Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-12 (2025) 61–66

Practical Bounds of Dynamic System Simulator for Stability Verification ⋆

Jongrae Kim *

* *School of Mechanical Engineering, University of Leeds, Leeds LS2 9JT, UK (e-mail: menjkim@leeds.ac.uk).*

Abstract: There are recent shifts in demand for design controllers from simple to complex dynamic systems. Stability verification is one of the important steps in finalising the design of control systems. Stability checks including the complexity of dynamic systems have become increasingly important. Growing computational power provides opportunities to implement efficient tools to address stability verification. To provide the stability confirmation for dynamic simulators, we first establish state propagation bounds for the simulator implemented using the Runge-Kutta 4th-order method. Secondly, we establish computational methods to verify the stability of dynamic systems with finite numbers of simulations over the given range of state space using the state propagation bounds. The algorithms provide the deterministic stability guarantee for the continuous state space. Finally, we demonstrate the effectiveness and limitations of the algorithms using an inverted pendulum system with a reinforcement learning controller combined with an LQR (Linear Quadratic Regulator) controller.

Keywords: Stability, Simulation, Runge-Kutta Integrator, Reinforcement Learning, LQR

1. INTRODUCTION

Almost all classical control design approaches rely on simplified dynamic model descriptions for complex real-world systems. These approaches have made significant success in many engineering system developments over the past few decades (Lehtomaki et al., 1981; Doyle et al., 1989; Utkin, 1977; Morari and Lee, 1999). However, these control system designs must undergo costly, labour-intensive, and tedious system verification procedures through computational (Kapinski et al., 2016) or experimental methods (Karimi et al., 2010; Chen et al., 2019).

Several simulation-based control design approaches have been on the rise. Examples include reinforcement learning algorithms by Sutton et al. (1999) and Lillicrap et al. (2015). The application of reinforcement learning to challenging optimal control problems is particularly noticeable in robotics. For example, the contact dynamics of robot manipulators are too complex to include explicitly in the control design steps (Vukobratovic et al., 2003). The nature of simulation-based control design approaches of reinforcement learning makes it the ideal control design tool for complex systems.

The lack of assurances about the stability and robustness of simulation-based control systems made it challenging to deploy the designs in safety-critical systems such as aircraft, spacecraft and rockets. Researchers have suggested various approaches to guarantee the stability and safety of the reinforcement learning control. Berkenkamp et al. (2017) present a stability-guaranteed reinforcement learning design method using a Lyapunov stability verification approach, where the systems with discontinuities are excluded. Han et al. (2020) propose a stability-guaranteed actor-critic reinforcement learning approach for the systems modelled by the Markov decision process, and they verified the methods using numerical simulations. Osinenko et al. (2022) emphasise the importance of merging model-based control methods with reinforcement learning to achieve a performance guarantee. And, Brunke et al. (2022) highlight the four major issues of safe learning control in robotics related to complex dynamics, sensor noise, sampling efficiency of collecting data and strong modelling assumptions.

This paper focuses on the inclusion of complex dynamics and the relaxation of modelling assumptions. Although the controller is designed to provide a stability guarantee, some of the assumptions in the design process might be too strong. And, it is common that high-fidelity dynamic simulators are used to verify the design in engineering projects. For example, Lee et al. (2021) present the rotor and wing combined models for a compound aircraft simulator. A detailed quad-copter vehicle simulator for abnormal simulations including the full rigid-body dynamics, the propulsion model, the aerodynamics model and the low-level controller has been shown in Foster and Hartman (2017). And, Hieb (2013) demonstrates design procedures for implementing a simulator model of an electric motor producing the same responses as the real motor.

While there are several immediate challenges in implementing accurate and computationally efficient numerical simulators, it is a common practice to utilize high-fidelity simulators to verify the stability, performance and

⋆ This work was funded by Unmanned Vehicles Core Technology Research and Development Program (No. 2020M3C1C1A0108316111) through the National Research Foundation of Korea (NRF) and Unmanned Vehicle Advanced Research Center(UVARC) funded by the Ministry of Science and ICT, Republic of Korea.
** The earlier version of the paper is published in arXiv:2311.11372.

10.1016/j.ifacol.2025.09.568

robustness of control systems, particularly in the industry within the aerospace engineering sectors (Schwartz and Hall, 2003; Yoo et al., 2024; Qi et al., 2024).

In the following sections, we derive the norm bound of states propagated by a numerical simulator. A good estimation of the norm bounds is critical for stability verification. Based on the norm condition, a stability assurance algorithm using a finite number of simulations is presented. The limitations of the algorithm are identified in terms of the magnitude of discontinuities in the dynamic systems. The control system example of an inverted pendulum with the reinforcement learning control algorithm shows the effectiveness of the proposed method. Finally, the conclusions and future works are presented.

