Nonlinear Dynamics (2026) 114:920
https://doi.org/10.1007/s11071-026-12761-9

RESEARCH

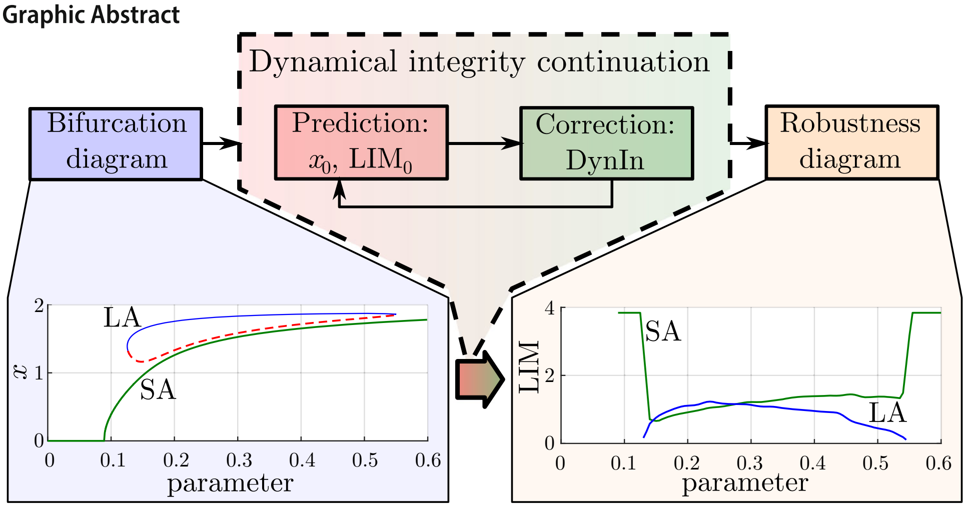
Rapid mapping of basin stability: A numerical continuation framework for dynamical integrity

Márton Szabó[1,2] · **Giuseppe Habib**[1,2]

Received: 10 January 2026 / Revised: 2 June 2026 / Accepted: 13 June 2026

Abstract In this paper, for the first time, a numerical continuation algorithm for dynamical integrity assessment—also known as basin stability—is developed. The method exploits information about the dynamical integrity of previous parameter sets—computed through a DynIn based algorithm—for estimating the local integrity measure of a neighboring parameter set, whose value is then corrected. Overall, the algorithm resembles a classical predictor-corrector scheme used for numerical continuation of steady-state solutions, but it is applied to the local integrity measure. The method is tested on three diverse dynamical systems, namely, a non-autonomous Duffing oscillator, a van der Pol-Duffing oscillator with an attached vibration absorber (autonomous system), and an industrial model of a self-synchronizing vibrating screen. Despite the challenging scenarios encountered, encompassing fractal basins of attraction, several coexisting stable solutions and moderately large dimensional systems, the method was able to rapidly and accurately generate robustness maps for all of them, illustrating the systems' dynamical integrity for large ranges of the parameter sets. The algorithm also unveiled branches of isolated periodic solutions, which would be unattainable by traditional continuation methods. Additionally, we extended the capabilities of the DynIn toolbox, enabling it to handle chaotic solutions and to efficiently compute basins of attraction sections.

Graphic Abstract



Keywords Dynamical integrity · Numerical continuation · Parametric study · Robustness · Basin stability

✉ Márton Szabó
marton.szabo@mm.bme.hu

Giuseppe Habib
habib@mm.bme.hu

[1] Department of Applied Mechanics, Budapest University of Technology and Economics, Műegyetem rkp. 3., Budapest H-1111, Hungary

[2] MTA-BME Lendület Global Dynamics Research Group, Műegyetem rkp. 3., Budapest H-1111, Hungary

Published online: 17 July 2026

1 Introduction

Engineers dealing with nonlinear dynamical systems often face challenges related to the existence of more than one steady-state solution. In these circumstances, the local stability of a steady state does not guarantee its global stability, meaning that external perturbations larger than a certain magnitude can cause the system to leave the desired solution and diverge, or converge to another state. Thus, a measure to quantify the actual, practical stability of an attractor is needed. In order to characterize the robustness of a steady-state solution to withstand external perturbations to the operating initial conditions, the terms dynamical integrity [1, 2] or basin stability [3] are often used.

