Fast Option Ranking in Autonomous Systems for Criticality Evasion under Uncertainties

Wild-and-Crazy-Idea Paper

Bineet Ghosh[†], Parasara Sridhar Duggirala[‡], Samarjit Chakraborty[‡]
[†]The University of Alabama, Tuscaloosa, USA, bineet@ua.edu
[‡]The University of North Carolina at Chapel Hill, Chapel Hill, USA, {psd,samarjit}@cs.unc.edu

***Abstract*—We study the problem where an autonomous system is in a critical situation and is faced with multiple options among which it has to choose to *safely* evade the criticality. Each of these options is also associated with some uncertainty. Traditional approaches from formal methods require a *reachability analysis* to evaluate which of the options is safe. While the computational cost of reachability analysis is well known, the presence of uncertainty adds an additional layer of complexity. As a result, performing reachability analysis for all the options before choosing one will not be feasible due to time constraints. This is a practical problem that arises is various scenarios, such as an autonomous vehicle in a potential accident that it has to evade to minimize damage. While models and algorithms for reachability analysis have been widely studied, reachability analysis in the presence of uncertainties have been less so. Despite its many applications, to the best of our knowledge, the problem of choosing in real-time, one of the many options for criticality evasion has not been studied in the past. We address this problem by proposing a new real-time reachable set computation technique for uncertain linear systems using techniques from perturbation theory.**

***Index Terms*—Uncertain Linear Systems, Parameterized Systems, Robust Reachable Sets.**

I. Introduction

Fig. 1 describes the scenario we study in this paper. It depicts a fast-moving autonomous vehicle in a factory warehouse with multiple static and dynamic obstacles. The autonomous vehicle has to navigate through these obstacles, where there are no well-defined paths or lanes. Moreover, there are various types of uncertainties: because of sensor errors, or inference errors in the machine learning (ML) pipeline that perceives the surrounding scene, there are locational uncertainties associated with the various obstacles. There could also be uncertainties in the dynamics of the autonomous vehicle itself, *e.g.,* due to the unknown friction coefficient of the floor. Given such a scenario, how should the autonomous vehicle at the center of Fig. 1 navigate itself without colliding with any of the obstacles? In particular, which of the multiple options in Fig. 1 should it choose? The state-of-the-art approach to this problem will require performing a reachability analysis for each of these options to identify one that meets the required *safety properties*, *e.g.,* a collision-free trajectory. But reachability analysis, *i.e.,* computing the set of states reachable by the system under a chosen dynamics, is known to be computationally expensive. The presence of uncertainties adds an additional layer of modeling and computational complexity to the problem.

While there is a considerable volume of literature on reachability analysis, and less work on reachability analysis under uncertainties, we are not aware of any prior study on fast ranking of options for criticality evasion, of the form shown in Fig. 1. This is despite the common occurrence of this scenario, *e.g.*, when an autonomous vehicle is in a potential accident and has to very quickly decide among multiple maneuvers to minimize damage or avoid the accident.

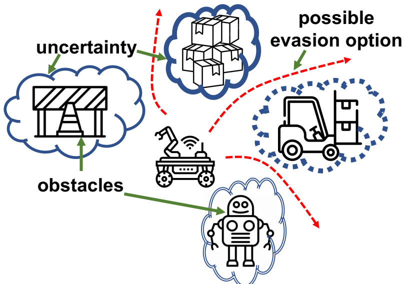

Fig. 1: An autonomous vehicle (center) in a factory warehouse trying to evaluate different trajectories to move without colliding with other obstacles, which have uncertain dynamics and possibly also locational uncertainties.

The goal of this paper is to propose such a real-time *option ranking* strategy, using which the various available options could be very quickly ranked in the order of being safe with decreasing likelihood. This way, a full reachability analysis could be carried out first for the options that are most likely to be safe. Such a ranking can make a significant difference in many time- and safety-critical autonomous systems, and also prove to be invaluable in safety certification. More importantly, our proposed ranking strategy takes into account the different uncertainties associated with each option.