2. STATE BOUNDS OF NUMERICAL SIMULATOR

Consider the following integral-type nonlinear system

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \int_{\tau=t}^{\tau=t+\Delta t} \mathbf{f}[\mathbf{x}(\tau)]d\tau \tag{1}$$

where $\mathbf{x}$ is a real-valued n-dimensional vector in the n-dimensional real space, $\mathbb{R}^n$, $\mathbf{x}(t)$ is the value of $\mathbf{x}$ at time, t, $\mathbf{f}(\mathbf{x})$ is Lebesgue integrable and there exist non-negative L and M satisfying the following inequality:

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\| + M \tag{2}$$

for any $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{D}$, $\mathbb{D} = \{\mathbf{x} \mid \|\mathbf{x}\| < r\}$ for $r > 0$, L corresponds to the Lipschitz constant and M is the maximum possible discontinuity of $\mathbf{f}(\mathbf{x})$.

For $L > 0$ and $M = 0$, (1) is called the Carathéodory solution (Trumpf and Mahony, 2010), which considers, however, $\mathbf{f}(t, \mathbf{x})$, where it can be discontinuous in t as long as $\mathbf{f}(t, \mathbf{x})$ is Lebesgue integrable. In many engineering systems, discontinuities depend on the states rather than time, and we consider the nonlinear system specified by (1) and (2).

Definition 1. (State-Transition Function). The state $\mathbf{x}$ at $t+\Delta t$ for (1) is obtained by the state-transition function, $\mathbf{\Phi}$, as follows:

$$\mathbf{x}(t+\Delta t) = \mathbf{\Phi}[t+\Delta t, t; \mathbf{x}(t)] \tag{3}$$

The numerical implementation of (3) using the Runge-Kutta 4-th order method is (Press et al., 1988)

$$\mathbf{\Phi}[t+\Delta t, t; \mathbf{x}(t)] = \mathbf{x}(t) + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{4}$$

where $\mathbf{\Phi}[t+\Delta t, t; \mathbf{x}(t)]$ is the state transition function from t to $t+\Delta t$ starting at the initial state, $\mathbf{x}$, at t and

$$\begin{aligned}
\mathbf{k}_1[\Delta t, \mathbf{x}(t)] &= \mathbf{f}[\mathbf{x}(t)] \\
\mathbf{k}_2[\Delta t, \mathbf{x}(t)] &= \mathbf{f}[\mathbf{x}(t) + \mathbf{k}_1\Delta t/2] \\
\mathbf{k}_3[\Delta t, \mathbf{x}(t)] &= \mathbf{f}[\mathbf{x}(t) + \mathbf{k}_2\Delta t/2] \\
\mathbf{k}_4[\Delta t, \mathbf{x}(t)] &= \mathbf{f}[\mathbf{x}(t) + \mathbf{k}_3\Delta t]
\end{aligned}$$

□

The function implemented in a computer simulator, $\mathbf{f}(\mathbf{x})$, typically includes nonlinear and discontinuous components such as saturation, friction, backlash, hysteresis, deadband and so forth. The numerical result obtained by (4) is frequently the only available solution of (1). Without loss of generality, the equilibrium point concerning the stability is set to zero.

Assumption 1. We assume that the category of nonlinear systems given by (1) and (2) has a unique solution. In addition, the numerical implementation in (4) provides a numerical solution sufficiently close to the solution for an appropriately chosen simulation step size, Δt.

Using (2) and (4), the following theorem establishes a bound of the state-transition function.

Theorem 1. (Bounds for $\Delta\mathbf{\Phi}$). For any $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{D}$

$$\|\mathbf{\Phi}(t+\Delta t, t; \mathbf{x}) - \mathbf{\Phi}(t+\Delta t, t; \mathbf{y})\| \le a\|\mathbf{x}-\mathbf{y}\| + b \tag{5}$$

where $a = (1 + L\alpha)$, $b = \alpha M$ and

$$\alpha = \Delta t\left[1 + \frac{L\Delta t}{2} + \frac{(L\Delta t)^2}{6} + \frac{(L\Delta t)^3}{24}\right] \tag{6}$$

Proof: By the definition of $\mathbf{\Phi}$ in (4),

$$\begin{aligned}
&\|\mathbf{\Phi}(t+\Delta t, t; \mathbf{x}) - \mathbf{\Phi}(t+\Delta t, t; \mathbf{y})\| \\
&\quad \le \|\mathbf{x}-\mathbf{y}\| + \frac{\Delta t}{6}(\Delta\mathbf{k}_1 + 2\Delta\mathbf{k}_2 + 2\Delta\mathbf{k}_3 + \Delta\mathbf{k}_4)
\end{aligned}$$

where $\Delta\mathbf{k}_i = \|\mathbf{k}_i(\Delta t, \mathbf{x}) - \mathbf{k}_i(\Delta t, \mathbf{y})\|$ for $i = 1, 2, 3$ and 4. Each $\Delta\mathbf{k}_i$ is bounded as follows:

$$\begin{aligned}
\Delta\mathbf{k}_1 &= \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\| + M \\
\Delta\mathbf{k}_2 &= \|\mathbf{f}[\mathbf{x} + \mathbf{k}_1(\Delta t, \mathbf{x})\Delta t/2] - \mathbf{f}[\mathbf{y} + \mathbf{k}_1(\Delta t, \mathbf{y})\Delta t/2]\| \\
&\le L\|\mathbf{x}-\mathbf{y}\| + \frac{L\Delta t}{2}\|\mathbf{k}_1(\Delta t, \mathbf{x}) - \mathbf{k}_1(\Delta t, \mathbf{y})\| + M \\
&\le L\left(1 + \frac{L\Delta t}{2}\right)\|\mathbf{x}-\mathbf{y}\| + \left(1 + \frac{L\Delta t}{2}\right)M \\
\Delta\mathbf{k}_3 &= \|\mathbf{f}[\mathbf{x} + \mathbf{k}_2(\Delta t, \mathbf{x})\Delta t/2] - \mathbf{f}[\mathbf{y} + \mathbf{k}_2(\Delta t, \mathbf{y})\Delta t/2]\| \\
&\le L\left(1 + \frac{L\Delta t}{2} + \frac{L^2\Delta t^2}{4}\right)\|\mathbf{x}-\mathbf{y}\| \\
&\quad + \left(1 + \frac{L\Delta t}{2} + \frac{L^2\Delta t^2}{4}\right)M
\end{aligned}$$

and

$$\begin{aligned}
\Delta\mathbf{k}_4 &= \|\mathbf{f}[\mathbf{x} + \mathbf{k}_3(\Delta t, \mathbf{x})\Delta t] - \mathbf{f}[\mathbf{y} + \mathbf{k}_3(\Delta t, \mathbf{y})\Delta t]\| \\
&\le L\left(1 + L\Delta t + \frac{L^2\Delta t^2}{2} + \frac{L^3\Delta t^3}{4}\right)\|\mathbf{x}-\mathbf{y}\| \\
&\quad + \left(1 + L\Delta t + \frac{L^2\Delta t^2}{2} + \frac{L^3\Delta t^3}{4}\right)M
\end{aligned}$$

Hence,

$$\begin{aligned}
&\Delta\mathbf{k}_1 + 2\Delta\mathbf{k}_2 + 2\Delta\mathbf{k}_3 + \Delta\mathbf{k}_4 \\
&\le L\left(6 + 3L\Delta t + L^2\Delta t^2 + \frac{L^3\Delta t^3}{4}\right)\|\mathbf{x}-\mathbf{y}\| \\
&\quad + \left(6 + 3L\Delta t + L^2\Delta t^2 + \frac{L^3\Delta t^3}{4}\right)M \\
&= L\left(\frac{6}{\Delta t}\alpha\right)\|\mathbf{x}-\mathbf{y}\| + \left(\frac{6}{\Delta t}\alpha\right)M
\end{aligned}$$

Therefore,

$$\frac{\Delta t}{6}(\Delta\mathbf{k}_1 + 2\Delta\mathbf{k}_2 + 2\Delta\mathbf{k}_3 + \Delta\mathbf{k}_4) \le L\alpha\|\mathbf{x}-\mathbf{y}\| + \alpha M$$

and (5) is obtained. ■

Given the inequality condition for $\mathbf{f}(\mathbf{x})$ in (2), the transition function is bounded by (5), where α is a function of Δt. We cannot, however, make the upper bound in (5) arbitrarily small by decreasing Δt. As Δt is the simulation step, the smaller Δt is, the longer the computation time to finish the simulation is.

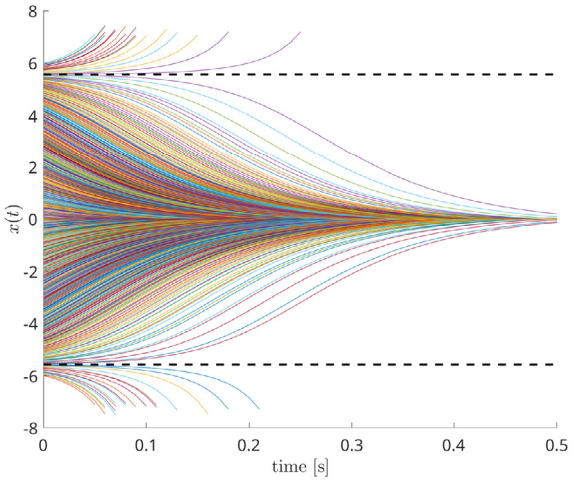

Fig. 1. Example 1 nonlinear system 500 states history

Example 1. Consider a nonlinear system

$$\dot{x} = -2\,\mathrm{sgn}(x) - 10x + \frac{x^3}{3} \tag{7}$$

As shown by the time histories of the states in Figure 1, all $|x(0)|$ less than 5.57 converge to the equilibrium point, 0. Otherwise, it diverges to $\pm\infty$. For $\mathbb{D} = \left\{x \mid |x| < \sqrt{20}\right\}$ and $\Delta t = 0.01$, the bound is as follows:

$$\begin{aligned}
&|f(x) - f(y)| \\
&\le 2|\mathrm{sgn}(x) - \mathrm{sgn}(y)| + 10|x-y| + \frac{1}{3}|x^3 - y^3| \\
&\le 2 \times 2 + 10|x-y| + \frac{1}{3} \times \left(\max_{x\in\mathbb{D}} \left| \frac{dx^3}{dx} \right| \right) |x-y| \\
&= 4 + 10|x-y| + (\sqrt{20})^2 |x-y| = 30|x-y| + 4
\end{aligned}$$

i.e., $L = 30$ and $M = 4$. Therefore,

$$\begin{aligned}
&\|\mathbf{\Phi}(t+\Delta t, t; x) - \mathbf{\Phi}(t+\Delta t, t; y)\| \\
&= a\|x-y\| + b = 1.35\|x-y\| + 0.047
\end{aligned}$$

In the following section, we develop procedures for verifying stability using the bound established in (5).