The importance of dynamical integrity has already been highlighted in engineering practice [2, 4, 5], since this matter can occur in a wide variety of problems. In most cases, the examination of the robustness is necessary for preventing the system from diverging towards undesirable motions when perturbations are not infinitesimal. For instance, a rolling wheel can diverge from its stable equilibrium towards shimmy oscillations as a result of an external force [6]. In a similar fashion, power grids might experience limit cycle oscillations if their equilibrium is disturbed, leading to blackouts [7], airplane wings can undergo flutter vibrations triggered by gusts [8], and epileptic seizures may be initiated as a result of afferent perturbation [9]. There are, however, other cases, where the complete opposite is the goal of the investigation. For example, in snap-through actuators used in micro-electromechanical systems (MEMS), it is favorable to have minimal dynamical integrity to ensure that less voltage is sufficient for the operation [10].

As the name already suggests, the basin stability of an attractor can be studied through the size of its basin of attraction (BoA), which is the set of all initial conditions in the phase space, from which the trajectories asymptotically converge to that attractor; in a general sense, the larger this basin is, the more robust the solutions will be to external impacts [3]. However, also its shape is very important towards a solution's dynamical integrity, as thoroughly discussed in [11], and well expressed by the dynamical integrity measures, as detailed later.

Unfortunately, analytical methods used for the approximation of the BoA, generally based on Lyapunov functions, are usually difficult, or even unfeasible for complex, or large dimensional systems [12–14], which indicates the need for numerical methods for this objective.

Traditionally, numerical methods include the discretization of the phase space into cells, after which, each cell is classified based on where the trajectory initiated there converges to [15]. Even though this approach has the advantage of providing detailed information about the size and shape of the BoA, it is computationally very inefficient and expensive, since from each trajectory it collects only one bit of information. This problem is brilliantly overcome through the cell mapping method [16], which exploits a lot of very short trajectories to generate a Markov graph, which enables the identification of the BoA. However, the cell mapping method suffers significantly from high memory requirements, which increase exponentially with the system dimension [17–19]. Likewise, statistical methods, based on Monte Carlo sampling [20], can slightly reduce the computational cost; however, they do not provide any insight about the BoA's shape, and they might miss rare events if the number of samples is not large enough [21].

Another problem of BoAs, especially relevant for engineering applications, is their quantitative evaluation. Since they can have very complex and exotic shapes, typically involving fractal geometries, this task is not trivial. Attempting to solve this problem, Soliman and Thompson [1] and later Lenci and Rega [11] introduced the so-called dynamical integrity measures, which allow one to quantify a BoA with a single scalar quantity. Several integrity measures exist, like the global integrity measure, which is the volume of the BoA [22], or the local integrity measure (LIM), which is defined as the radius of the largest hypersphere centered at the steady state and entirely included within the solution's BoA [1]. This latter measure has the advantage that it excludes fractal regions of the basin, and quantifies only the compact and safe portion of the BoA.

In this context, an algorithm was introduced in [23] to directly estimate the LIM of a steady-state solution, without computing the BoA. The idea behind the method is to perform only a limited number of simulations and exploit a cell subdivision of the phase space for classifying the trajectories as either convergent to the desired solution or divergent. This allows a rapid exploration of the phase space and permits the iterative calculation of the LIM as the minimal distance between the steady-state solution and any divergent trajectory. Overall, at the expense of full BoA knowledge, this approach drastically reduces computational time, while delivering a single scalar value that quantifies the dynamical integrity of a steady state, making it a suitable method for practical applications. The algorithm in [23] was first developed for fixed points, later generalized for the periodic orbits of autonomous and non-autonomous systems in [24], and for time delayed systems in [25]. The methods ability to assess non-smooth systems is also highlighted in these studies. This set of algorithms is implemented in the open source `DynIn` Toolbox. In a recent study [26], the method was also coupled with the spectral submanifold model reduction technique for addressing large and experimental systems.

In engineering design, parametric analysis has an essential role in aiding design and parameter optimization. The general objective of this study is to optimize the algorithm developed in [23] and [24] for parametric analysis, as detailed below. Parametric studies of nonlinear systems normally include the continuation of the steady-state solutions and bifurcation analysis [27]. Commonly, numerical continuation methods are applied for these tasks, because they are easier to automate than analytical methods [28]. These algorithms are widely available nowadays for fixed points [29, 30], and, through shooting methods or boundary value problems, for limit cycles of autonomous systems [31–33], or periodic orbits in non-autonomous systems [28]. They have also been implemented in open source software, such as `Auto` [34], `MatCont` [35], `CoCo` [36], `Manlab` [37], and `DDEBifTool` [38], to aid engineers. Even though there is great variety in the methods, they usually involve the utilization of a predictor-corrector scheme, which allows for the effective continuation of solution branches. The prediction step, generally through a tangent vector, achieves the reduction of iterative steps needed by the corrector, commonly a Newton-Raphson iteration, while also making the algorithm able to continue a given branch through a turning point [33]. Numerical approximation of the monodromy matrix also makes these algorithms capable of computing the local stability of steady states from the Floquet multipliers, thus making bifurcation analysis possible [39].