Technical contributions: We consider a continuous-time linear dynamics of the autonomous system under consideration, which is described by $\dot{x} = Ax$, where $x \in \mathbb{R}^n$ represents the n-dimensional state vector, and $A \in \mathbb{R}^{n \times n}$ is the system's nominal dynamics matrix. While this equation models the idealized system behavior, real-world navigation involves a sequence of decisions or choices (as illustrated in Fig. 1), each introducing its own form of uncertainty [1]. These uncertainties are incorporated into the dynamics as additive perturbations to the nominal matrix A.

Specifically, for a given navigation choice i, the associated uncertainty is captured by an interval matrix Λ_i, where each cell is an interval representing a bounded value. Consequently, the effective system dynamics under choice i can be expressed as $\dot{x} = (A + \Lambda_i)x$, showing how the uncertainties from that choice affect the system's evolution.

A concrete example of such a navigation choice is illustrated in Fig. 2, where a robot on a factory floor must avoid colliding

979-8-3315-9379-7/25/$31.00 ©2025 IEEE

with a wall (shown in red). This scenario corresponds to a specific choice i in the robot's navigation strategy, where the robot has the option to navigate through a path where there is an oil spill. Had the robot had a path over a clean floor, the resulting reachable set of states is shown in light blue. However, when the robot chooses a path that goes over the oil spill, additional uncertainty is introduced—reflected by a larger Λ_i—resulting in a larger reachable set (shown in dark blue, encompassing the light blue region). This illustrates how even a single navigation choice can significantly impact the system's behavior through its associated uncertainty.

It is worth noting at this point that the effective dynamics of the system under choice i is no longer strictly linear. This is because the matrix Λ_i is not a point matrix with fixed values, but an interval matrix where each entry represents a range of possible values. As a result, the system behaves like a special kind of nonlinear system. Due to this non-linearity, computing the reachable set for safety verification becomes more expensive. In fact, such computations often become infeasible in situations where the autonomous system must make quick decisions about its navigation, as considered in this work.

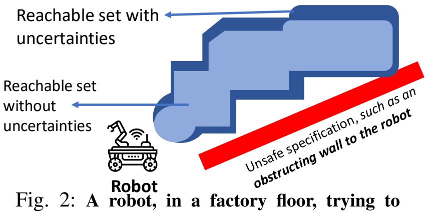
Reachable set with uncertainties
Reachable set without uncertainties
Unsafe specification, such as an obstructing wall to the robot
Robot

Fig. 2: **A robot, in a factory floor, trying to avoid collision with the wall (in red).** The set of states reached by the robot on a clean factory floor is marked in light blue. The set of states reached by the robot with an oil spill on the floor is marked in dark blue (including the region in light blue).

In the scenario considered in this work, where an autonomous system must choose between different navigation paths, each with its own uncertainty, the typical approach involves blindly computing reachable sets for each choice to assess its safety. This process is often done without any guidance on which path to prioritize. That is, there is no indication of which option is more likely to be safe.

Clearly, having prior information about the safety of each path would significantly speed up the process of selecting a safe navigation option. This work aims to address this gap by providing a mathematical framework that quantitatively assesses the impact of uncertainty associated with each navigation choice. By doing so, it allows for the ranking of these choices, enabling more efficient computation of reachable sets and ultimately improving the problem of navigation choice exploration.

Given a navigation choice i, and the associated uncertainty Λ_i, we analyze the effect of this uncertainty on the nominal system dynamics A using results from *perturbation theory*. Our goal is to estimate how much the uncertainty perturbs the reachable set of the system over time. Specifically, we derive an upper bound on the *norm ratio* between the reachable set induced by the nominal dynamics and the additional portion of the reachable set due to the uncertainty (visually represented by the dark blue region in Fig. 2).