3. STABILITY VERIFICATION

There are many practical methods to construct a candidate Lyapunov function. We assume the candidate Lyapunov function has an energy function form given by

$$V(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T P\, \mathbf{x} \tag{8}$$

where P is an $n \times n$ matrix and positive-definite, whose maximum eigenvalue is equal to $\lambda_{\max} > 0$.

Definition 2. (Lyapunov Stability). Let $\mathbb{S}$ be a subset of $\mathbb{D}$, whose corresponding energy is less than a positive real number, ℓ, as follows:

$$\mathbb{S} = \{\mathbf{x} | V(\mathbf{x}) \le \ell\} \tag{9}$$

where the equilibrium point, $\mathbf{x}_{\mathrm{eq}}$, belongs to $\mathbb{S}$ and the following inequality is satisfied for all $\mathbf{x}$ in $\mathbb{S}$

$$V[\mathbf{\Phi}(t+\Delta t, t; \mathbf{x})] \le V(\mathbf{x}) \le \ell, \tag{10}$$

then $\mathbf{x}_{\mathrm{eq}}$ is *stable*.

□

By the definition, the Lyapunov function slope is bounded by

$$\left\| \frac{\partial V}{\partial \mathbf{x}} \right\| = \|P\mathbf{x}\| \le \lambda_{\max} r \tag{11}$$

for all $\mathbf{x} \in \mathbb{S}$.

Example 2. For the nonlinear system given in Example 1, let $\mathbb{S} = \{x \mid V(x) = x^2/2 \le 10\}$, where $P = 1$, $\ell = 10$, $\lambda_{\max} = 1$, $r = \sqrt{20}$ and $\partial V/\partial x \le \sqrt{20}$. The equilibrium point, $x_{\mathrm{eq}} = 0$, is stable if $V[\Phi(t+\Delta t, t; x)] \le 10$ for all $x \in \mathbb{S}$.

Theorem 2. (Bound for $\Delta V(\mathbf{x}, \mathbf{y})$). The difference in the Lyapunov function is bounded by

$$\begin{aligned}
&\Delta V(\mathbf{x}, \mathbf{y}) = |V[\mathbf{\Phi}(t+\Delta t, t; \mathbf{x})] - V[\mathbf{\Phi}(t+\Delta t, t; \mathbf{y})]| \\
&\le \lambda_{\max} r (a\|\mathbf{x} - \mathbf{y}\| + b)
\end{aligned} \tag{12}$$

for all $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{S}$.

Proof: By the slope bound, the difference is bounded by

$$\begin{aligned}
&|V[\mathbf{\Phi}(t+\Delta t, t; \mathbf{x}) - V[\mathbf{\Phi}(t+\Delta t, t; \mathbf{y})]| \\
&\le \lambda_{\max} r \|\mathbf{\Phi}(t+\Delta t, t; \mathbf{x}) - \mathbf{\Phi}(t+\Delta t, t; \mathbf{y})\|
\end{aligned}$$

and because of (5), the last term is also bounded by

$$\begin{aligned}
&\lambda_{\max} r \mathbf{\Phi}(t+\Delta t, t; \mathbf{x}) - \mathbf{\Phi}(t+\Delta t, t; \mathbf{y})\| \\
&\le \lambda_{\max} r\, (a\|\mathbf{x} - \mathbf{y}\| + b)
\end{aligned}$$

■

Example 3. For the nonlinear system given in Example 1 and $\mathbb{S}$ given in Example 2,

$$\begin{aligned}
&|V[\Phi[t+\Delta t, t; x] - V[\Phi[t+\Delta t, t; y]| \\
&= \sqrt{20}\,(1.35|x-y| + 0.047) = 6.04|x-y| + 0.21
\end{aligned}$$

for all x and y in $\mathbb{S}$.

To verify the stability of the equilibrium point in $\mathbb{S}$, we sample a finite number of points in $\mathbb{S}$. A set is called a δ-sampling set if, for all $\mathbf{x}$ in $\mathbb{S}$, at least one sample is within distance δ from $\mathbf{x}$, and the set contains all such sampling points.

