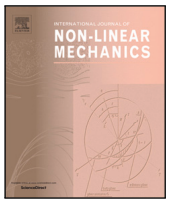
International Journal of Non–Linear Mechanics 185 (2026) 105337

Contents lists available at ScienceDirect

International Journal of Non-Linear Mechanics

journal homepage: www.elsevier.com/locate/nlm



An efficient algorithm for estimating dynamical integrity in time-delay systems

Bence Szaksz *, Giuseppe Habib

Department of Applied Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Muegyetem rkp. 3, Budapest, H-1111, Hungary
MTA-BME Lendulet "Momentum" Global Dynamics Research Group, Budapest University of Technology and Economics, Muegyetem rkp. 3, Budapest, H-1111, Hungary

ARTICLE INFO

Keywords:
Dynamical integrity
Basin of attraction
Local integrity measure
Time delay
Limit cycle oscillation
DynIn

ABSTRACT

In nonlinear systems, the dynamical integrity of a steady state solution characterizes its robustness against external perturbations. Determining the dynamical integrity is a challenging task even for moderately sized systems, and it becomes even more complex in time-delay systems due to their infinite-dimensional state-space representation. Building upon previous studies, the current paper targets this problem and introduces an efficient iterative algorithm for estimating the local integrity measure of both fixed points and limit cycles in time-delay systems. The proposed method is computationally fast, enabling parametric analysis and making it a powerful tool for identifying safety-critical parameter regions, where even small disturbances can cause the system to escape the desired steady state. The effectiveness of the algorithm is demonstrated through two case studies: a delayed van der Pol–Duffing oscillator and an inverted pendulum equipped with a tuned vibration absorber.

1. Introduction

A peculiar problem of nonlinear dynamical systems is that multiple stable solutions can coexist, a phenomenon with no counterpart in linear systems, where at most one attractor exists. This issue has significant practical consequences. For a linear system, the stability of a steady state solution implies that this state is *globally* attractive, that is, if it is subject to an arbitrary finite perturbation, the system dynamics will return to its original state. Conversely, in nonlinear systems, the stability of a steady state only entails that the system is *locally* attractive, meaning that the system dynamics will converge back to its original state only if the perturbation is sufficiently small. However, the threshold amplitude beyond which the system may transition to a different attractor is generally unknown. In practice, a physical nonlinear system, although correctly functioning in a stable condition, if subject to an excessively large perturbation related to transient surrounding conditions, might converge towards other undesired regimes [1].

This problem is encountered across very diverse disciplines. For example, in traffic flow, a single driver temporarily slowing down can cause a traffic jam on an otherwise free-flowing highway [2–6]. In machining, microscopic material heterogeneities may trigger vigorous chatter vibrations [7–9]. Epilepsy can result from a perturbation that drives the dynamical system of the brain towards pathological synchronization and self-sustaining oscillatory activity of neuronal populations [10,11]. Similarly, power grid blackouts can occur when faults push generators out of synchrony [12], a problem that becomes particularly relevant with the extensive integration of distributed renewable energy sources [13,14]. The phenomenon is relevant for numerous other applications, such as wheel shimmy [15–17], valve dynamics [18,19], robot control [20–22], aeroelastic flutter of wings [23–25], energy harvesters [26–28], and epidemic models [29,30], among others.

Arguably, the most comprehensive tool for studying the robustness of dynamical systems against external perturbations – a property termed dynamical integrity (DI) – is the basin of attraction (BoA). The BoA of a stable solution is the set of initial conditions in the state-space that asymptotically converge to that solution. However, computing BoAs is challenging. Analytical methods, mainly based on Lyapunov functions, are typically limited to specific nonlinear forms of the equations of motion [31], and there are no general algorithms to find them [32,33]. Conversely, numerical approaches suffer from high computational cost and memory issues, which are only mitigated, for example, by the cell-mapping method [34,35] and its variants [36,37].

Another problem of BoAs is the interpretation of the information they provide in a practical way, an issue solved through the introduction of DI measures, which are scalar quantities defining the DI of a

* Corresponding author at: Department of Applied Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Muegyetem rkp. 3, Budapest, H-1111, Hungary.
E-mail address: szaksz@mm.bme.hu (B. Szaksz).

https://doi.org/10.1016/j.ijnonlinmec.2026.105337
Received 13 November 2025; Received in revised form 15 January 2026; Accepted 6 February 2026
Available online 9 February 2026

steady state solution [38–41]. These are particularly useful for studying basin erosion [40,42]. For example, the global integrity measure [39], later renamed basin stability [12], is the ratio of a BoA's volume to that of a reference state-space region, while the integrity factor [43], which aims to exclude fractal portions of the BoA, is the radius of the largest hypersphere entirely contained in the BoA. Another practical DI measure, relevant for this study, is the local integrity measure (LIM), defined as the minimal distance between a steady state solution and any point external to its BoA.

In [44], we introduced a fast method for directly estimating the LIM of a solution without requiring to compute its BoA. The method, implemented in the `MATLAB` toolbox `DynIn`, exploits an iterative estimation of the LIM through the subsequent generation of system's trajectories in the state-space, enabling it to rapidly identify nearest escape routes without full BoA reconstruction. The algorithm is particularly rapid thanks to a subdivision of the state-space in cells, which enables fast categorization of the trajectories and identification of other solutions, significantly reducing computational time. Additionally, since the algorithm does not require to explore the entire state-space, it does not present memory issues, which instead affect the cell mapping method [37]. The method was recently enhanced for the analysis of periodic solutions of autonomous and non-autonomous systems in [45]. The main objective of this study is to extend it also to periodic solutions of time-delay systems.

Time delay is relevant for a wide variety of dynamical systems and phenomena. Examples include traveling wave phenomena [19], regenerative effects [7,46,47], feedback control [21,22,48,49], and other phenomena where the current behavior of the system is influenced by its history [50–54]. The corresponding dynamics is governed by delay differential equations (DDEs), which have an infinite-dimensional state-space representation [55,56]. This means that the initial condition is a function of time, and not a point in a finite-dimensional state-space. Consequently, the classical definition of BoA is not directly applicable to time-delay systems.

Some studies in the literature attempt to define BoA of time-delay systems, for example based on Lyapunov-Krasovskii functionals [57]. However, the derivation of a proper Lyapunov-Krasovskii functional for a generic nonlinear dynamical system is challenging. Thus, instead of this analytical approach, often the initial function is restricted to a specific type (constant/linear/jump type) [58,59], reducing initial conditions to single points in a finite state-space.

Recently, the LIM estimation algorithm [44] was generalized to the case of DDEs [60], where the BoA, and so the LIM, were defined based on the headpoint of a restricted type of initial conditions. The method proved very efficient, enabling extensive parametric analysis of a system's DI. However, it was limited to the analysis of equilibrium points, and it could not deal with periodic solutions. This limitation was strictly related to the definition of the BoA based on the headpoints of the initial functions, which prevented the analysis of the full trajectory for the LIM estimation. In practice, a similar framework makes the analysis of periodic solutions extremely long and impractical.

In this study, we overcome this limitation by discretizing time to obtain a finite-dimensional representation, allowing BoAs (and LIMs) of both equilibria and limit cycles to be defined and estimated as in non-delayed systems. Accordingly, whole trajectories are used for the LIM estimation, and not only their headpoints, greatly accelerating computations for periodic orbits. The algorithm is then validated on two case studies – a van der Pol-Duffing oscillator and an inverted pendulum with an attached nonlinear tuned vibration absorber subjected to delayed control – which prove its effectiveness for the parametric analysis of a system's LIM.

The rest of the paper is organized as follows. In Section 2, the proposed algorithm is described in details. Section 3 presents the two case studies. In Section 4, pros and cons of the algorithm are discussed. Finally, Section 5 contains concluding remarks and future developments.

2. Algorithm

2.1. Overview

The proposed algorithm, building upon the developments presented in [44,45,60], operates by simulating multiple trajectories of the time-delay system under study. For each trajectory that does not converge to the desired solution, the LIM is computed as the minimum distance between the desired solution and the non-converging trajectory. This approach provides an overestimate of the LIM, which is iteratively reduced as additional non-converging trajectories are identified.

Since time-delay systems have an infinite-dimensional state-space representation, time discretization is used to limit its dimension to a finite number. We note that this time-discretization does not affect the accuracy of the trajectory, as discussed below.

The trajectories are categorized as either converging or non-converging, depending on their asymptotic behavior with respect to the investigated steady state solution. In order to speed up the algorithm, each time integration is interrupted when the related trajectory gets close enough to an already categorized one, assigning to the former the same convergence classification of the latter. The iterative procedure is continued until a stopping criterion is met; in this paper, a prescribed iteration number is imposed.

As discussed in the introduction, the algorithm is applicable to fixed points and periodic orbit types of solutions. The following subsections provide further details on the key components of the algorithm.

2.2. Trajectory simulation

Consider a dynamical system in the form of the first order DDE

$$\dot{\mathbf{x}}(t) = \mathbf{A}_0\mathbf{x}(t) + \mathbf{A}_\tau\mathbf{x}(t-\tau) + \mathbf{g}(\mathbf{x}(t), \mathbf{x}(t-\tau)), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is an n dimensional vector of state variables, $\mathbf{A}_0 \in \mathbb{R}^{n\times n}$ and $\mathbf{A}_\tau \in \mathbb{R}^{n\times n}$ are the coefficient matrices of the non-delayed and delayed states, respectively, $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is the vector of nonlinear terms and $\tau \in \mathbb{R}$ refers to the time delay in the system.

At every time instance, the state of a trajectory is a function of time in the interval $[t-\tau, t]$, so the system has an infinite dimensional state-space representation.

Since nonlinear delay differential equations do not have closed-form algebraic solutions, a numerical integration scheme and an appropriate time discretization are required to obtain the corresponding trajectories (see Fig. 1). The time integration of the solution is performed using the fourth-order Runge–Kutta (RK4) method, which ensures accuracy if the time steps are kept small enough.

2.3. State discretization

The trajectory simulation is carried out with tiny time steps, meaning that the theoretically infinite-dimensional state of a solution is still represented as a high-dimensional vector. Accurately estimating a solution's DI in this large-dimensional state-space is practically infeasible; therefore, a less dense time-discretization is required for the LIM calculation and trajectory comparison (see Fig. 1).