All the mentioned algorithms limit their range of analysis to the surrounding of the steady state. In other words, they perform a local analysis of the system and its evolution for parameter variations. Up to the authors' knowledge, currently no method for performing the parameter continuation of a global analysis exists, while such an approach was only theorized in [40]. Thus, in this study, we aim to develop a numerical method, which can not only follow the steady-state solution of a nonlinear system, but is also able to continue its dynamical integrity, making it applicable in practical engineering problems. We use a predictor-corrector scheme to speed up safe basin approximations, while also increasing the precision of the estimations. As detailed in the next section, the algorithm can rapidly identify the majority, if not all, of the solutions of a system, helping to acquire knowledge about the entire phase space and not just one solution.

In addition to the continuation algorithm, this study proposes two developments of the `DynIn` set of algorithms. First, it exploits the `DynIn` cell subdivision framework to rapidly identify BoA sections, overcoming the memory issues of the cell-mapping method [16], at the price of restricting the result to single sections of the BoA (not necessarily bi-dimensional). Then, we extend the set of analyzable steady-state solutions to chaotic ones, rarely addressed in global analysis studies.

The remaining part of the paper is organized as follows. In Sect. 2, we further detail the goals of the algorithm, and describe the realized computational method, specifying its capabilities. In Sect. 3, we assess the performance of the algorithm on three different systems, namely a forced Duffing-oscillator, a Van der Pol-Duffing oscillator supplemented with a nonlinear vibration absorber, and a self-synchronizing vibrating screen model. A brief discussion about the obtained results, their practical implications, and the limitations of the method are discussed in the concluding section.

2 Algorithm

The main objective of the present study is to develop an algorithm, which allows for the rapid assessment of dynamical integrity over a given parameter range, leveraging the algorithms for quick robustness assessment implemented in the `DynIn` toolbox [23, 24]. The method should exploit the information about the system's dynamics, acquired for a parameter set, for accelerating the dynamical integrity estimation of another parameter set in close proximity to the previous one. Essentially, this approach constitutes the first attempt to apply concepts of numerical continuation to dynamical integrity assessment.

Additionally, two important features, so far missing in the `DynIn` toolbox, are introduced in this study. Firstly, an algorithm for computing entire sections of the BoA, which exploits a similar cell subdivision approach as the fundamental `DynIn` algorithm [41]. This allows for a very rapid computation of BoA sections, which is relevant both for validating results about the LIM, and for obtaining a deeper insight for particularly important parameter sets. Secondly, the capabilities of `DynIn` are extended to the LIM estimation of chaotic solutions, by exploiting a discretization of the solution. The same approach can be utilized also for quasiperiodic solutions, which, together with equilibrium points and periodic solutions, cover all standard bounded steady-state solutions that finite dimensional dynamical systems can experience [42].

The continuation algorithm, the BoA identification algorithm and the LIM estimation for chaotic solutions are discussed in the following three subsections.

2.1 Continuation algorithm

In order to perform the above described continuation task, the algorithm should first of all numerically continue a user provided solution branch, either a fixed point or a periodic orbit of autonomous or non-autonomous systems, while also addressing the local stability of the solutions. Information about a branch of steady-state solutions and their linear stability properties constitute the base on which the dynamical integrity assessment is performed.

The numerical continuation implements the shooting method coupled with a pseudo-arclength algorithm, for the continuation of fixed points and periodic orbits, as described in [28, 33]. Information about the stability of the solutions

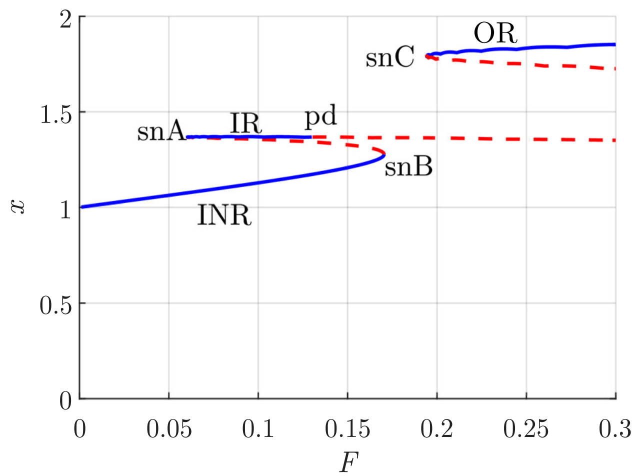


Fig. 1 Bifurcation diagram of the periodic solutions of the system described in (1), with parameters $\zeta = 0.05$ and $\omega = 1.1$. Stable and unstable solutions are represented by solid and dashed lines, respectively. Here, INR, IR, and OR represent the in-well non-resonant, in-well resonant, and out-of-well resonant solutions of the system, while snA, snB, snC and pd refer to saddle-node and period-doubling bifurcations