Definition 3. (δ-sampling set $\mathbb{S}_\delta$). For a positive δ, the δ-sampling set, $\mathbb{S}_\delta$, is defined by

$$\mathbb{S}_\delta = \{\mathbf{x}_\delta \mid \|\mathbf{x} - \mathbf{x}_\delta\| \le \delta \text{ for all } \mathbf{x} \in \mathbb{S}\} \tag{13}$$

□

Example 4. For the set $\mathbb{S}$ given in Example 2, firstly, divide the range of x in $\mathbb{S}$ into n_s equal intervals, e.g., n_s is set to 500 and the length of each interval, δ, is $2\sqrt{20}/500 = 0.0179$. Secondly, sample 500 x_δ points with the interval 0.0179 starting from $-\sqrt{20}$. Finally, perturb each x_δ by δ_p, which is a random number from the uniform distribution between 0 and 0.0179, i.e.,

$$x_\delta \leftarrow x_\delta + \delta_p$$

where the random perturbation is to avoid the deterministic grid sampling. Then, for any x in $\mathbb{S}$, there is x_δ in the range of δ, i.e.,

$$\mathbb{S}_\delta = \{x_\delta \mid |x - x_\delta| \le \delta = 0.0179 \text{ for all } x \in \mathbb{S}\}$$

Theorem 3. (Verification of Forward Invariant). For all $\mathbf{x}_\delta$ in $\mathbb{S}_\delta$, if there exists a positive real γ such that

$$V[\mathbf{\Phi}(t+\Delta t, t; \mathbf{x}_\delta)] \le \ell - \gamma \tag{14}$$

and δ satisfies

$$\lambda_{\max} r\, (a\delta + b) \le \gamma \tag{15}$$

then $\mathbb{S}$ is forward invariant (Kapinski and Deshmukh, 2015).

Proof: Prove it by contradiction as in Kapinski and Deshmukh (2015). Assume (14) and (15) are satisfied but there exists $\mathbf{x}^*$ in $\mathbb{S}$ such that $V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}^*)] > \ell$. Multiply -1 for both sides

$$-V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}^*)] < -\ell$$

Add (14) and (16)

$$V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}_\delta)] - V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}^*)] < -\gamma < 0$$

Hence,

$$\gamma < |V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}_\delta)] - V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}^*)]|$$

Because the energy slope bound is given by (12) and $\mathbf{x}_\delta$ belongs to the δ-sampling set, $\mathbb{S}_\delta$, the right-most term is bounded by (15)

$$\begin{aligned}\gamma &< |V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}_\delta)] - V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}^*)]| \\ &\le \lambda_{\max} r\,(a\|\mathbf{x}_\delta - \mathbf{x}^*\| + b) \le \lambda_{\max} r\,(a\delta + b) \le \gamma\end{aligned}$$

The inequality contradicts as $\Delta V(\mathbf{x}_\delta, \mathbf{x}^*)$ is strictly greater than γ and less than equal to γ simultaneously. Hence, the assumption of the existence of $\mathbf{x}^*$ such that $V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}^*)] > \ell$ is incorrect. ■

In the forward invariant verification step, γ cannot be an arbitrary value but is determined by the maximum value of $V(\mathbf{x})$ as follows:

$$\hat{\gamma} = \ell - \hat{V}(\mathbf{x}_\delta) \tag{16}$$

where

$$\hat{V}(\mathbf{x}_\delta) = \max_{\mathbf{x}_\delta \in \mathbb{S}_\delta} V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x}_\delta)]$$

(Kapinski and Deshmukh, 2015).

Example 5. With the values given in Examples 1, 2, 3 and 4, run the numerical simulation of the nonlinear system given in (7) using the Runge-Kutta 4-th order integration for each sample generated in Example 4. Update $\hat{\gamma}$ by the maximum $V[\Phi(t+\Delta t;t;x_\delta)]$ found by the simulations. If $\ell - \hat{\gamma}$ is less than or equal to zero or the inequality in (15) is violated, then the stability verification fails. If both of the inequality conditions, (14) and (15), are satisfied for all $\mathbf{x}_\delta$, the invariant set is verified for $\mathbb{S}_\delta$. In this example, $\hat{\gamma}$ obtained is 9.22. $\ell - \hat{\gamma} = 10 - 9.22 = 0.78$ is greater than zero and $\lambda_{\max} r\,(a\delta + b) = 0.425$ is less than $\hat{\gamma}$.

The estimated γ, i.e., $\hat{\gamma}$, relies on the sampled maximum value, $\hat{V}(\mathbf{x}_\delta)$. The true maximum in $\mathbb{S}$, i.e.,

$$V(\mathbf{x}^*) = \max_{\mathbf{x}^* \in \mathbb{S}} V[\mathbf{\Phi}(t+\Delta t,t;\mathbf{x})]$$

is larger than or equal to $\hat{V}(\mathbf{x}_\delta)$. The true maximum value is bounded as follows:

Theorem 4. (Bound for $V(\mathbf{x}^*)$). $V(\mathbf{x}^*)$ is bounded by

$$V(\mathbf{x}^*) \le \bar{V}(\mathbf{x}^*)$$

where $\bar{V}(\mathbf{x}^*) = \hat{V}(\mathbf{x}_\delta) + \lambda_{\max} r\ \bar{\delta}$ and $\bar{\delta} = \max \|\mathbf{x}^* - \mathbf{x}_\delta\|$

Proof: By the slope bound in (11) and the definition of δ-sampling in Definition 3, there exists at least one $\mathbf{x}_\delta$ in $\mathbb{S}_\delta$ within the distance less than δ from $\mathbf{x}^*$ satisfying the following inequality:

$$V(\mathbf{x}^*) - V(\mathbf{x}_\delta) \le \lambda_{\max} r \|\mathbf{x}^* - \mathbf{x}_\delta\| \le \lambda_{\max} r\ \bar{\delta}$$