Let this coarse mesh consist of N equally distributed sampling points within a time interval of length τ, resulting in a time step of $\tau/(N-1)$. This implies that the discretized state for comparison is

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}\left(t - \frac{\tau}{N-1}\right) \\ \mathbf{x}\left(t - \frac{2\tau}{N-1}\right) \\ \vdots \\ \mathbf{x}\left(t - \frac{(N-2)\tau}{N-1}\right) \\ \mathbf{x}(t-\tau) \end{bmatrix}. \tag{2}$$

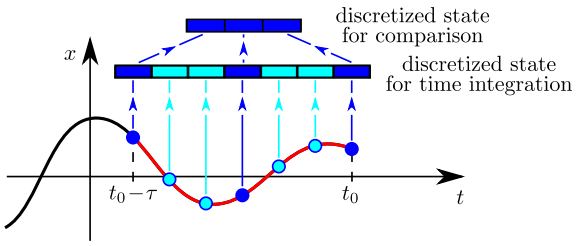


Fig. 1. Visualization of the two discretization levels for $N = 3$, $k = 3$. The red curve refers to the exact state of the system. The fine mesh, including both the light and dark blue cells, refers to the discretized state for time integration, while the coarse mesh of the dark blue cells is the discretized state for trajectory comparison. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Note that it contains the state of the system at both ends of the τ long time interval.

In this discretized state-space, the distance between two generic points A and B is defined as

$$d_{\mathrm{AB}} = \sqrt{\sum_{i=1}^{nN} \alpha_i (y_{i\mathrm{A}} - y_{i\mathrm{B}})^2}\,, \tag{3}$$

where $y_{i\mathrm{A}}$ and $y_{i\mathrm{B}}$ are the ith coordinates of the two points and α_i is the corresponding weight. The weights can be selected based on the relevance of the different coordinates of the engineering system, as discussed in [44,60].

For the sake of simplicity, the mesh for the discretized state has k-times larger time steps than the fine temporal discretization for time integration, that is, the state for time integration consist of $k(N-1)+1$ sampling points, with k and N integer numbers (see Fig. 1).

Fig. 2 presents the state discretization for $N = 3$. As it can be seen, while in the original system the initial condition is a function of time, it becomes a point in the discretized $nN = 6$ dimensional space ($n = 2$ in this example).

2.4. BoA and LIM definition

Let us first define the BoA of a steady state solution in the infinite-dimensional state-space of the original DDE, then reformulate it to the discretized problem.

A state in the state-space belongs to the basin of attraction of a fixed point (limit cycle), if a trajectory, initiated from that state, converges to the fixed point (limit cycle). This definition, although mathematically rigorous, has two practical problems. First, according to it, the BoA is infinite-dimensional as the state-space, leading to obvious computational challenges. Additionally, large regions of the BoA correspond to physically unrealistic states of the system (for example, a non-smooth function that has jumps almost everywhere).

The first issue is solved by the time discretization, which reduces the state-space to a finite dimension. In the discrete case, a finite-dimensional vector represents the state of the system at a given time instance (see Fig. 1, and 2). The BoA is defined in this finite state-space as for a non-delay system, since it is assumed that each point identifies a specific unique state. Although this is not true, as each point of the discrete state-space corresponds to infinitely many physical states, this simplification did not lead to problems during the testing of the algorithm because those physical states have similar convergence properties in the vast majority of cases.

Despite its finite dimension, large part of the discrete space state mostly corresponds to physically irrelevant states, as it was the case for the original state-space as well. This problem is solved by constraining the type of possible initial conditions [60]. Here, we propose to use jump-type initial conditions, where the system is in equilibrium in the interval $t \in [-\tau, 0)$, either on a fixed point or along a periodic solution, while at $t = 0$, an impulse-like perturbation occurs and the state jumps to a certain value $\mathbf{x}_0$. This type of initial condition leads to a special type of integro-differential equations [61]. Notably, the algorithm also works with different types of constrained initial conditions.

We note that in the algorithm proposed in [60] (where the state-space was not discretized), the BoA was only defined based on the headpoints of the initial conditions. Although that algorithm works properly, this restriction significantly slows down the procedure, especially in the case of limit cycles, because for each trajectory only a single point can be used for the LIM estimation. Conversely, in the discretized state-space, all points of a trajectory contribute to define the BoA and estimate the LIM. This aspect is crucial for rapidly assessing the DI of a periodic solution, as discussed in Section 2.7.

In this paper, we describe the DI of a steady state solution with the LIM, which is defined as the minimal distance between a solution and its BoA boundary. Clearly, different BoA definitions lead to slightly different LIM values. However, usually, the qualitative trend of the LIM for parameter variations is almost unaffected by changes in the definition of the BoA.

In Fig. 2a, the light green domain marks the hypersphere of convergence of a fixed point, i.e. the hypersphere defining the LIM. In Fig. 2b, the light green area defines the hypertorus of convergence of the limit cycle, i.e. the hypertorus generated by translating the hypersphere of convergence along the periodic orbit. As it will be discussed later, the algorithm gives an upper estimate of the LIM based on the minimal distance between the investigated solution – either a fixed point or a limit cycle – and the non-converging trajectories. Note that, in Fig. 2a, the initial condition is not of jump type to better express its infinite-dimensional nature.

2.5. Trajectory categorization

In this section, we present the trajectory categorization applied in the proposed algorithm. We apply the same concept as utilized in the earlier methods of the `DynIn` toolbox [44,45,60]; however, some slight modifications are required to address the current case of time-delay systems. First, we focus on the DI of stable fixed points, which will be later generalized to the case of stable limit cycles.

The algorithm requires to set state-space boundaries, within which the LIM is estimated. This can be related to physically relevant safe configurations of the engineering system, outside of which the system may be damaged. Alternatively, it may refer to the boundaries beyond which the implemented equations of motion become inaccurate.

We also apply a spatial discretization, that is, the discrete state-space is divided into a finite number of (multidimensional) cells. Note that the initial condition of the trajectories and their time integration are independent of the cell structure.

Each numerical integration is continued until the related trajectory converges to a steady state solution, leaves the pre-described space boundary, or the number of integration steps reaches a maximal value. Then, the trajectory is categorized as either converging or non-converging, and all the cells containing points of the trajectory are also classified according to the trajectory's convergence property. During the integration of each trajectory, it is continuously monitored whether it enters a cell that was previously classified. If it occurs, the numerical integration stops and the new trajectory is classified according to the category of the entered cell.

When searching for the LIM of a desired fixed point, four types of trajectories are distinguished:

- *Converging to the desired fixed point.* The cell containing the desired fixed point is automatically classified as convergent (dark green cells in Fig. 3). If a new trajectory enters this cell, the integration is stopped and the trajectory is classified as convergent (see trajectories #1 and #2 in Fig. 3a and Fig. 3b). If the system is

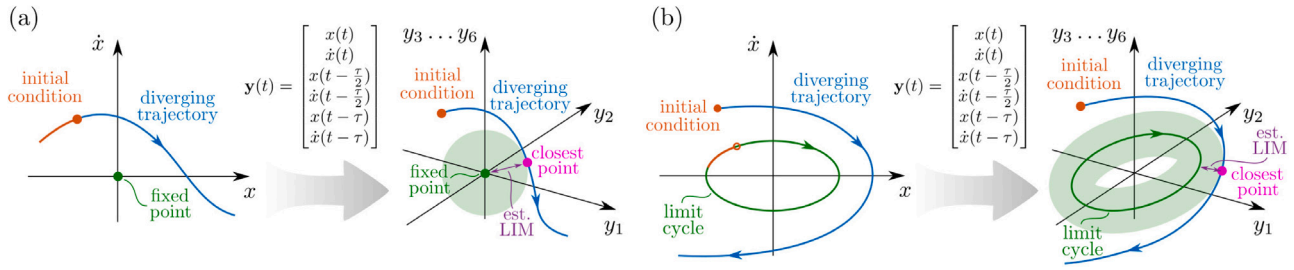


Fig. 2. Estimation of the LIM in the discretized space for $N = 3$ and $n = 2$. Panels (a) and (b) refer to the LIM of a fixed point and of a limit cycle, respectively. In each panel, the left figure depicts the steady state solution (dark green), a non-converging trajectory (blue curves), and its initial condition (orange curves) in the space of the original coordinates, where the state of the system is a function of time. The right figures present the same objects in the discretized state-space, in which the LIM calculation and the trajectory comparison are carried out. The state of the system is reduced to a single point in this space; thus, fixed points are represented as distinct points, while periodic orbits are reduced to a set of time-independent points. The smallest distance between the steady state solution and the diverging trajectory provides the current (upper) estimation of the LIM. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

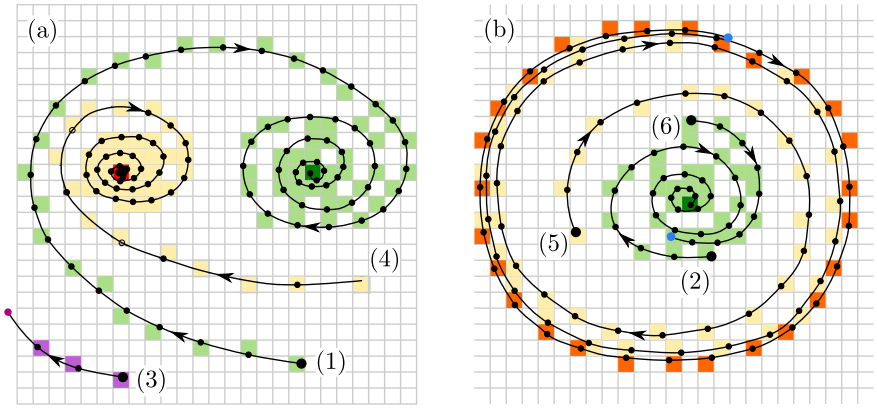


Fig. 3. Trajectory categorization using the supplementary cell structure of the discretized state-space (each cell has nN dimensions, cf. Fig. 2). The cells converging to the desired equilibrium are presented with green colors, the cells leading to an undesired steady state are yellow, while those cells which yield trajectories exiting the predefined space-boundary are purple. The dark green cells contain the desired fixed point. The red cells in panel (b) refer to a limit cycle identified by the algorithm. Blue dots indicate the termination of time integration due to entry into a previously categorized cell (trajectory #6) or multiple re-entry into the same cell, indicating a limit cycle (trajectory #5). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

slightly damped, the trajectory may oscillate around the fixed point for a long time without entering the cell. To solve this problem, the user may choose to classify not only the cell of the fixed point but also the neighboring cells as convergent.