To understand this in more detail (discussed further in later sections), consider the following: The reachable set at time t for the nominal system (*i.e.*, without uncertainty) is governed by the matrix exponential e^{At}. When uncertainty is introduced—encapsulated by the interval matrix Λ_i for choice i—the dynamics become $A + \Lambda_i$, and the reachable set is described by $e^{(A+\Lambda_i)t}$. However, directly computing the matrix exponential of an interval matrix is computationally expensive and often impractical. To avoid this, we instead apply perturbation theory to bound the impact of uncertainty. Specifically, we bound the norm of the difference $\left\| e^{(A+\Lambda_i)t} - e^{At} \right\|$, and compare it relative to $\left\| e^{At} \right\|$, giving us a norm ratio that quantifies impact of uncertainty on the reachable set. Once this bound is computed for each available navigation choice, we use it to *prioritize* the reachable set computations. Intuitively, a higher bound indicates that the uncertainty has a greater impact on the system's behavior, potentially increasing safety risks. Therefore, choices with lower bounds are prioritized, as they are less likely to result in unsafe outcomes.

In summary, our approach enables efficient navigation decision-making by ranking choices based on their sensitivity to uncertainty, without the need for expensive interval matrix computations. Section IV concludes the paper by outlining some directions for future work. More details are in [2].

A. Related Work

In [1], a method for efficiently computing the exact reachable set of linear dynamical systems with specific uncertainty classes is presented. Our technique, however, handles arbitrary uncertainties. In [3], a sampling-based approach for uncertain linear systems is proposed. Unlike this, our method is independent of the size of uncertainties and does not rely on sampling. [4] computes reachable sets via discretization and Zonotope representation, while our method, based on perturbation theory, provides an analytical solution. In [5], an algorithm using Zonotopes approximates reachable sets for uncertain systems. Unlike [1], it does not provide an exact reachable set. Our contribution is a fast symbolic approach for ranking of computing reachable sets using perturbation theory. We also acknowledge works on computing reachable sets for non-linear systems, such as [6]–[13]. In a slightly broader setting, this work is also related to our recent efforts on modeling the impact of timing uncertainties in autonomous and cyber-physical systems via approximate reachability analysis [14], [15]. Here, the goal has been to compute the deviation between the dynamics of a closed-loop system under ideal timing behavior (*e.g.*, no deadline misses) and that where the control task occasionally misses its deadline [16]. Such a deviation is used as a measure of safety and has been used for schedule synthesis to ensure system safety under deadline misses [17]–[21], neural network sizing [22], and edge-cloud partitioning [23].

Paper Organization: The next section is the main technical contribution of this paper. This is followed by some experimental results to illustrate the utility of our framework.

II. Fast option Ranking

In the early days of studying numerical techniques for solving ordinary differential equations, there was a significant interest in bounding the errors introduced by using finite precision representation of real numbers. Seminal works in this domain, such as [24] and [25], investigated analytical methods for computing upper bounds on the sensitivity of matrix exponential.

Given a linear dynamical system $\dot{x} = Ax$ and its perturbation $\dot{x} = (A + E)x$, these methods presented expressions to upper bound the relative distance between the trajectories of the nominal (unperturbed) and the perturbed system. The relative distance, denoted by $\phi_{(A,E)}(t)$, is given as:

$$\phi_{(A,E)}(t) = \frac{||\, e^{(A+E)t} - e^{At} \,||}{||\, e^{At} \,||}. \tag{1}$$

The results presented in [24] and [25] are applicable to specific, fixed matrices A and E. However, in the context of our work, each navigation choice is associated with a bounded range of uncertainty rather than a single fixed value. That is, the uncertainties corresponding the to navigation choices are represented using interval matrices Λ instead of fixed matrices like E. As a result, we extend the prior results to handle interval matrices. To do this, we introduce a generalized formulation that captures the effect of uncertainty over the entire interval:

$$\phi_{(A,\Lambda)}(t) = \sup_{E \in \Lambda} \frac{\|e^{(A+E)t} - e^{At}\|}{\|e^{At}\|} \tag{2}$$

To transition from point matrices to interval matrices, we consider all (infinitely many) possible matrices E contained within the interval matrix Λ and take the supremum over the resulting norms. By building on the analytical expressions for $\phi_{(A,E)}(t)$ derived in [24] and [25], we derive analogous expressions for $\phi_{(A,\Lambda)}(t)$ that account for interval uncertainty. This function $\phi_{(A,\Lambda)}(t)$ provides a quantitative measure of the impact of uncertainty associated with a given navigation choice on the system's reachable set. A higher value of this function indicates greater bloating of the reachable set, which in turn implies a higher likelihood of safety violations. In the rest of the section, we provide different upper bounds for $\phi_{(A,\Lambda)}(t)$ using three different methods, and finally provide an overview of these are used to rank the navigation choices for performing safety verification using reachable sets.