Rearrange it and by the definition of $\hat{V}(\mathbf{x}_\delta)$

$$V(\mathbf{x}^*) \le V(\mathbf{x}_\delta) + \lambda_{\max} r\ \bar{\delta} \le \hat{V}(\mathbf{x}_\delta) + \lambda_{\max} r\ \bar{\delta}$$

■

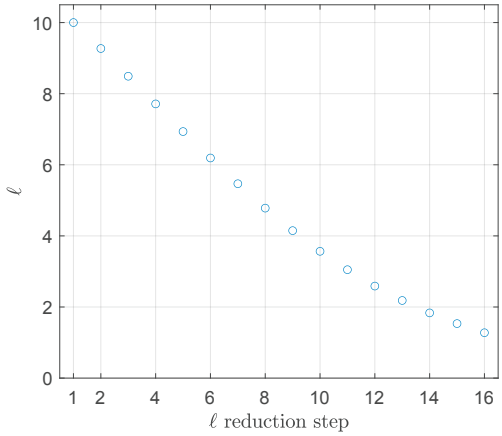

Fig. 2. ℓ reduction for each step in Example 1

Example 6. The true maximum, $V(x^*)$, in Example 5 is bounded by $\bar{V}(x^*)$ and it is equal to

$$\bar{V}(x^*) = 9.22 + 1 \times \sqrt{20} \times \bar{\delta}$$

Based on the upper bound of the true maximum, 9.3 in this case, we can confirm the condition, γ, is positive as $\ell - \bar{V}(x^*) = 0.78 - 4.47\bar{\delta}$. Hence, as long as $\bar{\delta} < 0.174$, $\hat{\gamma}$ is positive. Once the inequality, (15), is confirm to be satisfied with $\hat{\gamma}$. We can confirm that all the propagated states from any x in $\mathbb{S}$ have their corresponding $V(x)$ less than or equal to 9.3. Update ℓ equal to 9.3, redefine $\mathbb{S}$ with the updated ℓ, sample x_δ from $\mathbb{S}$ and repeat the process for verifying the forward invariant. We keep the sample numbers equal to 500 and perform the procedures 16 times starting from $\ell = 10$. Figure 2 shows the reduction of ℓ for each step.

When the final ℓ is not small enough to the equilibrium point in the performance aspects of the system, then further ℓ reduction procedures with a smaller δ, i.e., more number of the samples, will be performed, which reduces the left-hand side of the values in the inequality, (15). There is, however, a fundamental limitation caused by the discontinuities in the system. The inequality condition in (15) is bounded below by

$$\lambda_{\max} r b \le \lambda_{\max} r\,(a\delta + b) \le \gamma \tag{17}$$

When there are infinite samples in $\mathbb{S}$, i.e., $\delta = 0$, the lower bound is the minimum to achieve. If the lower bound is greater than γ, we cannot verify the invariant property and stability using the proposed method. This is one of the fundamental limitations caused by the discontinuities in the system.

Example 7. In Example 6, ℓ decreases every step until the last step shown in Figure 2. The last value of ℓ is 1.28 corresponding to the set $\mathbb{S}$ with $|x| < 2.26$. Further reduction of ℓ is possible with the increased number of samples by 10 times, i.e., 5000 samples and reduce r with new ℓ found. ℓ reduces further to 0.040. However, further increase of n_s, i.e., decrease of δ, will not reduce ℓ. γ found is 0.0128 and $\lambda_{\max} r b$ is 0.0129. Hence, even for $\delta = 0$, the inequality, (17), cannot be satisfied.

The summary of the invariant verification procedure is in Algorithm 1, where the input variables are the number of samples, n_s, the bound for $\mathbb{S}$, ℓ, the simulation integration step, Δt, and P for the Lyapunov function. The loop

Algorithm 1 Invariant Verification(n_s, ℓ, Δt, P)

Set $\hat{V} = -\infty$ and $n_c = 0$ and Generate $\mathbf{x}_\delta \in \mathbb{S}_\delta \subset \mathbb{S}$
for $k = 1, 2, \ldots, n_s$ **do**
  Propagate $\mathbf{x}_\delta^k$ in $\mathbb{S}_\delta$ using (4)
  Calculate $V_{\text{new}} = V[\mathbf{\Phi}(t, t + \Delta t; \mathbf{x}_\delta^k)]$
  if $V_{\text{new}} > \hat{V}$ **then**
    $\hat{V} \leftarrow V_{\text{new}}$
  end if
  $\gamma \leftarrow \ell - \hat{V}$
  if γ does not satisfy (15) **then**
    Terminate the loop
  else
    $n_c \leftarrow n_c + 1$
  end if
end for
if n_c is equal to n_s **then**
  Declare the forward invariant is verified
else
  Declare no verification
end if
Terminal Condition The loop terminates when all n_s number of $\mathbf{x}_\delta$ are checked.