- If the trajectory *exits the predefined space boundary*, as trajectory #3 in Fig. 3a, then, it is classified as divergent. Note that if the time integration were continued, the trajectory might re-enter the state-space of interest. However, considering the trajectory as undesirable is a relevant decision from an engineering point of view if the boundary also corresponds to a physical constraint of the system, as discussed above.
- *Converging to another fixed point.* If a trajectory remains in the same unclassified cell for several time steps, the algorithm assumes that it reached a fixed point, previously unknown. Consequently, the trajectory is classified as "non-convergent" from the point of view of the desired fixed point. See trajectory #4 in Fig. 3a.
- The fourth option is the case of a *stable limit cycle*. If a trajectory passes through the same cell several non-consecutive times, i.e., the trajectory also passes through other cells, then the integration is stopped and the solution is classified as converging to a stable limit cycle. A schematic representation of this case is provided by trajectory #5 in Fig. 3b. Here, the integration is interrupted when the trajectory enters the same cell for the third time (indicated by the blue dot). Of course, a slowly decaying oscillatory trajectory may be misclassified as converging to a limit cycle. In this case, the required number of cell re-entries could be increased. However, this increases the length of the trajectory and ultimately represents a trade-off between accuracy and computational time. Note that this issue is also present in similar algorithms, such as the cell-mapping method [62].

In Fig. 3b, trajectory #6 converges to a cell previously tracked by #2. In such cases, we assume that the two solutions will converge to the same state; thus, the new trajectory receives the same classification as the previously categorized one. This significantly speeds up the algorithm, since, as the iteration number increases, more and more cells are already classified, and the average length of the new trajectories decreases. In addition, even trajectories with very different initial conditions tend to get closer to each other, while organizing along invariant manifolds [63,64]. This increases the probability of reaching already tracked cells.

Finally, if a trajectory is not categorized in the prescribed number of steps, it is considered as non-convergent. This is a conservative decision, which also reflects the engineering perspective that excessively long transients should be avoided. Besides, this approach enables the classification of trajectories that reach quasi-periodic or chaotic attractors as non-convergent. Note that these types of solutions are otherwise difficult to be classified automatically. In addition, the algorithm gives the possibility to remove this automatic classification, allowing the user to manually define the convergence properties of otherwise unclassified trajectories.

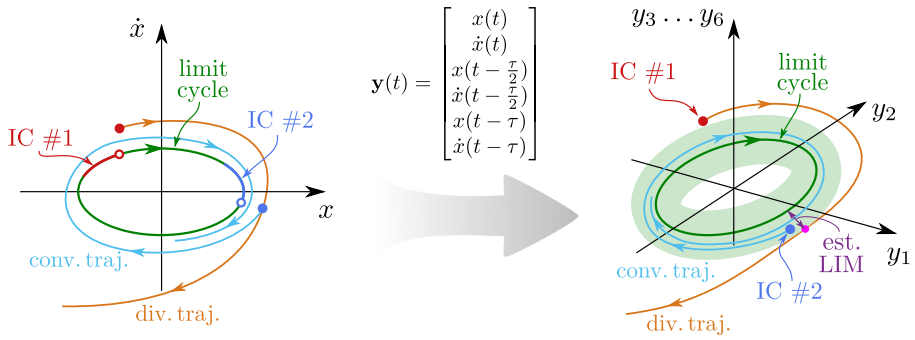


Fig. 4. Calculation of the LIM for limit cycles. The smallest distance between the divergent trajectory and the limit cycle yields the estimated LIM. Then, the next initial condition is constructed in a way, that its headpoint is determined by the closest point od the divergent trajectory, while the history of the initial condition lies on the limit cycle.

It should be mentioned that, because of the infinite dimensional nature of time-delay systems, the trajectories may cross each other in the discretized state-space, potentially leading to wrong results. Increasing the dimension of the discretized space reduces the probability of intersections. However, this also slows down the algorithm since it reduces the probability of convergence of a trajectory to already investigated cells. Therefore, the optimal dimension of the discretized space is a compromise between accuracy and computational time. In some sense, the choice of the discretized time step is analogous to the sampling time for frequency analysis as governed by the Nyquist-Shannon theorem.

2.6. Selection of new initial conditions

To obtain an efficient iteration, the appropriate selection of initial conditions is a critical aspect. The algorithm aims to identify non-converging trajectories, which lie as close as possible to the BoA of the desired solution, or, better, to the true hypersphere of convergence, because they provide the best LIM estimation. This is obtained by estimating the closest state-space region to the desired solution, which is external to its BoA, and by initiating trajectories from the vicinity of the BoA's boundary, as these trajectories tend to move along the boundary for a while before leaving it, thus providing significant information about the steady state's DI. This is obtained through a sort of bisection method, as detailed below.

As mentioned above, jump-type initial conditions are considered, that is, the history of the initial condition is restricted to the desired solution (for $t \in [-\tau, 0)$), while the headpoint at $t = 0$ is located freely in the n-dimensional space of the state variables.

If the desired solution is a fixed point, then the histories of the initial conditions are always the same and the headpoint uniquely determines the new trajectory. In contrast, in the case of limit cycles, the initial condition history can also be shifted along the limit cycle yielding an additional free parameter during the selection of the new initial condition. In this subsection, we focus on the fixed point case (see Section 2.7 for the case of limit cycles).

The algorithm first calculates the smallest distance between the desired solution and any of the space boundaries. This will serve as a first (upper) estimate of the LIM and the corresponding estimated hypersphere of convergence. Then, the first few initial condition headpoints are selected randomly until a non-convergent trajectory is found. If that is the case, the algorithm looks for that point of the non-convergent trajectory, which is closest to the desired fixed point; the corresponding distance will be the next estimate of the LIM. Note that this distance calculation is done in the discretized state-space.

Then, the closest state's headpoint is extracted, and it is utilized as the next initial condition headpoint, while the corresponding history is reset to the desired solution. Since the history is changed, the new trajectory is not the same as the previous one; however, they can be expected to behave similarly, so the new trajectory is a good candidate for a non-converging trajectory. If this is the case, the LIM is reduced further, because setting the initial condition's history to the fixed point moved the corresponding discretized state inside the previously estimated hypersphere of convergence.

Then, a bisection method is applied between the initial condition headpoint and the fixed point, in order to start a trajectory from the vicinity of the BoA boundary. After a fixed number of iteration steps, we select again random initial conditions to explore other regions of the state-space. In the meantime, it is continuously monitored where the closest point of the trajectory is located. If it is not at the initial condition, the new initial condition will be determined by that.

Random initial conditions are chosen in the vicinity of the hypersphere of convergence, where there is more probability of having initial conditions leading to non-converging trajectories, but still close to the BoA boundary.

2.7. Estimation of the LIM of limit cycles

The LIM of limit cycles in time delay-systems is estimated in a similar manner as in the fixed point case. The algorithm automatically identifies the limit cycle starting from a user-defined initial condition, exploiting the same procedure discussed in [45], i.e., through a Poincaré map and measuring the distance of subsequent crossing points. The exact limit cycle can also be directly provided by the user.

In the discretized state-space, the limit cycle is then reduced to a finite number of points, each of them belonging to a specific cell. Note that several points may belong to the same cell. During preprocessing, the algorithm classifies these cells as convergent. The iteration then begins: a new trajectory is considered convergent if it converges to any of the already classified convergent cells; otherwise, it is classified as non-convergent, as for the fixed point case.

The LIM is defined as the minimal distance between the limit cycle and any boundary of its BoA (in the discretized space), which is approximated by the minimal distance between the limit cycle and non-converging trajectories. Also in this case, jump-type initial conditions are used, where the history of the initial function lies on the limit cycle, while the initial condition's headpoint is located in another point in the state-space. This implies that the initial point in the discretized space is not on the limit cycle (see Fig. 2b). In order to uniformly explore the BoA, the iterative procedure is carried out for various initial conditions shifted along the limit cycle.

The selection of the new initial conditions is similar to the fixed points case. First, a random point of the limit cycle is selected; then, keeping the corresponding history, the headpoint is selected randomly from the n-dimensional state-space (see IC #1 in Fig. 4). If the corresponding trajectory is convergent, then new random headpoints are selected until a divergent trajectory is found. Then, the algorithm looks for the smallest distance between the trajectory and the limit cycle (see the pink point in the right panel of Fig. 4). This provides the new

Table 1
Parameters of the algorithm used in the case studies. The two-level trajectory discretization is determined by the state-space discretization number N, and the time step ratio k (cf. Eq. (2) and Fig. 1). Furthermore, α denotes the vector of weights (cf. (3)). During preprocessing, one must also specify the number of iteration steps, the predefined space boundary, the spatial discretization number used for the cell structure, and the maximum allowed elapsed time, after which the time integration stops and the simulation is considered non-convergent. The expressions 1D and 2D refer to the one- and two-dimensional parameter analyses, respectively. Namely, the analysis of the inverted pendulum problem, when only the proportional gain p was varied and when the parameter sweeping was carried out in the plane of the proportional and derivative gains p and d.

		N	k	α	number of steps	spaceboundary	discr. number	t_{final}
van der Pol - Duffing	fixed point	3	20	[100,1]	100	[-10,−100,10,100]	501	5000
	limit cycle	3	100	[100,1]	100	[-20,−300,20,300]	501	5000
inverted pendulum	fixed point 1D	3	10	[1,1,1,1]	400	[-50,−50,50,50]	501	1000
	fixed point 2D	3	10	[1,1,1,1]	100	[-50,−50,50,50]	501	1000
	limit cycle 1D	3	20	[1,1,1,1]	400	[-50,−50,50,50]	501	1000
	limit cycle 2D	3	20	[1,1,1,1]	100	[-50,−50,50,50]	501	1000

estimate of the LIM and also forms the basis for the selection of the next initial condition.

If it is the first point of the trajectory, then a bisection method starts between the limit cycle and the original initial condition. However, if the closest point is somewhere else, then, that will determine the headpoint of the new initial condition, while the tail of the initial condition is projected to the limit cycle in the following way: The closest point of the limit cycle to the non-converging trajectory is identified; from this point, the algorithm reconstructs the history of the initial condition along the limit cycle in the original state-space. The corresponding new initial condition IC #2 is presented in blue in Fig. 4. This will be the next candidate for a new bisection method. Note that this new initial condition (blue) is different from the closest point (pink) of the divergent trajectory.