Bound I: Using p-Approximation. Given point matrices A and E, the closed-form upper bound for $\phi_{(A,E)}(t)$ can be computed using perturbation theory [24]. Specifically, the bound depends on a function $p_{n-1}(x)$, which is a truncated exponential series, and the 2-norm of E. It is given as follows:

$$\begin{aligned} \phi_{(A,E)}(t) \quad \leq \quad & p_{n-1}(||A||_2 t) \times \\ & \Big(\exp\left(p_{n-1}\left(\|A\|_2 t\right) \|E\|_2 t\right) - 1 \Big) \end{aligned} \tag{3}$$

where $p_{n-1}(x) = \Sigma_{k=0}^{n-1} \frac{x^k}{k!}$, and $\|E\|_2$ is the 2-norm of matrix E. This monotonicity is key to extending the result to interval matrices. Since each matrix E within the interval matrix Λ contributes differently based on its 2-norm, we can upper-bound the effect of all such matrices by simply computing the maximum 2-norm across the interval—denoted as $\|\Lambda\|_2$. Formally, this is given as: $\sup\limits_{E \in \Lambda} \{||\, E \,||_2\} = ||\, \Lambda \,||_2$

Furthermore, the above expression can be computed in polynomial time using the algorithm in [26]. Using this insight (*i.e.*, the monotonicity), we generalize the original bound to interval matrices by replacing $\|E\|_2$ with $\|\Lambda\|_2$ in the formula. The result gives us an upper bound on $\phi(A, \Lambda)(t)$, which quantifies the effect of uncertainty on the system dynamics when that uncertainty is described by an interval matrix:

$$\begin{aligned} \phi_{(A,\Lambda)}(t) \leq \; & p_{n-1}(\|A\|_2 t) \\ & \times \left(\exp\left(p_{n-1}(\|A\|_2 t)\|\Lambda\|_2 t\right) - 1\right) \end{aligned} \tag{4}$$

We denote this upper bound as **`Kagstrom1`** (following the work of Kågström [24] in this domain).

Bound II: Using Condition Number. Given a matrix A and a perturbation $E \in \mathbb{R}^{n \times n}$, Kaagström's result [24] provides the following upper bound:

$$\phi_{(A,E)}(t) \leq K(SD) \cdot e^{\epsilon t} \left(e^{K(SD) \cdot \|E\|_2 t} - 1 \right) \tag{5}$$

Here, $K(M)$ denotes the condition number of a matrix M, defined as $\|M\| \cdot \|M^{-1}\|$. The matrix A is decomposed into its Jordan form as SJS^{-1}, and D is a diagonal matrix such that $\|D^{-1}JD\|_2 \leq \epsilon$. As in the previous case, the function $\phi_{(A,E)}(t)$ increases monotonically with $\|E\|_2$. This observation allows us to extend the result from a point perturbation E to an interval matrix Λ, by taking the supremum of the 2-norm over all matrices in Λ. The resulting upper bound becomes:

$$\phi_{(A,\Lambda)}(t) \leq K(SD)\, e^{\epsilon t} \times \left(e^{K(SD) \cdot \|\Lambda\|_2 t} - 1 \right) \tag{6}$$

We denote this upper bound as **`Kagstrom2`**.