along the branch is important to avoid computing the dynamical integrity of unstable solutions, which would always be zero. This is usually obtained by numerically calculating the monodromy matrix of the solutions [28]. Then, based on the eigenvalues of the said matrix, the stability and bifurcation points of the system are computed. This approach can be well integrated into the continuation of the solution branch and can provide the stability metrics before the continuation of the dynamical integrity is initiated. Thus, this part of the algorithm produces the required information regarding the assessed solution branch, and prepares the ground for the subsequent LIM calculation. Results achieved by this method can be seen on Fig. 1, where the bifurcation diagram of the system described later in Section 3.1 was computed using the pseudo-arclength continuation algorithm.

Once the solution branch has been computed, the solutions are saved only for a user provided set of parameter values. The knowledge of some of the existing steady states significantly accelerate the estimation of the LIM performed later, by facilitating the trajectory classification, which is particularly time consuming if a trajectory converges towards a previously unknown solution.

The next step of the method, the dynamical integrity estimation, is based on a predictor-corrector scheme, where the correction is done by the iterative procedure of the incorporated `DynIn` software [23, 24]. Therefore, the main innovation of our algorithm is the provision of inputs to the correction phase, which would allow it to perform faster, as well as need less iterating steps, through the prediction phase. A sound analogy is the pseudo-arclength continuation method, where the predictor step's purpose is to reduce the number of iterations required by the Newton-Raphson method to converge [31]. Thus, since the corrector step estimates the LIM as a measure of robustness, the prediction's primary objective is to prepare a good first estimate of this measure making the convergence of the corrector step to the final LIM value require less computational time and provide more precise results.

The key point of LIM estimation is the identification of the closest portion of the BoA boundary to the solution, that we call critical point. Although this can suddenly and discontinuously change for parameter variations, it generally varies continuously with the parameters. Accordingly, the predictor should transfer the information about the critical point from the previous to the following parameter set. Two different approaches are applied for this purpose. The first one consists of providing the exact coordinate—in the phase space—of the critical point identified for the previous parameter set to the following one, while the second approach consists of analyzing the trend of the LIM, then extrapolating from it the predicted LIM value, and from variations of the steady-state solution's position, the predicted position of the critical point. In the continuation scheme developed, both approaches are implemented, constituting the first and second phase of the prediction, as detailed below. Obviously, if the critical region has a discontinuous change, both approaches will not lead to a correct estimation; however, they may still provide a LIM estimate close to the exact one. In fact, a discontinuous variation of the critical zone does not imply a discontinuity in the LIM value. However, if the parameter variation leads to the appearance of a new BoA boundary that also makes the LIM drop—for example in correspondence of a fold bifurcation of another branch—the prediction will need a significant correction, since in this case, the predicted values are no better than the random initial conditions used by the classical `DynIn`.

In the first phase of the prediction, a trajectory is initiated from the critical point of the previous parameter set. If the trajectory is divergent, then the closest point of this trajectory to the solution already provides the first overestimate of the LIM. Conversely, if it is convergent, we increase the previous critical point's distance from the currently investigated solution (denoted by d on Fig. 2) by 10%, obtaining a point from which another trajectory is initiated, whose convergence properties are also investigate. The procedure is continued until a divergent solution is obtained, or for a predefined maximal number of attempts. In the investigated case studies, discussed in Sect. 3, rarely more than three simulations were required for this phase. Once a divergent trajectory is detected, we attain a first overestimate of the LIM, and the iterative procedure of the corrector can be initiated. It must be noted that the 10% increase used here was determined based on practical experience. The procedure is graphically illustrated in Fig. 2, where, in accordance with Fig. 3, the previous LIM value is LIM_4.