If the number of iteration steps in a given bisection method reaches a prescribed limit, a new initial condition is selected randomly: both its phase along the limit cycle and the corresponding headpoint are randomly chosen.

This procedure yields a rapid and reasonably accurate estimate of the hypertorus of convergence around the stable limit cycle.

2.8. Practical considerations for parameter tuning

After introducing the algorithm, let us briefly discuss the suggested range of the main parameters and the corresponding tuning procedure.

In general, the state-space discretization number N should be small, between 3 and 5, otherwise the nN dimensional space for trajectory comparison and the corresponding cell structure becomes too large slowing down the algorithm. Larger N values help avoiding misclassifications of trajectories; however, in the case of parametric studies, such misclassifications are easily identified during post-processing as an anomaly of the LIM trend. The time step ratio k plays a role similar to the tolerance of the `ode45` solver of `MATLAB`, i.e., it defines the accuracy of the time integration. In the case studies considered below, it was selected between 10 and 100, and this range proved effective for other cases not discussed here. However, in the case of large time delays with respect to the system dynamics, or for very stiff problems, larger k values might be appropriate. Finally, the spatial partition density of the cells influences the trajectory comparison, the suggested range of the corresponding discretization number is between 50 (for rapid rough estimations) to 1000. A larger number of cells increases the accuracy of the estimation at the price of a higher computational cost. We note that, in the case of systems slowly converging to a stable focus, transient trajectories might be erroneously interpreted as periodic orbits; this problem is mitigated by increasing the number of cells, or adjusting the parameters for periodic solution identification, as better detailed in [44]. For further details we refer to the discussion in Section 4.

In the following, we provide practical indications for tuning the parameters in a typical scenario. After setting an initial value of the parameters (even adopting the default parameter values in the toolbox), it is worth running a few time simulations, and look for the smallest acceptable value of k for which the trajectories are accurate enough. Then, the algorithm can be applied to perform a LIM estimation for the selected stable steady state solution. During the first runs, it is suggested to plot the obtained trajectories at each iteration step (trajectory plotting is usually opted out for reducing computational time). This helps to check whether there is an issue with the settings. If the trajectory convergence seems to be wrong, that is, the algorithm cannot find a fixed point or limit cycle, while it is expected to exist, then first it is suggested to increase the spatial partition density. This helps in most of the time. Additionally, if the dynamics of the system is fast with respect to the time delay, the trajectories might cross each other in the discretized space. This issue is solved by increasing the value of N.

3. Case studies

The algorithm was successfully tested on various models. Here we discuss the analysis of a 1 degree of freedom (DoF) and a 2 DoF dynamical system; namely, the delayed van der Pol-Duffing oscillator and the case of an inverted pendulum with an attached nonlinear tuned vibration absorber subjected to delayed proportional-derivative (PD) control.

3.1. Van der Pol-Duffing oscillator

First, let us consider the van der Pol-Duffing oscillator subjected to derivative control with constant time delay τ:

$$\ddot{x}(t) + \mu(x^2(t) - 1)\dot{x}(t) + x(t) + \alpha x^3(t) = -d\,\dot{x}(t-\tau)\,, \tag{4}$$

where d denotes the derivative gain, while μ and α are the nonlinear damping and nonlinear stiffness coefficients, respectively. Note that for the sake of simplicity, the parameters and variables are considered dimensionless.

First, we study the stability of the trivial solution. The characteristic equation corresponding to the linearized dynamics assumes the form

$$\lambda^2 - \mu\lambda + 1 + d\,\lambda \mathrm{e}^{-\lambda\tau} = 0\,. \tag{5}$$

According to the D-subdivision method [56], the static (saddle–node bifurcation) and the dynamic (Hopf bifurcation) stability boundaries of the system can be obtained by substituting $\lambda = 0$ and $\lambda = \mathrm{i}\omega$ with $\omega > 0$ into the characteristic Eq. (5). The former one is never satisfied, thus, no saddle–node bifurcations occur in this system. The latter one leads to the complex equation

$$-\omega^2 - \mathrm{i}\omega\mu + 1 + \mathrm{i}\omega d(\cos(\omega\tau) - \mathrm{i}\sin(\omega\tau)) = 0\,. \tag{6}$$

Separating the real and imaginary parts results

$$-\omega^2 + 1 + \omega d\,\sin(\omega\tau) = 0\,, \tag{7}$$

$$-\omega\mu + \omega d\,\cos(\omega\tau) = 0\,, \tag{8}$$

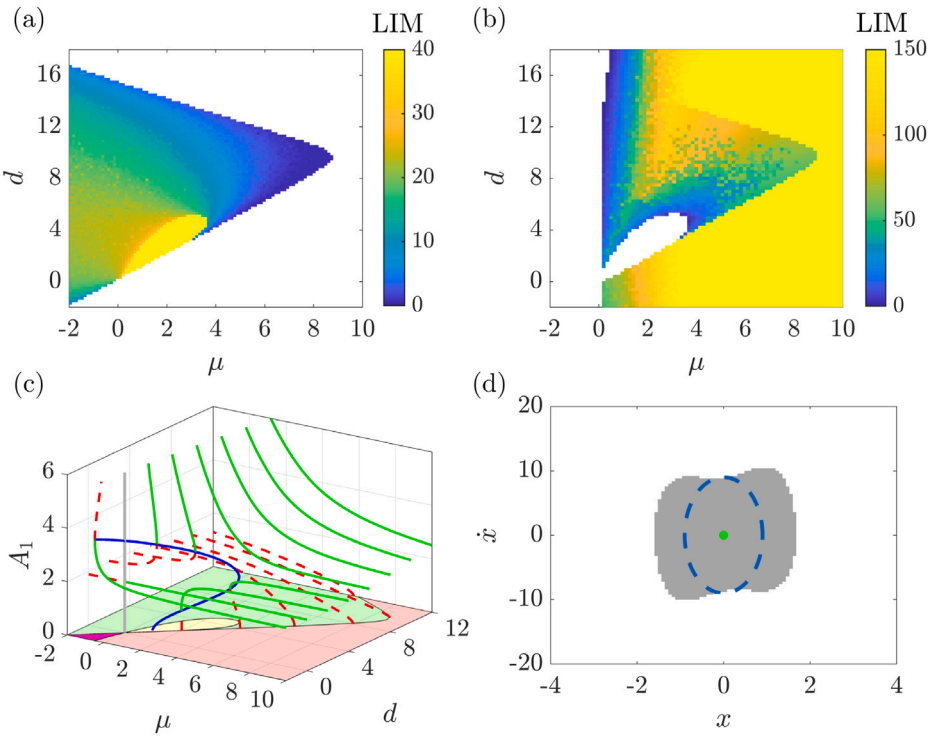


Fig. 5. Nonlinear analysis of the van der Pol-Duffing oscillator with delayed derivative control. In panel (a), the linearly stable domain of the (μ, d) plane is colored according to the LIM of the zero equilibrium. Panel (b) is colored according to the LIM of the stable limit cycle in the region of its existence. Panel (c) depicts the 3-dimensional bifurcation diagram obtained with the continuation-based MATLAB package DDE-BIFTOOL. In the green and yellow regions, the fixed point is locally and globally stable, respectively. In the red domain, the fixed point is unstable, while the magenta region corresponds to parameter combinations with no stable attractor. The solid green and dashed red lines represent unstable and stable limit cycles, respectively; finally, the blue curve denotes the fold of the limit cycles. Panel (d) shows the result of a brute-force basin of attraction estimation. The blue dashed circle is the 2-dimensional section of the hypersphere of convergence estimated with the proposed algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which yields the Hopf bifurcation boundary in the (μ, d) plane parametrized with ω

$$d = \frac{\omega^2 - 1}{\omega \sin(\omega\tau)}, \tag{9}$$

$$\mu = \frac{\omega^2 - 1}{\omega} \cot(\omega\tau). \tag{10}$$

These generate the stability chart reproduced in Fig. 5a for $\tau = 0.1$ and $\alpha = 60$ (the colored area marks the stable region). Neglecting the time delay, the damping coefficient of the system sums up to $(d - \mu)$ (cf. Eq. (4)). This explains that for small values of ω, the stability boundary is approximately the $d = \mu$ identity line and the system is stable above that. Then, for larger values of ω, the effect of time delay becomes relevant and therefore there is an upper limit for the value of the differential gain d. This results in the triangular-shaped stable domain.

The stable domain is colored according to the LIM of the trivial solution. For the LIM estimation, we set the algorithm parameters to those presented in Table 1. In most cases, the LIM is small close to the stability boundary, and it increases as the parameters are shifted towards the center of the stable domain. The bifurcations occurring at the corresponding stability boundaries are subcritical, that is, unstable limit cycles emerge from the boundary curve, which limit the BoA of the stable equilibrium (see also Fig. 5c). However, we note a portion of the stable domain where the LIM value reaches its maximum, marked in yellow in Fig. 5a, around $\mu = 2$ and $d = 3$. In this area the trivial solution is globally stable, as highlighted by the bifurcation diagrams in Fig. 5c. The Hopf bifurcations that occur around this region at the stability loss are supercritical. Finally, the DI of the equilibrium point is lower at the top and right sides of the stable area, that is, for large values of the derivative gain d and/or the nonlinear damping μ.

Overall, the results showed that the BoA of the fixed point is almost everywhere bounded. Furthermore, the algorithm revealed that the non-convergent trajectories tend towards a stable limit cycle. Therefore, we computed the LIM of this stable limit cycle, and Fig. 5b is colored accordingly.

For the computation, the algorithm first has to identify the stable limit cycle. Thus, the numerical integration has to be initialized within the BoA of the stable limit cycle, preferably as close as possible to the steady state solution to reduce computational time. Here, the computation was performed for gradually decreasing values of the nonlinear damping μ, so that for each new μ value, the computation was initialized with the stable limit cycle obtained from the previous parameter setting as an initial guess.

As can be seen, the stable limit cycle exists for $\mu \gtrapprox 0$ except for the domain where the equilibrium is globally stable. Moreover, the limit cycle is globally attractive where the trivial fixed point is unstable, except for the zero measure set of the fixed point. The amplitude of the limit cycle increases as μ decreases (see Fig. 5c); therefore, on the left side of the figure, the LIM is not limited by the BoA of the fixed point, but by the predefined state space boundary. At a certain point, even the limit cycle exceeds these boundaries; this is the reason why the LIM shrinks to zero as μ tends to 0.