Bound III: Using Eigenvalues. Given a matrix A and a perturbation $E \in \mathbb{R}^{n \times n}$, the result from [25] provides the following closed-form upper bound:

$$\phi_{(A,E)}(t) \leq t\|E\|_2 \cdot e^{(|A|_2 - \alpha(A) + \|E\|_2)t} \tag{7}$$

where $\alpha(M)$ denotes the spectral abscissa of matrix M, *i.e.*, the largest real part among the eigenvalues of M. As in previous cases, we observe that the right-hand side is monotonically increasing with $\|E\|_2$. Using this observation, we extend the bound to interval matrices by replacing the norm of E with the supremum norm over the interval matrix Λ, which gives: $\phi_{(A,\Lambda)}(t) \leq t\|\Lambda\|_2, e^{(\|A\|_2 - \alpha(A) + \|\Lambda\|_2)t}$ (8)

We denote the above upper bound as **`Loan`**. We would like to note that both [24] and [25] offer several analytical formulas for computing upper bounds on $\phi_{(A,E)}(t)$. In this work, we focus on three specific formulas, as they demonstrated the strongest performance across standard verification benchmarks. The corresponding upper bounds—**`Kagstrom1`**, **`Kagstrom2`**, and **`Loan`**—represent our key contributions in extending perturbation-based techniques to interval matrices. These extensions enable principled reasoning about navigation choices in time-critical scenarios under uncertainty. We further note that the presented results can also be extended to use the Frobenius norm in place of the 2-norm. This is justified by the fact that, for any matrix M, the Frobenius norm $\|M\|_F$ is always greater than or equal to the 2-norm $\|M\|_2$. In our experiments, we use the 2-norm when working with low-dimensional systems, where its computation is tractable. However, for high-dimensional systems where computing the 2-norm becomes expensive, we instead use the Frobenius norm as a conservative and computationally efficient alternative.

Ranking Navigation Choices. In the scenario considered in this work, suppose the robot has K possible navigation choices. Each of these choices comes with its own bounded uncertainty, represented as interval matrices $\Lambda_1, \Lambda_2, \ldots, \Lambda_K$. For each navigation choice i, we compute a corresponding bloating factor $b_i = \phi(A, \Lambda_i)$, which quantifies the impact of the uncertainty Λ_i on the nominal reachable set. This is done using the tightest of the three available upper bounds—**Kagstrom1**, **Kagstrom2**, or **Loan**—whichever yields the lowest estimate for b_i. Once we have computed all the bloating factors $\{b_1, b_2, \ldots, b_K\}$, we sort the navigation choices in increasing order of these values. We then verify the safety of the navigation choices in this order using reachability analysis. As soon as we find a navigation choice whose reachable set does not intersect with the unsafe region, we terminate the verification process and select that choice for execution. This approach helps prioritize safer navigation options efficiently, while avoiding unnecessary computation.

Discussion. While we used the derived upper bounds—**Kagstrom1**, **Kagstrom2**, and **Loan**—to prioritize the order in which navigation choices are verified for safety, these bounds can also be used directly to compute the reachable set of an uncertain linear system, without relying on other reachability techniques. The approach is straightforward. First, we compute the nominal reachable set of the system $\dot{x} = Ax$ assuming no uncertainty. Then, we calculate an upper bound on the relative error due to bounded uncertainties, *i.e.*, $\phi_{(A,\Lambda)}(t)$, and use this to bloat the nominal reachable set accordingly. Our evaluation on both low- and high-dimensional benchmarks demonstrates that this method allows reachable sets to be computed in just a few seconds—often in under a second for most systems. Despite its efficiency, we currently do not use this method to compute reachable sets directly. Instead, we use it to rank the safety verification order of navigation choices. This decision is due to the technique's inherent limitation: it treats uncertainty in a structure-agnostic way by relying solely on matrix norm sensitivity. As a result, the reachable set is bloated uniformly in all directions, regardless of the actual system dynamics.

III. Experiments

We compute the bloating factors using **Kagstrom1**, **Kagstrom2**, and **Loan** on a set of linear dynamics benchmarks from the proceedings of ARCH workshop [27]. Experiments were performed on a Lenovo ThinkPad with i7-8750H CPU, 32 GiB memory on Ubuntu 20.04 LTS. For each benchmark, the initial set is chosen according to the benchmark specifications, and reachable sets are computed over a time horizon of 20 time units. When benchmarks include inherent parametric uncertainties, we use those directly; otherwise, we synthetically introduce perturbations by modifying randomly selected entries in the system matrix.