Algorithm 2 Stability Verification

Set $n_{\text{nr}} = 0$, n_s, Δt, P, ℓ and IsReduced = True
Set the number of maximum no reduction in ℓ, $\bar{n}$
while $n_{\text{nr}} \leq \bar{n}$ **do**
  if IsReduced = False **then**
    Increase n_s
  end if
  IsReduced $\leftarrow$ False
  $n_{\text{nr}} \leftarrow n_{\text{nr}} + 1$
  Run Invariant Verification(n_s, ℓ, Δt, P)
  if invariant verified and $\bar{V}(\mathbf{x}^*) < \ell$ **then**
    $\ell \leftarrow \bar{V}(\mathbf{x}^*)$
    IsReduced$\leftarrow$True, $n_{\text{nr}} \leftarrow 0$
  end if
end while
Terminal Condition The loop terminates if ℓ does not decrease $\bar{n}$ consecutive times.

terminates when it finishes checking n_s samples. The summary of the stability check procedure is in Algorithm 2. The algorithm terminates if ℓ does not reduce for $\bar{n}$ times in a row.

4. INVERTED PENDULUM

The dynamics of an inverted pendulum is given by

$$\ddot{\theta} = \frac{g}{l}\sin\theta + \frac{1}{ml^2}u \tag{18}$$

where the gravitational acceleration, g, is equal to 9.821 m/s^2, the length of the pendulum, l, is equal to 1.3 m, the mass of the pendulum, m, is equal to 2.2 kg and u is the control torque in Nm, whose magnitude is restricted to the ± 12 Nm range. The control algorithm has two parts: LQR (Linear Quadratic Regulator) and reinforcement learning.

The LQR controller is given by

$$u_{\text{LQR}} = -10\,\theta - 10\,\dot{\theta} \tag{19}$$

The LQR acts on the system if the control command magnitude is within the range of ± 12 Nm. Otherwise, the reinforcement learning algorithm acts for all the other cases. The detailed control design is shown in Kim (2022).

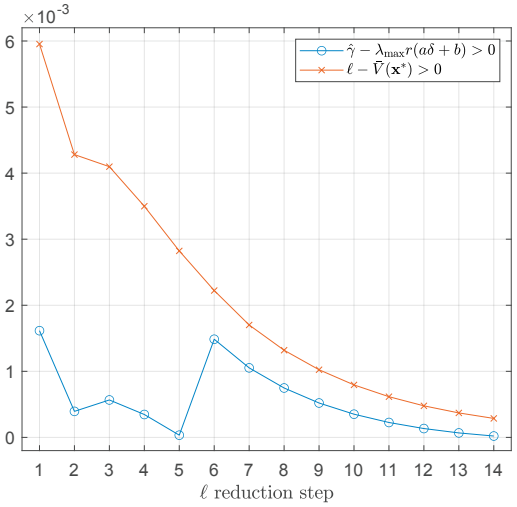

Fig. 3. The inequality conditions in Example (18)

The reinforcement learning controller has seven sensor inputs. The two sensors are θ and $\dot{\theta}$. The other five are accelerometers attached to the pendulum at an equal distance. The reinforcement learning controller has four layers of artificial neural networks. The input layer has 800 nodes. The second, third and last output layers have 600 nodes. The reinforcement learning algorithm is the DDPG (Deep Deterministic Policy Gradient) reinforcement learning method in Lillicrap et al. (2015). The activation functions of the first three layers are ReLU (Rectified Linear Unit). The output layer activation function is the $\tanh(\cdot)$ function.

The simulator propagates the states every 0.01 s to achieve tolerable numerical errors. It is, however, too short for the states to make any meaningful magnitude reduction for the given δ determined by the number of samples. For every 0.3 s, i.e., 10 integration steps, we check the reduction of ℓ. The state-transition function propagates the states 30 times with 0.01 s intervals while the bounds are calculated for $\Delta t = 0.3$. This is another limitation of the algorithm. We cannot extend the propagation length too long. It will cause a large increase in the bounds and the difference in the states between 30 times propagation with 0.01 s and one-time propagation with 0.3 s.

The stability check procedure starts from $|\theta| \leq 10^\circ$ and $|\dot{\theta}| \leq 10^\circ$/s. Initially, the number of $\mathbf{x}_\delta$ samples is 2,250k, which produces $\delta = 0.00033$. The while-loop in Algorithm 2 repeats 14 times. Figure 3 shows the two inequality conditions, where P is the 2×2 identity matrix, $\lambda_{\max} = 1$ and $r = 10\pi/180$. The algorithm runs in parallel on the NVIDIA Jetson AGX 64GB Orin Developer Kit, which has 2048 NVIDIA CUDA cores. The algorithm is implemented using the CUDA package in Julia (Besard et al., 2018). The calculation takes about 8 minutes to complete the while-loop in Algorithm 2.

5. CONCLUSIONS & FUTURE WORKS

We present a stability verification method to provide deterministic stability assurance of dynamical systems im-

plemented as numerical simulators. The simulator may include strong nonlinear components such as discontinuous jumps in the states and simulation-based control design algorithms. The proposed bounds directly connect to the numerical simulators and the parameters used in the simulations. It reveals a fundamental limitation of the stability check for discontinuous components in the function of states. Immediate future research is to derive tighter bounds for the longer propagation horizon. Current works include applying the stability checking algorithm to high-fidelity dynamic simulators.