Note that, for larger values of d but still within the stable domain, the basin of attraction of the fixed point is small. This sometimes makes the algorithm, while estimating the LIM of the limit cycle, fail to find any initial condition whose trajectory converges to the fixed point rather than to the stable limit cycle. Here, we should emphasize again that for the estimation of the LIM of a stable limit cycle, the initial condition of a new trajectory is on the limit cycle and only its headpoint is located elsewhere. Therefore, the solution converges to the

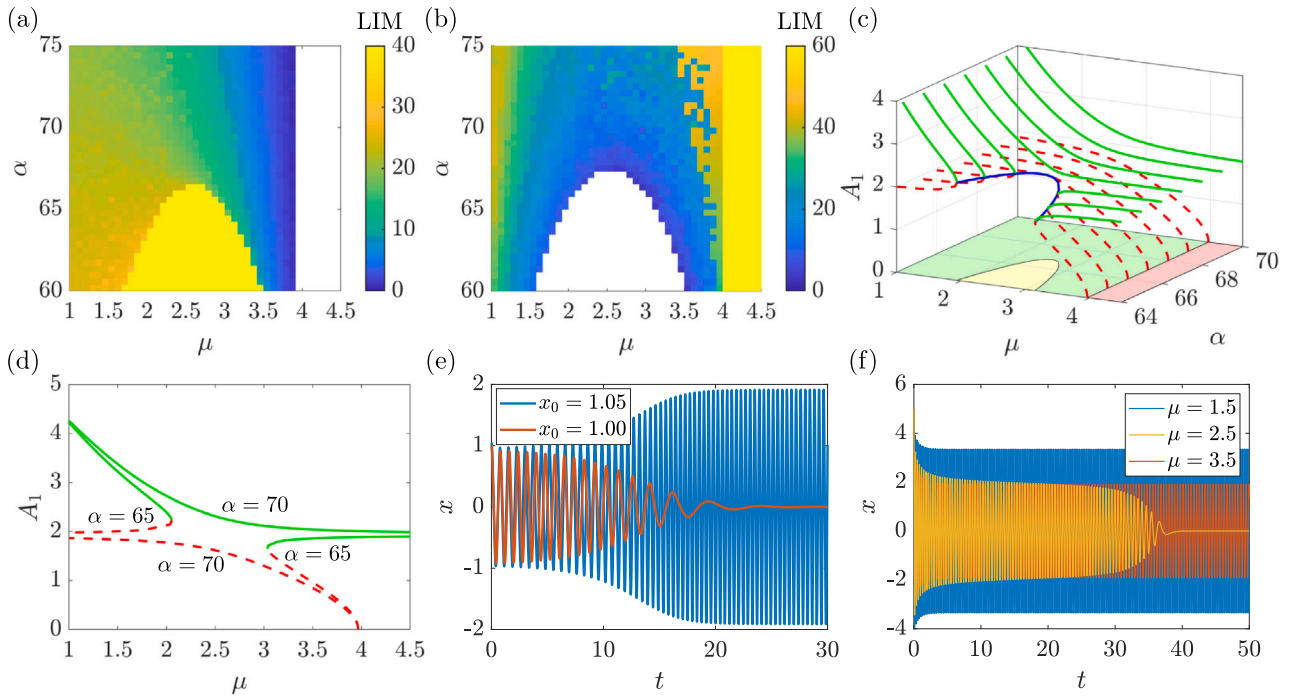


Fig. 6. Nonlinear analysis of the van der Pol-Duffing oscillator with delayed derivative control. Panels (a) and (b) are colored according to the LIM of the stable fixed point and of the stable limit cycle, respectively. Panels (c) and (d) show continuation-based bifurcation diagrams for various values of the nonlinear stiffness α selecting the nonlinear damping μ as the bifurcation parameter. Panel (e) depicts trajectories from slightly different initial conditions for $\alpha = 65$ and $\mu = 3.5$, while panel (f) presents trajectories for $\alpha = 65$ and various values of μ. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

origin with low probability. In these situations, it is worth providing the algorithm the discrete state of the stable equilibrium, since it is a known point of the state-space that is definitely outside of the BoA of the stable limit cycle. So, it limits the LIM.

Fig. 5c presents a bifurcation diagram above the (μ, d) space obtained with the help of the continuation-based MATLAB package called DDE-BIFTOOL [65]. The stable and unstable branches of the limit cycles are presented by dashed red and solid green curves, respectively. The toolbox is also able to follow fold bifurcations; the corresponding curve is depicted in blue. The figure also shows the linear stability boundary, and the projection of the fold bifurcation curve onto the (μ, d) plane. According to the obtained bifurcation diagrams, the equilibrium is globally stable in the yellow region and locally stable in the green domain, while it is unstable in the red area. Finally, the magenta color indicates parameter combinations for which the system has no stable attractor.

This figure confirms the results shown in Fig. 5a and Fig. 5b: as the derivative gain d increases, the amplitude of the stable limit cycle increases as well, while that of the unstable limit cycle decreases. This suggests a decreasing trend in the LIM of the fixed point and an increasing trend in the LIM of the stable limit cycle. It also shows that, as μ is decreased, the amplitude of the stable limit cycle increases, which explains the empty domain on the left side of Fig. 5b as discussed above. In addition, the globally stable domain is obtained for the same parameters as in Fig. 5a and Fig. 5b. As shown in Fig. 5b, the LIM of the limit cycle is low near the fold bifurcation, since even a small perturbation can push the trajectory below the unstable limit cycle, causing it to settle into the stable fixed point.

Note that the combination of parameters $d = 0$, $\mu = 0$, is located on the dynamic stability boundary (see Eqs. (9), (10)); in this case, the system reduces to an undamped Duffing oscillator, that is, the trivial fixed point is a center. This is indicated by the gray vertical line in Fig. 5c. This represents a critical point for the Hopf bifurcation curve, the so-called Bautin point, where the criticality of the bifurcation changes. Another Bautin point is located at $d \approx 1$ where the amplitude of the blue fold bifurcation curve tends to zero. The Hopf bifurcation is subcritical for $d < 0$, then it is supercritical for $0 < d \lesssim 1$; finally, it is again subcritical for $d \gtrsim 1$.

Fixing the derivative gain d, there always exists a branch of stable or unstable limit cycles, emerging from the linear stability boundary. This is easy to obtain with numerical continuation. However, in addition, for the same value of d, there also exists an isolated branch. The presence of these isolas is easy to predict from the LIM of the fixed point (Fig. 5a), whereas they are quite difficult to find with numerical continuation.

Fig. 5d shows the brute-force estimation of the BoA of the fixed point (green dot) for $\mu = 4$ and $d = 6$. The initial conditions were restricted to the same jump type as in the proposed algorithm: the IC history lay on the fixed point, while the IC headpoint was chosen elsewhere in the $(x, \dot{x})$-plane. Time integration was performed with MATLAB's built-in solver dde23, with simulations terminated at a fixed horizon of $t_{\max} = 20$. Each trajectory was then classified according to whether it converged to the fixed point or to the stable limit cycle. Note that if $t_{\max}$ time was not enough for classification, then the trajectory integration was continued and the termination time was extended to $t_{\max,\text{new}} = t_{\max,\text{old}} + 20$.

These simulations were carried out on a 101×101 grid of IC headpoints. The gray domain represents IC headpoints of trajectories converging to the fixed point, while ICs from the white domain converged to the stable limit cycle. The blue dashed circle represents a two-dimensional section of the hypersphere of convergence obtained from the proposed algorithm. Note that the hypersphere of convergence is defined in the 6 dimensional discrete state-space (cf. Fig. 2). As it can be seen, the circle slightly exceeds the true BoA; this is because the algorithm slightly overestimates the true LIM. Nevertheless, the error is considered acceptable, as it does not qualitatively modify the trend of the LIM in the parameter space.

In order to fully understand the advantage provided by the proposed algorithm, it is relevant evaluating its rapidity with respect to a brute

force approach. Computing the brute-force BoA on the 101×101 grid required approximately 15 min. Reconstructing Fig. 5a in this way (which consists of 4590 nonzero LIM values) would require about 1.5 months. In contrast, the proposed method generated the figure in just 5 h and 10 min. Moreover, this is also roughly three times as fast as the algorithm defined in [60], which required 15 h to generate the same figure.

In the following, we fix the differential gain to an arbitrary value of $d = 4$, and we study the dynamical integrity of the existing steady state solutions in the (μ, α) space. For $d = 4$ and $\tau = 0.1$, the Hopf bifurcation occurs at $\mu_{cr} \approx 3.967$ and so the zero equilibrium of the linearized system is stable for $\mu < \mu_{cr}$. As we have already observed in Fig. 5c, the Hopf bifurcation is subcritical, and for some parameter combinations, the system shows bistable behavior.

The estimated LIM of the trivial solution and the stable limit cycle in the 2-dimensional parameter space (μ, α) are presented in Fig. 6a and Fig. 6b, respectively.

As expected, the LIM of the fixed point is small in the vicinity of the stability boundary at $\mu = \mu_{cr}$, and it increases as μ decreases (see Fig. 6a). On the other hand, for $\mu > \mu_{cr}$, the fixed point is unstable and the limit cycle is globally attractive; this leads to the high values of the LIM in the corresponding region of Fig. 6b.

Both figures exhibit a characteristic gate-shaped region with remarkably different LIM values. In this region, the LIM of the fixed point is high (see Fig. 6a), while the algorithm could not found any limit cycle, thus, the LIM in Fig. 6b is undefined. This suggests that the fixed point is globally stable. To investigate it in more detail, we used DDE-BIFTOOL to obtain the bifurcation curves in the (μ, α) parameter space (see Fig. 6c and Fig. 6d). Note that a similar structure is present in the (μ, d) plane at the upper boundary of the yellow, globally attractive domain of the fixed point (see Fig. 5c).

As also obtained in Fig. 5, in Fig. 6b, the LIM of the limit cycle is small near the fold bifurcation, because even a small perturbation is enough to push the solution below the unstable limit cycle, making the system leave its BoA.