Benchmark	Dim	Pert	Kagstrom1	Kagstrom2	Loan
Holes	10	4	**0.004** s	0.4 s	**0.004** s
ACC	4	2	0.02 s	0.06 s	**0.014** s
Lane Change	7	3	0.007 s	0.29 s	**0.001** s
PK/PD	4	3	0.005 s	9.06 s	**0.0012** s
Motor	7	2	0.06 s	0.11 s	**0.008** s
Girard I	2	2	0.002 s	0.02 s	**0.001** s
Girard II	5	3	0.002 s	0.28 s	**0.0006** s
Space	6	4	0.002 s	0.2 s	**0.0006** s
Aircraft	4	2	**0.01** s	0.04 s	**0.01** s

TABLE I: **Timing for Upper Bound Computation.** Time (in seconds) taken by each method to compute the perturbation-based bloating factor for benchmarks from [27]. **Dim**: system dimension; **Pert**: number of perturbed cells.

Table I presents the runtime performance of the three perturbation-based upper bound techniques—**Kagstrom1**, **Kagstrom2**, and **Loan**—across a variety of linear system benchmarks drawn from the ARCH workshop suite [27]. For each benchmark, we report the system's dimensionality, the number of perturbed entries in the dynamics matrix, and the time taken by each method to compute the bloating factor $\phi_{(A,\Lambda)}(t)$. As the table shows, all three methods are extremely efficient, even for high-dimensional systems like Holes and Lane Change. This efficiency makes them particularly suitable for real-time decision-making on prioritizing navigation choices in uncertain environments. In addition to the results presented in Table I, we further evaluated the performance of the proposed methods (**Kagstrom1**, **Kagstrom2**, and **Loan**) on the ACC model [28]. While Table I focused on the 0–20 time range, we extended the analysis to 50–70 (with a step size of 0.1) to stress-test the methods. Our observations show that the upper bounds computed by **Kagstrom1** and **Kagstrom2** grew exponentially with time, whereas **Loan** exhibited almost linear growth. For instance, at time step 50, **Kagstrom1** and **Kagstrom2** yielded an upper bound of approximately $50\times$ the nominal value, while **Loan** produced bounds around $25\times$. At time step 70, the upper bounds of **Kagstrom1** and **Kagstrom2** increased to approximately $500\times$, while **Loan** remained comparatively moderate at around $40\times$. This shows that, in this particular case study (and not necessarily for other case studies), **Loan** produced significantly tighter upper bounds at higher time values.

IV. Concluding Remarks and Future Work

In safety-critical, time-sensitive scenarios, quickly selecting the safest navigation option is essential, yet traditional methods often rely on performing safety verification sequentially or arbitrarily, leading to delays. In this work, we proposed a highly efficient ranking method that prioritizes navigation choices based on their sensitivity to uncertainty, enabling faster decision-making. Our experiments demonstrate that the method is extremely fast, often completing in under 0.5 seconds even on large-scale benchmarks. As future work, we aim to extend this approach beyond prioritization to directly compute reachable sets using our bounds and further refine the technique to incorporate structure-aware reachability, addressing the current limitation of direction-agnostic bloating.

Acknowledgment: We acknowledge the NSF grant 2038960.

References

[1] B. Ghosh and P. S. Duggirala, "Robust reachable set: Accounting for uncertainties in linear dynamical systems," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, Oct. 2019.

[2] B. Ghosh, "Design and verification of autonomous systems in the presence of uncertainties," Ph.D. dissertation, University of North Carolina, Chapel Hill, USA, 2023. [Online]. Available: https://doi.org/10.17615/6qja-gn92

[3] R. Lal and P. Prabhakar, "Bounded error flowpipe computation of parameterized linear systems," in *International Conference on Embedded Software (EMSOFT)*, 2015.

[4] M. Althoff, C. Le Guernic, and B. H. Krogh, "Reachable set computation for uncertain time-varying linear systems," in *14th International Conference on Hybrid Systems: Computation and Control (HSCC)*, 2011.

