tan and André Platzer. Deductive stability proofs for ordinary differential equations. In Jan Friso Groote and Kim G. Larsen, editors, *TACAS*, volume 12652 of *LNCS*. Springer, 2021.

A Appendix

A.1 Further Proof Rules

Let $C \equiv \dot{\mathbf{x}} = f(\mathbf{x})$ be a purely continuous hybrid program in the following. For reference, we provide an overview of further $d\mathcal{L}$ sequent inference rules used in this work:

dC_c The compatible ODE cut rule, defined as

$$\frac{\Gamma, [C \& Q]R \vdash C(\langle C \& Q \rangle P), \Delta \quad \Gamma \vdash [C \& Q]R, \Delta}{\Gamma \vdash C(\langle C \& Q \rangle P), \Delta} \mathbf{dC}_c$$

can prepare for proving a differential variant (**dV**). The branches in **dC_c** 's premise enable showing and using the compatible box modality assumption $[C \& Q]R$.

$\langle \& \rangle$ The domain diamond rule, defined as

$$\frac{\Gamma \vdash \langle C \& Q \rangle R, \Delta \quad \Gamma \vdash [C \& Q \wedge \neg P] \neg R, \Delta}{\Gamma \vdash \langle C \& Q \rangle P, \Delta} \langle \& \rangle$$

can prepare for proving a differential variant (**dV**) based on postcondition R implying the original condition P . The contrapositive $\neg P \rightarrow \neg R$ thereof is encoded in **$\langle \& \rangle$** 's right-hand branch.

dW The differential weaken rule, defined as

$$\frac{\Gamma_{const}, Q \vdash P, \Delta_{const}}{\Gamma \vdash [C \& Q]P, \Delta} \mathbf{dW}$$

allows one to use the circumstance that if Q holds and C is restricted to Q , P holds independent of C 's execution.

A.2 Supplemental Materials

For comparison, Fig. 5 provides the validation data set for the non-trigonometric pendulum controller.

Fig. 6 shows the user interface of the KeYmaera X proof assistant for navigating through the branches of a finalized $d\mathcal{L}$ sequent proof tree.

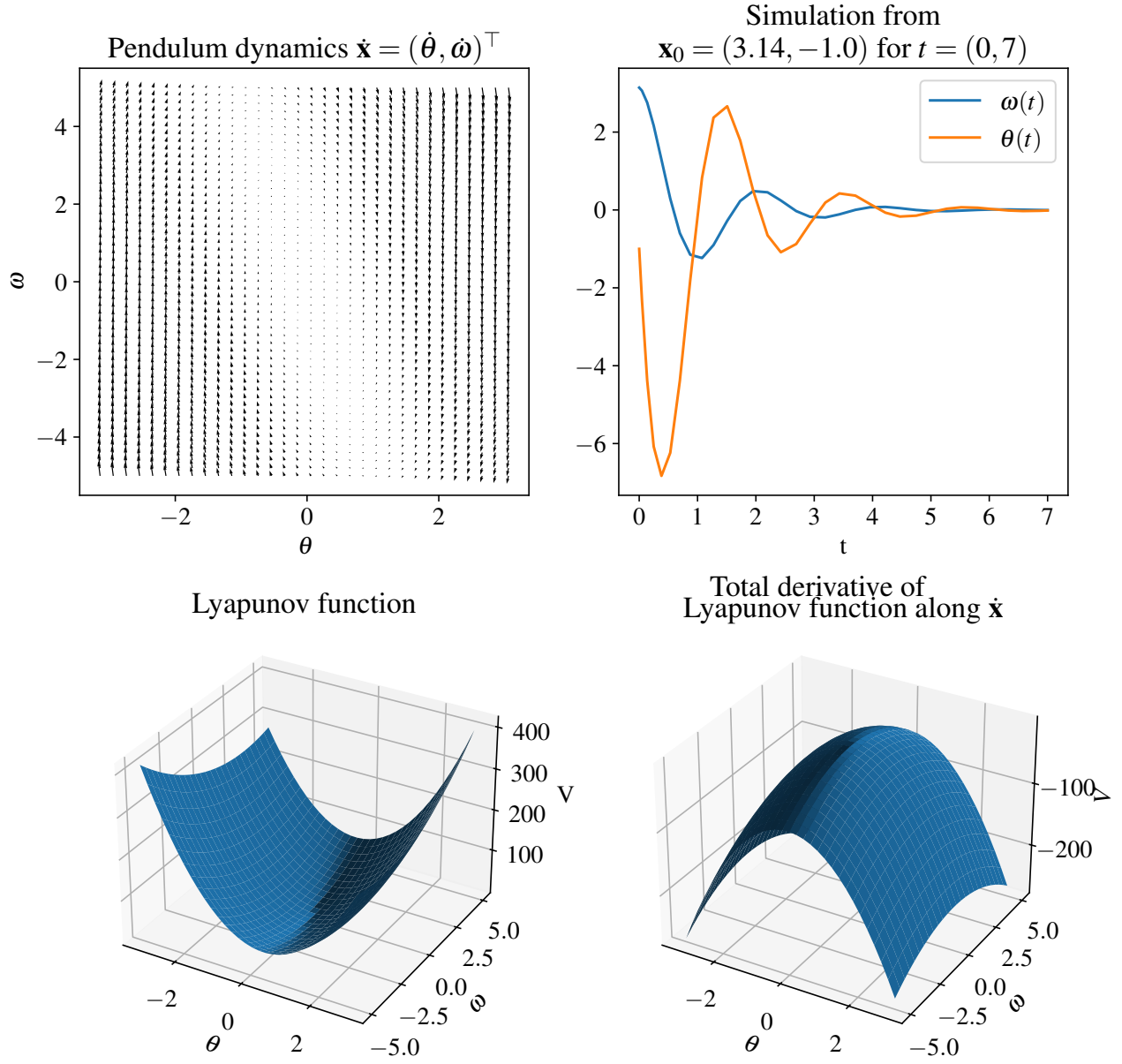


Figure 5: The non-trigonometric variant according to (9)

KeYmaera X

Theme Help ? [Icons]

custom (227,-1) custom (232,1) WL (235,0) dC (242,0) $\wedge R$ (247,1) custom (252,-1) $K<\&>$ (274,0) $K<\&>$ (274,1) WR (289,0) QE (329,-1) compatCut (299,0) compatCut (299,1)

Expand more details

wellformed(th_0,w_0)	$\vdash$	$\langle \{th'=w, w'=d^*th+b^*w\} \rangle \text{ in}(th,w,ep)$
dl>0		$[\{th'=w, w'=d^*th+b^*w\}](\neg \text{in}(th,w,ep))$
$\{th'=w, w'=d^*th+b^*w\}$		
in(th,w,1)		
in(th,w,dl)		
ep>0		
$\forall th \forall w (\text{in}(th,w, 1) \rightarrow V(th,w) \geq \text{bot})$		
comj		
unto		
cust		
Full :		
$\forall L$		
$\forall L$		
unto		
$\exists R$		
unto		
$\rightarrow L$		
$\forall L$		
Expc		
WR		
cust		
$\rightarrow R$		
$\wedge R$		
WH		
cut		
Expc		
$\rightarrow R$		
>		

Figure 6: Navigating the finalized dL sequent proof in the KeYmaera X tool