Fixing $\alpha = 65$ and $\mu = 3.5$, Fig. 6e presents the response of the system for two slightly different initial conditions. The initial functions were zero except for their headpoint at $t = 0$, which was set to $x_0 = 1.05$ and $x_0 = 1.00$. As it can be observed, the oscillation amplitudes decay for the smaller amplitude initial condition, and the trajectory converges to the fixed point, while in case of the larger amplitude initial condition, the oscillation amplitude increases and the trajectory converges to the stable limit cycle. This confirms the presence of the unstable limit cycle.

Finally, Fig. 6f shows the response of the system fixing the initial function but changing the nonlinear damping parameter μ (for $\alpha = 65$). In the cases of $\mu = 1.5$ and 3.5, the trajectory tends to the corresponding stable limit cycles with larger and smaller amplitudes, respectively. However, when the nonlinear damping value is $\mu = 2.5$, which is just between the two values discussed above, then, the trajectory tends to the origin after an initial transient. The amplitudes of these time-domain simulations are consistent with and confirm the results of the LIM estimation algorithm and bifurcation analysis.

Note that in Fig. 6b, there are some orange colored domains for μ slightly smaller than μ_{cr}. These apparently large LIM values are due to the very small BoA of the trivial solution in that region, which prevents the algorithm from finding any trajectory converging to the equilibrium. This leads to very high LIM values for the periodic solution (which appears globally stable for the algorithm). The same phenomenon was discussed referring to Fig. 5b.

3.2. Inverted pendulum with NLTVA

For the second case study, we consider an inverted pendulum with an attached nonlinear tuned vibration absorber (NLTVA) (see Fig. 7); this configuration was also studied in the papers [60,66]. The tilt angle of the inverted pendulum and that of the vibration absorber are denoted by φ_1 and φ_2, respectively, while the delayed proportional-derivative control torque M intends to stabilize the pendulum in the upward position in the gravitational space.

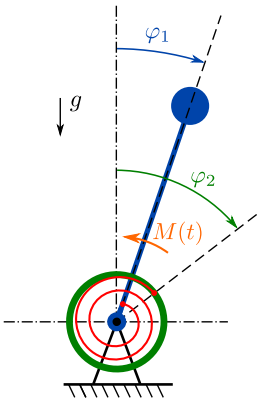


Fig. 7. Mechanical model of the inverted pendulum with nonlinear tuned vibration absorber in the gravitational space. The delayed proportional derivative control torque is denoted by $M(t)$.

The corresponding dimensionless governing equations assume the form

$$\ddot{\varphi}_1(t) - \sin(\varphi_1(t)) + 2\zeta_2\mu\gamma(\dot{\varphi}_1(t) - \dot{\varphi}_2(t)) + \mu\gamma^2(\varphi_1(t) - \varphi_2(t)) = M(t), \tag{11}$$

$$\mu(\ddot{\varphi}_2(t) + 2\zeta_2\gamma(\dot{\varphi}_2(t) - \dot{\varphi}_1(t)) + \gamma^2(\varphi_2(t) - \varphi_1(t))) = 0, \tag{12}$$

where μ refers to the ratio of mass moment of inertia of the NLTVA with respect to that of the inverted pendulum, ζ_2 denotes the relative damping of the NLTVA, and γ is the natural frequency ratio. Moreover, the control torque M is subjected to constant time delay τ, and it takes the form

$$M(t) = -p\varphi_1(t-\tau) - d\dot{\varphi}_1(t-\tau), \tag{13}$$

where p and d denote the dimensionless proportional and derivative control gains, respectively.

In the followings, the system parameters are fixed to $\mu = 0.1$, $\gamma = 2.3$, $\zeta_2 = 0.174$ and $\tau = 0.5$.

First, we apply the proposed algorithm to estimate the LIM of the fixed point; the corresponding results are presented in the upper panels of Fig. 8. Fig. 8a shows a bifurcation diagram obtained with DDE-BIFTOOL for $d = 2.8$. In this case, the fixed point is stable for $p < p_{cr} \approx 2.695$, and a subcritical Hopf bifurcation occurs at the stability boundary. As p is decreased, the amplitude of the corresponding unstable limit cycles increases, while at $p = p_{f,1} \approx 1.257$, this branch folds back yielding stable limit cycles coexisting with the stable fixed point. This is followed by a second fold bifurcation at $p = p_{f,2} \approx 1.502$. Looking even further, a series of fold bifurcations occur, which lead to alternately stable and unstable branches with increasing amplitude.

Fig. 8b presents the estimated LIM of the fixed point as the bifurcation parameter p is varied. The results of the algorithm are consistent with the trend of the unstable branch of the bifurcation diagram. If $p < p_{f,1}$, the zero equilibrium point is globally attractive yielding that the corresponding LIM is infinite. Accordingly, the LIM curve has a sharp turn at that critical value. Note that in the algorithm, the upper limit for the value of the LIM is restricted by the space boundary; still, the prediction of the fold point is good. For increasing p values, the LIM monotonously decreases (apart from numerical errors in the estimation), until reaching zero at the stability loss, for $p = p_{cr}$.

We remark that exactly the same problem was analyzed in [60] with a different algorithm and different definition of the LIM. Qualitatively

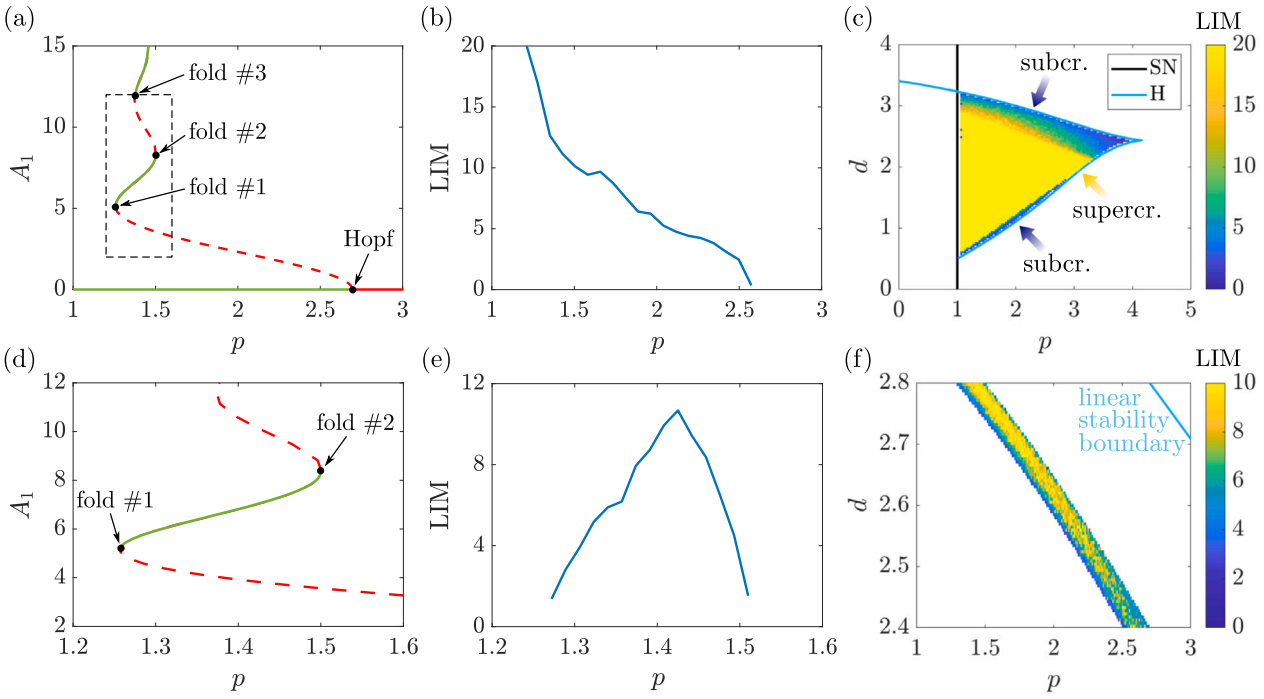


Fig. 8. Nonlinear analysis of the inverted pendulum with NLTVA. Panel (a) shows a bifurcation diagram for $d = 2.8$, while panel (d) presents an enlarged version of that focusing on the stable limit cycle between the first two fold points. Still for $d = 2.8$, panels (b) and (e) show the estimated LIM of the fixed point and of the limit cycle, respectively. Panel (c) depicts a stability chart in the plane of the proportional and differential gains, which is colored according to the LIM of the fixed point. Finally, panel (f) is colored according to the LIM of the stable limit cycle that exists in a strip in the (p, d) plane. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the same results are obtained in that paper as in this current one. However, the new algorithm is more efficient because the LIM estimation is based not only on the headpoint of an initial condition, but on the entire trajectory.

The improved efficiency makes the algorithm applicable for two-dimensional parameter sweeping, which is presented in Fig. 8c. The black and light blue curves show the saddle–node and Hopf bifurcation curves, respectively, while the stable domain is colored according to the value of the estimated LIM. As it can be seen, there is a triangular shaped yellow area, where the fixed point is globally stable. As one goes along the Hopf bifurcation curve from the bottom left corner of the stability chart, the bifurcation is first subcritical, then it turns to be supercritical for a while and, above $d = 2.1$, it is again subcritical. This phenomenon is well captured by the proposed algorithm. The second subcritical branch is more relevant since the corresponding unstable limit cycle penetrates deep into the stable domain. This yields the small values of the LIM, which implies that even a small perturbation is enough to obtain a divergent trajectory. Note that the results in Figs. 8a and 8b also fall within this region.

As it was discussed above, for particular parameter domains, a stable limit cycle coexist with the stable equilibrium. The corresponding analysis is presented in the lower panels of Fig. 8. Fig. 8d shows the first stable branch of the bifurcation curve (see the dashed black rectangle in Fig. 8a).

The DI of the limit cycle is reduced to zero at both the left and right fold points. This can be seen in Fig. 8e, which presents the LIM as the control gain p is varied. We note that the LIM trend presents a sort of non-smoothness for $p \approx 1.425$. This is probably related to the competing effect of the two different unstable limit cycles, which bound the LIM of the periodic solution. For $p < 1.425$ the smaller unstable limit cycle restricts the LIM, while for $p > 1.425$ the larger one delimits it. The transition is marked by the non-smoothness.