[5] C. Combastel and S. Raka, "On computing envelopes for discrete-time linear systems with affine parametric uncertainties and bounded inputs," *IFAC Proceedings Volumes*, vol. 44, no. 1, 2011.

[6] X. Chen and S. Sankaranarayanan, "Decomposed reachability analysis for nonlinear systems," in *IEEE Real-Time Systems Symp. (RTSS)*, 2016.

[7] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *Computer Aided Verification (CAV)*, 2013.

[8] X. Chen, S. Sankaranarayanan, and E. Ábrahám, "Under-approximate flowpipes for non-linear continuous systems," in *Formal Methods in Computer-Aided Design (FMCAD)*, 2014.

[9] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok, "C2e2: A verification tool for stateflow models," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2015.

[10] A. Eggers, N. Ramdani, N. Nedialkov, and M. Fränzle, "Improving SAT modulo ODE for hybrid systems analysis by combining different enclosure methods," in *Software Engineering and Formal Methods (SEFM)*, G. Barthe, A. Pardo, and G. Schneider, Eds., 2011.

[11] S. Kong, S. Gao, W. Chen, and E. Clarke, "dReach: δ-reachability analysis for hybrid systems," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2015.

[12] R. Testylier and T. Dang, "NLTOOLBOX: A library for reachability computation of nonlinear dynamical systems," in *Automated Technology for Verification and Analysis (ATVA)*, 2013.

[13] M. Althoff, "An introduction to CORA 2015," in *1st and 2nd International Workshop on Applied veRification for Continuous and Hybrid Systems (ARCH'14 and 15)*, vol. 34, 2015, pp. 120–151.

[14] C. Hobbs *et al.*, "Safety analysis of embedded controllers under implementation platform timing uncertainties," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 41, no. 11, pp. 4016–4027, 2022.

[15] B. Ghosh *et al.*, "Statistical verification of autonomous system controllers under timing uncertainties," *Real Time Syst.*, vol. 60, no. 1, pp. 108–149, 2024.

[16] C. Hobbs *et al.*, "Quantitative safety-driven co-synthesis of cyber-physical system implementations," in *15th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2024.

[17] A. Yeolekar, S. Chakraborty, R. Venkatesh, and S. Chakraborty, "Repairing control safety violations via scheduler patch synthesis," in *ACM/IEEE 16th International Conference on Cyber-Physical Systems (ICCPS)*, 2025.

[18] S. Xu *et al.*, "Safety-aware flexible schedule synthesis for cyber-physical systems using weakly-hard constraints," in *28th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2023.

[19] A. Yeolekar *et al.*, "Checking scheduling-induced violations of control safety properties," in *20th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, ser. Lecture Notes in Computer Science. Springer, 2022.

[20] S. Xu *et al.*, "Statistical approach to efficient and deterministic schedule synthesis for cyber-physical systems," in *21st International Symposium on Automated Technology for Verification and Analysis (ATVA)*, ser. Lecture Notes in Computer Science, vol. 14215. Springer, 2023, pp. 312–333.

[21] ——, "Safety-aware implementation of control tasks via scheduling with period boosting and compressing," in *29th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2023.

[22] ——, "GPU partitioning & neural architecture sizing for safety-driven sensing in autonomous systems," in *International Conference on Assured Autonomy (ICAA)*, 2024.

[23] T. Zhu *et al.*, "Controllers for edge-cloud cyber-physical systems," in *17th International Conference on COMmunication Systems and NETworks (COMSNETS)*, 2025.

[24] B. Kågström, "Bounds and perturbation bounds for the matrix exponential," *BIT Numerical Mathematics*, vol. 17, no. 1, pp. 39–57, 1977.

[25] C. V. Loan, "The sensitivity of the matrix exponential," *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 971–981, 1977.

[26] R. Farhadsefat, J. Rohn, and T. Lotfi, "Norms of interval matrices," 2011. [Online]. Available: www3.cs.cas.cz/ics/reports/v1122-11.pdf

[27] "ARCH Workshop," https://cps-vo.org/group/ARCH.

[28] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1294–1307, 2016.