ACKNOWLEDGEMENTS

I would like to express my appreciation to the National Research Foundation of Korea (NRF), Unmanned Vehicle Advanced Research Center(UVARC) and Inha University for providing the grant for the research.

REFERENCES

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. (2017). Safe model-based reinforcement learning with stability guarantees. In I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Besard, T., Foket, C., and De Sutter, B. (2018). Effective extensible programming: Unleashing Julia on GPUs. *IEEE Transactions on Parallel and Distributed Systems*. doi:10.1109/TPDS.2018.2872064.

Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A.P. (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(Volume 5, 2022), 411–444.

Chen, S., Chen, Y., Zhang, S., and Zheng, N. (2019). A novel integrated simulation and testing platform for self-driving cars with hardware in the loop. *IEEE Transactions on Intelligent Vehicles*, 4(3), 425–436. doi: 10.1109/TIV.2019.2919470.

Doyle, J., Glover, K., Khargonekar, P., and Francis, B. (1989). State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8), 831–847. doi:10.1109/9.29425.

Foster, J.V. and Hartman, D. (2017). High-fidelity multi-rotor unmanned aircraft system (UAS) simulation development for trajectory prediction under off-nominal flight dynamics. In *17th AIAA Aviation Technology, Integration, and Operations Conference*, 3271.

Han, M., Zhang, L., Wang, J., and Pan, W. (2020). Actor-critic reinforcement learning for control with stability guarantee. *IEEE Robotics and Automation Letters*, 5(4), 6217–6224.

Hieb, B. (2013). Creating a high-fidelity model of an electric motor for control system design and verification. *Technical Article Published by The MathWorks*.

Kapinski, J. and Deshmukh, J. (2015). Discovering forward invariant sets for nonlinear dynamical systems. In M.G. Cojocaru, I.S. Kotsireas, R.N. Makarov, R.V.N. Melnik, and H. Shodiev (eds.), *Interdisciplinary Topics in Applied Mathematics, Modeling and Computational Science*, 259–264. Springer International Publishing, Cham.

Kapinski, J., Deshmukh, J.V., Jin, X., Ito, H., and Butts, K. (2016). Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques. *IEEE Control Systems Magazine*, 36(6), 45–64.

Karimi, S., Poure, P., and Saadate, S. (2010). A HIL-based reconfigurable platform for design, implementation, and verification of electrical system digital controllers. *IEEE Transactions on Industrial Electronics*, 57(4), 1226–1236. doi:10.1109/TIE.2009.2036644.

Kim, J. (2022). State-space segmentation for faster training reinforcement learning. *IFAC-PapersOnLine*, 55(25), 235–240. 10th IFAC Symposium on Robust Control Design ROCOND 2022.

Lee, D.H., Kim, C.J., and Lee, S.H. (2021). Development of unified high-fidelity flight dynamic modeling technique for unmanned compound aircraft. *International Journal of Aerospace Engineering*, 2021, 1–23.

Lehtomaki, N., Sandell, N., and Athans, M. (1981). Robustness results in linear-quadratic gaussian based multivariable control designs. *IEEE Transactions on Automatic Control*, 26(1), 75–93. doi:10.1109/TAC.1981.1102565.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Morari, M. and Lee, J.H. (1999). Model predictive control: past, present and future. *Computers & chemical engineering*, 23(4-5), 667–682.

Osinenko, P., Dobriborsci, D., and Aumer, W. (2022). Reinforcement learning with guarantees: a review. *IFAC-PapersOnLine*, 55(15), 123–128. 6th IFAC Conf. on Intelligent Control and Automation Sciences ICONS.

Press, W.H., Vetterling, W.T., Teukolsky, S.A., and Flannery, B.P. (1988). *Numerical Recipes in C*. Cambridge University Press, London, England.

Qi, P., Li, Q., and Wang, Y. (2024). Trim analysis and structured h-infinity control of wing-docked multibody aircraft. *Journal of Guidance, Control, and Dynamics*, 47(5), 964–978. doi:10.2514/1.G007835.

Schwartz, J.L. and Hall, C.D. (2003). The distributed spacecraft attitude control system simulator: development, progress, plans. In *Flight Control Symposium: Citeseer*. USA.

Sutton, R.S., Barto, A.G., et al. (1999). Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1), 126–134.

Trumpf, J. and Mahony, R. (2010). A converse liapunov theorem for uniformly locally exponentially stable systems admitting Carathèodory solutions. *IFAC Proceedings Volumes*, 43(14), 1374–1378.

Utkin, V. (1977). Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2), 212–222. doi:10.1109/TAC.1977.1101446.

Vukobratovic, M., Potkonjak, V., and Matijevic, V. (2003). *Dynamics of robots with contact tasks*, volume 26. Springer Science & Business Media.

Yoo, B.W., Park, K.P., and Oh, J. (2024). Dynamics simulation model for the analysis of aircraft movement characteristics on an aircraft carrier deck. *International Journal of Naval Architecture and Ocean Engineering*, 16, 100591.