Finally, Fig. 8f shows a 2-dimensional parametric analysis of the LIM of the stable limit cycle that exists in the colored strip. As it can be seen, the limit cycle is more robust against external perturbations for larger values of the derivative gain d. Moreover, it can also be observed from this 2-dimensional parametric analysis that at the left and right boundaries of the region of existence of the limit cycle, its LIM tends to zero, which agrees with our expectations. This is better visualized in the section of Fig. 8e, which clearly illustrates the increasing/decreasing trend of the LIM. However, the small region of existence of the stable limit cycle makes it sensitive also to parameter variations.

4. Discussion

The proposed algorithm is a powerful tool to obtain the DI of steady state solutions in time-delay systems. Thanks to its rapidity, it enables parametric analysis for DI investigation, allowing one to determine the parameter domains in which the system is sensitive to external perturbations. In the following, its advantages and limitations are discussed.

This algorithm builds upon the one introduced in [60]. That previous approach was already relatively fast and enabled parametric analysis — something unfeasible with other methods used for DI analysis, especially in time-delay systems. Its computational efficiency was achieved through applying cell subdivision in the state-space for trajectory identification, and directly targeting the LIM instead of the BoA, thereby overlooking intermingled and fractal-like BoA boundaries and avoiding memory-related limitations.

The main difference from [60] lies in the treatment of the state-space. Here, the infinite-dimensional state-space of the time-delay system is also discretized, which allows the full trajectory to be utilized for LIM identification and trajectory comparison. In contrast, the previous

method relied only on the headpoints of the trajectories' initial conditions. This extension not only further accelerates the computation but also enables the analysis of LIMs associated with limit cycles – a key advancement of the current algorithm.

It is worth noting that the discretization of the state-space in the time dimension is coarser than the time discretization of the fourth-order Runge–Kutta scheme used for generating the trajectories. This represents a necessary trade-off between accuracy and computational efficiency.

Although the numerical value of the system's LIM at a single parameter setup is not always informative, the algorithm's rapidity allows one to carry out one or two-dimensional parameter analyses that provide the qualitative trends in the DI. This makes the algorithm an efficient tool for the DI analysis of nonlinear time-delay systems, even in the context of engineering design.

The major limitation of the method is related to the initial functions used for trajectory generation. Although the algorithm can theoretically handle any initial function, it is convenient to restrict them to a specific type to avoid non-physical initial conditions (for example, encompassing multiple jumps in the position and velocity); this also accelerates the computation, by providing faster convergences towards similar regions of the state-space. Regardless of this limitation, most engineering systems have their own typical type of perturbation which can be implemented in the algorithm. This is often a jump in the state of the system, as was done in the current paper. Furthermore, this is a common problem to all methods analyzing the global dynamics of time-delay systems, which always require to restrict the system's initial conditions [58,59].

Also, we should emphasize that in the current form, the algorithm is developed for deterministic systems. In order to be applicable for stochastic systems, one should reconsider the convergence of trajectories and the utilization of the cell structure.

Another limitation is related to the discretization of the state-space in time. In the discretized state-space, a point can correspond to different real states of the infinite-dimensional system, which can lead to errors if the sampling time is too large with respect to the fastest frequencies relevant for the system's dynamics. This problem can be mitigated by reducing the temporal discretization; however, this leads to longer computational time because of the larger dimension of the discrete state-space.

Although the method has no theoretical limitation on the system dimension, its implementation to medium and large dimensional systems will significantly increase computational time, and potentially also generate memory issues. In this cases, it is convenient to implement a model reduction before running the LIM estimation code. In this respect, a first attempt was already made by Habib et al. [67], who analyzed the DI of high-dimensional and experimental systems by first obtaining a reduced-order model through the spectral submanifold reduction [63,68], and then applying the LIM estimation algorithm in [44] to the reduced model. This approach, which revealed accurate and computationally effective, could be easily extended to time-delay systems, since the theory for spectral submanifolds of time-delay systems is already established [64,69].

The algorithm was developed and tested only on retarded systems having a single discrete and fix time delay. Although it can be theoretically extended to other types of time-delay systems, such as neutral DDEs, DDEs with multiple, distributed, state-dependent and time varying delay [55], this would require to solve some non-trivial technicalities, which will be the subject of future studies.

Finally, the algorithm does not investigate the entire state-space, which would be unfeasible for large-dimensional systems. Consequently, it might overlook attractors or narrow tongues of basins of attractions competing with the one of interest. This implies that, although typically the estimated LIM is only slightly larger than the real one, in some cases it might be significantly larger, missing potentially dangerous regions of the state-space. Increasing the number of iterations of the algorithm mitigates the problem but does not solve it completely.

5. Conclusion

A rapid and generally applicable algorithm is introduced for the estimation of the DI of steady states in time-delay systems. The proposed algorithm allows parametric analysis for both fixed points and limit cycles, making it a particularly useful engineering tool for the determination of the robustness of a desired state against external perturbations.

The method is tested in two case studies – a van der Pol-Duffing oscillator with delayed derivative control and an inverted pendulum with NLTVA subjected to delayed proportional-derivative control – both of which have a stable limit cycle coexisting with the stable equilibrium or other solutions. The results were validated with continuation-based computations, which showed a good qualitative match.

Our future goal is to apply the algorithm to non-smooth systems, which poses further challenges, especially in the case of time-delay systems where the state of the system is a function of time [70]. This means that the non-smooth effect appears not only at the switching surface but it also affects the dynamics later, at integer multiples of the time delay, because of the memory effect. A typical example for this is the fly-over effect in machine tool vibrations [71]. Also, we plan to extend the proposed algorithm to be able to calculate the LIM of quasiperiodic and chaotic attractors, enabling it to handle all standard steady states encountered in deterministic systems.

In addition, the LIM estimation of fixed points could be significantly sped up by analytically calculating a conservative state-space region surrounding the solution, which is entirely within the BoA. In this case, the time integration of the trajectories could be interrupted if the system enters this attracting domain. For ordinary differential equations, there are promising Lyapunov function-based approaches [72–74], which might be implemented for this scope. However, it is not straightforward to construct them for generic dynamical systems. Their implementation in time-delay systems poses additional challenges, which might be overcome using Lyapunov-Krasovskii functionals [57]. This will be the subject of future studies.

CRediT authorship contribution statement

Bence Szaksz: Writing – original draft, Visualization, Software, Conceptualization. **Giuseppe Habib:** Writing – review & editing, Supervision, Software, Conceptualization.

Code availability

This study exploits codes available at the GitHub repository https://github.com/GlobalDynamicsRG/DynIn.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] S. Lenci, G. Rega, et al., Global nonlinear dynamics for engineering design and system safety, vol. 588, Springer, Cham, Switzerland, 2019.
[2] G. Orosz, G. Stepan, Subcritical Hopf bifurcations in a car-following model with reaction-time delay, Proc. R. Soc. A 462 (2073) (2006) 2643–2670.
[3] X. Zhao, G. Orosz, Nonlinear day-to-day traffic dynamics with driver experience delay: Modeling, stability and bifurcation analysis, Phys. D: Nonlinear Phenom. 275 (2014) 54–66.
[4] K. Martinovich, A.K. Kiss, Nonlinear effects of saturation in the car-following model, Nonlinear Dynam. 111 (3) (2023) 2555–2569.
[5] T.G. Molnár, G. Orosz, Destroying phantom jams with connectivity and automation: Nonlinear dynamics and control of mixed traffic, Transp. Sci. 58 (6) (2024) 1319–1334.
[6] Z. Zhou, Y. Wang, G. Zhou, X. Liu, M. Wu, K. Dai, Vehicle lateral dynamics-inspired hybrid model using neural network for parameter identification and error characterization, IEEE Trans. Veh. Technol. 73 (11) (2024) 16173–16186.
[7] T.G. Molnar, T. Insperger, G. Stepan, Closed-form estimations of the bistable region in metal cutting via the method of averaging, Int. J. Non-Linear Mech. 112 (2019) 49–56.
[8] Z. Dombovari, A. Iglesias, T.G. Molnar, G. Habib, J. Munoa, R. Kuske, G. Stepan, Experimental observations on unsafe zones in milling processes, Phil. Trans. R. Soc. A 377 (2153) (2019) 20180125.
[9] J. Lelkes, T. Kalmar-Nagy, Bifurcation analysis of a forced delay equation for machine tool vibrations, Nonlinear Dynam. 98 (4) (2019) 2961–2974.
[10] V.K. Jirsa, W.C. Stacey, P.P. Quilichini, A.I. Ivanov, C. Bernard, On the nature of seizure dynamics, Brain 137 (8) (2014) 2210–2230.
[11] M.A. Kramer, S.S. Cash, Epilepsy as a disorder of cortical network organization, Neurosci. 18 (4) (2012) 360–372.
[12] P.J. Menck, J. Heitzig, N. Marwan, J. Kurths, How basin stability complements the linear-stability paradigm, Nat. Phys. 9 (2) (2013) 89–92.
[13] M. Rohden, A. Sorge, M. Timme, D. Witthaut, Self-organized synchronization in decentralized power grids, Phys. Rev. Lett. 109 (6) (2012) 064101.
[14] M. Chertkov F. Dörfler, F. Bullo, Synchronization in complex oscillator networks and smart grids, Proc. Natl. Acad. Sci. 110 (6) (2013) 2005–2010.
[15] D. Takacs, G. Orosz, G. Stepan, Delay effects in shimmy dynamics of wheels with stretched string-like tyres, Eur. J. Mech. A Solids 28 (3) (2009) 516–525.
[16] S. Beregi, D. Takacs, G. Stepan, Bifurcation analysis of wheel shimmy with non-smooth effects and time delay in the tyre–ground contact, Nonlinear Dynam. 98 (1) (2019) 841–858.
[17] G. Habib, A. Epasto, Towed wheel shimmy suppression through a nonlinear tuned vibration absorber, Nonlinear Dynam. 111 (10) (2023) 8973–8986.
[18] S. Schroders, A. Fidlin, Asymptotic analysis of self-excited and forced vibrations of a self-regulating pressure control valve, Nonlinear Dynam. 103 (2021) 2315–2327.
[19] F. Kadar, G. Stepan, Nonlinear dynamics and safety aspects of pressure relief valves, Nonlinear Dynam. 111 (13) (2023) 12017–12032.
[20] Z. Veraszto, G. Stepan, Nonlinear dynamics of hardware-in-the-loop experiments on stick–slip phenomena, Int. J. Non-Linear Mech. 94 (2017) 380–391.
[21] G. Habib, A. Bartfai, A. Barrios, Z. Dombovari, Bistability and delayed acceleration feedback control analytical study of collocated and non-collocated cases, Nonlinear Dynam. 108 (3) (2022) 2075–2096.
[22] B. Szaksz, G. Stepan, Delay-induced bifurcations in collocated position control of an elastic arm, Nonlinear Dynam. 107 (2) (2022) 1611–1622.
[23] A. Malher, G. Habib C. Touzé, G. Kerschen, Flutter control of a two-degrees-of-freedom airfoil using a nonlinear tuned vibration absorber, J. Comput. Nonlinear Dyn. 12 (5) (2017) 051016.
[24] B. Takarics, B. Patartics, T. Luspay, B. Vanek, C. Roessler, J. Bartasevicius, S.J. Koeberle, M. Hornung, D. Teubl, M. Pusch, et al., Active flutter mitigation testing on the flexop demonstrator aircraft, in: AIAA Scitech 2020 Forum, 2020, p. 1970.
[25] A. Drachinsky, O. Avin, D.E. Raveh, M. Tur Y. Ben-Shmuel, Flutter tests of the pazy wing, AIAA J. 60 (9) (2022) 5414–5421.
[26] D. Costa, R. Kuske, D. Yurchenko, Qualitative changes in bifurcation structure for soft vs hard impact models of a vibro-impact energy harvester, Chaos 32 (10) (2022) 103120.
[27] Y. Fan, Y. Liao, X. Yang, M.-Q. Niu, L.-Q. Chen, A nonlinear multi-stable electromagnetic energy harvester with segmented moving magnet configuration, Int. J. Non-Linear Mech. (2025) 105203.
[28] S. Chiacchiari, F. Romeo, L.A. Bergman D.M. McFarland, A.F. Vakakis, Vibration energy harvesting from impulsive excitations via a bistable nonlinear attachment, Int. J. Non-Linear Mech. 94 (2017) 84–97.
[29] D. Juher, D. Rojas, J. Saldaña, Saddle–node bifurcation of limit cycles in an epidemic model with two levels of awareness, Phys. D: Nonlinear Phenom. 448 (2023) 133714.
[30] F.A. Rihan, H.J. Alsakaji, C. Rajivganthi, Stochastic SIRC epidemic model with time-delay for COVID-19, Adv. Difference Equ. 2020 (1) (2020) 502.
[31] P.T. Nagy, M. Kulcsár, Predicting the energy stability limit of shear flows using weighted velocity components, Phys. Fluids 35 (10) (2023).
[32] S. Ratschan, Z. She, Providing a basin of attraction to a target region of polynomial systems by computation of Lyapunov-like functions, SIAM J. Control Optim. 48 (7) (2010) 4377–4394.
[33] S. Gudmundsson, S. Hafstein, Probabilistic basin of attraction and its estimation using two Lyapunov functions, Complexity 2018 (1) (2018) 2895658.
[34] C.S. Hsu, A theory of cell-to-cell mapping dynamical systems, J. Appl. Mech. 47 (4) (1980) 931–939.
[35] C.S. Hsu, Cell-to-cell mapping: a method of global analysis for nonlinear systems, vol. 64, Springer, New York, 2013.
[36] P. Belardinelli, S. Lenci, An efficient parallel implementation of cell mapping methods for mdof systems, Nonlinear Dynam. 86 (2016) 2279–2290.
[37] J.-Q. Sun, F.-R. Xiong, O. Schütze, C. Hernández, Cell Mapping Methods, Springer, Singapore, 2018.
[38] J.M.T. Thompson, Chaotic phenomena triggering the escape from a potential well, Proc. R. Soc. A 421 (1861) (1989) 195–225.
[39] M. Soliman, J. Thompson, Integrity measures quantifying the erosion of smooth and fractal basins of attraction, J. Sound Vib. 135 (3) (1989) 453–475.
[40] S. Lenci, G. Rega, Optimal control of homoclinic bifurcation: theoretical treatment and practical reduction of safe basin erosion in the Helmholtz oscillator, J. Vib. Control 9 (3–4) (2003) 281–315.
[41] D. Orlando, G. Rega P.B. Gonçalves, S. Lenci, Influence of transient escape and added load noise on the dynamic integrity of multistable systems, Int. J. Non-Linear Mech. 109 (2019) 140–154.
[42] G. Rega, S. Lenci, Identifying, evaluating, and controlling dynamical integrity measures in non-linear mechanical oscillators, Nonlinear Anal. 63 (5–7) (2005) 902–914.
[43] S. Lenci, G. Rega, Optimal control of nonregular dynamics in a duffing oscillator, Nonlinear Dynam. 33 (2003) 71–86.
[44] G. Habib, Dynamical integrity assessment of stable equilibria: a new rapid iterative procedure, Nonlinear Dynam. 106 (3) (2021) 2073–2096.
[45] D. Patko, G. Habib, Dynamical integrity of periodic orbits: A computationally efficient approach, Nonlinear Dynam. 113 (2025) 24641–24661.
[46] T.G. Molnar, Z. Dombovari, T. Insperger, G. Stepan, On the analysis of the double Hopf bifurcation in machining processes via centre manifold reduction, Proc. R. Soc. A: Math. Phys. Eng. Sci. 473 (2207) (2017) 20170502.
[47] G. Habib, G. Kerschen, G. Stepan, Chatter mitigation using the nonlinear tuned vibration absorber, Int. J. Non-Linear Mech. 91 (2017) 103–112.
[48] L. Zhang, S.A. Campbell, L. Huang, Nonlinear analysis of a maglev system with time-delayed feedback control, Phys. D: Nonlinear Phenom. 240 (21) (2011) 1761–1770.
[49] D. Costa, V. Vaziri, E. Pavlovskaia, M.A. Savi, M. Wiercigroch, Switching between periodic orbits in impact oscillator by time-delayed feedback methods, Phys. D: Nonlinear Phenom. 443 (2023) 133587.
[50] A. Martin, S. Ruan, Predator-prey models with delay and prey harvesting, J. Math. Biol. 43 (2001) 247–267.
[51] R. Xu, Q. Gan, Z. Ma, Stability and bifurcation analysis on a ratio-dependent predator–prey model with time delay, J. Comput. Appl. Math. 230 (1) (2009) 187–203.
[52] P. Mishra, A. Ponosov, J. Wyller, On the dynamics of predator–prey models with role reversal, Phys. D: Nonlinear Phenom. 461 (2024) 134100.
[53] K. Green, B. Krauskopf, K. Engelborghs, Bistability and torus break-up in a semiconductor laser with phase-conjugate feedback, Phys. D: Nonlinear Phenom. 173 (1–2) (2002) 114–129.
[54] J. Wang, H. Wang, J. Song, X. Chen, J. Guo, K. Li, X. Li, B. Huang, Knowledge-guided self-learning control strategy for mixed vehicle platoons with delays, Nat. Commun. 16 (1) (2025) 7705.
[55] J.K. Hale, S.M. Verduyn Lunel, Introduction to functional differential equations, vol. 99, Springer, New York, 1993.
[56] G. Stepan, Retarded Dynamical Systems: Stability and Characteristic Functions, Longman Scientific & Technical, Harlow, 1989.
[57] J.B. Biemond, W. Michiels, Estimation of basins of attraction for controlled systems with input saturation and time-delays, IFAC Proc. Vol. 47 (3) (2014) 11006–11011.
[58] T.H. Scholl, V. Hagenmeyer, L. Gröll, On norm-based estimations for domains of attraction in nonlinear time-delay systems, Nonlinear Dynam. 100 (3) (2020) 2027–2045.
[59] K. Yoshida, K. Konishi, N. Hara, Basins and bifurcations of a delayed feedback control system and its experimental verification for a DC bus circuit, Nonlinear Dynam. 106 (2021) 2363–2376.
[60] B. Szaksz, G. Stepan, G. Habib, Dynamical integrity estimation in time delayed systems: a rapid iterative algorithm, J. Sound Vib. 571 (2024) 118045.
[61] T.A. Burton, Volterra Integral and Differential Equations, Academic Press, New York, 1983.
[62] G.A. Vio, G. Dimitriadis, J.E. Cooper, Bifurcation analysis and limit cycle oscillation amplitude prediction methods applied to the aeroelastic galloping problem, J. Fluids Struct. 23 (7) (2007) 983–1011.
[63] G. Haller, S. Ponsioen, Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction, Nonlinear Dynam. 86 (2016) 1493–1534.
[64] B. Szaksz, G. Orosz, G. Stepan, Spectral submanifolds in time delay systems, Nonlinear Dynam. (2025) 1–22.
[65] J. Sieber, K. Engelborghs, T. Luzyanina, G. Samaey, D. Roose, DDE-BIFTOOL manual-bifurcation analysis of delay differential equations, 2014, arXiv preprint arXiv:1406.7144.

[66] G. Habib, Suppression of time-delayed induced vibrations through the dynamic vibration absorber: Application to the inverted pendulum, in: Topics in Nonlinear Mechanics and Physics, Springer, Singapore, 2019, pp. 125–140.
[67] G. Habib, M. Cenedese, G. Rega, Spectral submanifold reduction of high dimensional nonlinear systems for dynamical integrity analysis, Nonlinear Dynam. (2025) http://dx.doi.org/10.1007/s11071-025-11668-1.
[68] M. Cenedese, K. Avila J. Axås, G. Haller, Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds, Nat. Commun. 13 (1) (2022) 872.
[69] G. Buza, G. Haller, Existence of spectral submanifolds in time delay systems, 2025, arXivArXiv:2512.22062.
[70] Z. Iklodi, Z. Dombovari, Bifurcation analysis of piecewise-smooth engineering systems with delays through numeric continuation of periodic orbits, Nonlinear Dynam. 112 (24) (2024) 21789–21818.
[71] Z. Dombovari, G. Stepan, On the bistable zone of milling processes, Phil. Trans. R. Soc. A 373 (2051) (2015) 20140409.
[72] A. Levin, An analytical method of estimating the domain of attraction for polynomial differential equations, IEEE Trans. Autom. Control 39 (12) (1994) 2471–2475.
[73] W. Tan, A. Packard, Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming, IEEE Trans. Autom. Control 53 (2) (2008) 565–571.
[74] M. Zarei, A. Kalhor, D. Brake, Arc length based maximal Lyapunov functions and domains of attraction estimation for polynomial nonlinear systems, Automatica 90 (2018) 164–171.