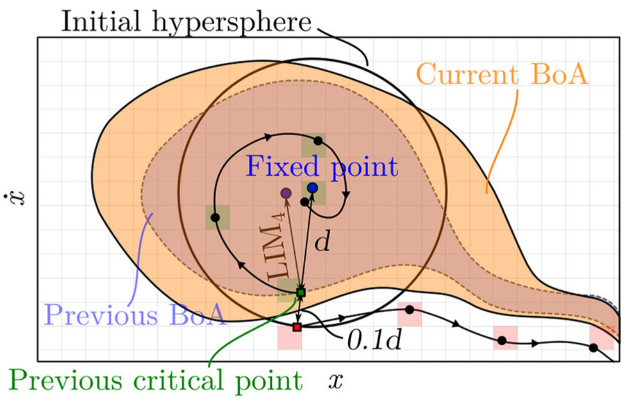


Fig. 2 Prediction procedure of the initial hypersphere of convergence

In the second phase of the prediction, initially the LIM value is extrapolated from the LIM values obtained for the previous parameter sets. From the predicted LIM, the critical point is estimated considering the same relative direction from the steady state to the critical point as for the previous parameter set, but from the current steady state, and considering the predicted LIM. A trajectory is then initiated from this point, constituting the first step of the correction algorithm. The algorithm is visualized in Fig. 3, where LIM_{pred} is the extrapolated LIM, and the vector connecting the solution to the predicted critical point is parallel to the vector connecting the previous solution to the corresponding critical point. The LIM extrapolation is a cubic spline extrapolation, which takes into account the 4 previous values, using a curve with C_2 continuity. When there are only three or less previous values available it falls back to quadratic or linear extrapolation, respectively, and if no previous LIM value is present, the point provided by the second phase is chosen randomly. The choice of spline extrapolation was made so that several previous data points are considered; however, further analysis not included in the paper, revealed that using linear extrapolation or cubic convolution would give very similar results.

In essence, while the first action aims to find the critical point in the exact area of the phase space, where the previous point was found, the second action assumes it shifts together with the steady state, and predicts its value accordingly. The first phase also has the advantage that it provides the initial LIM value as a radius of a hypersphere, which is sure to contain the critical point. This way, the region of the phase space that needs to be examined by the corrector is shrunken inside this hypersphere. Even though the algorithm explores both options and computes them methodologically in parallel, the prediction step does not produce significant computational cost grows.

The two prediction actions could be combined by looking at the evolution of the critical point for variations of the parameter values, and estimating the new critical position through a polynomial extrapolation of the coordinate of the

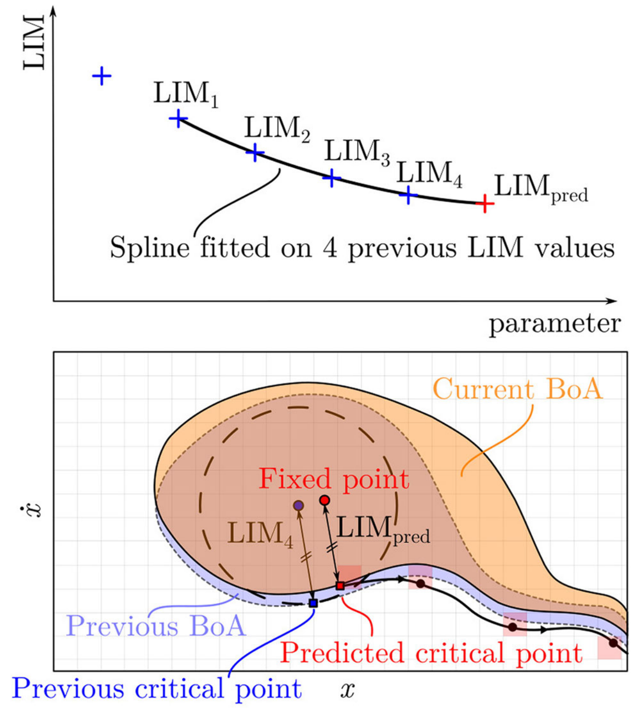


Fig. 3 Prediction procedure of the first initial condition

critical position. Theoretically, this approach could be very effective, but it requires a very good knowledge of the critical point, which typically is not obtained with the proposed method. In fact, while the `DynIn` algorithm has a relatively good accuracy in terms of LIM estimated value, it is much less accurate regarding the critical point position. This occurs because often far apart points of the BoA boundary have very similar distance from the solution, an issue detailed in [40], which might lead to inaccurate estimation of the critical point's exact position, especially in large dimensional systems. Therefore, breaking the prediction phase into two actions provides better results than combining them. Additionally, we note that inaccuracies in the prediction step do not compromize the accuracy of the LIM estimation, but only reduce or eliminate the gain in computational time provided by a correct prediction.

In the case of periodic orbits, the solutions are discretized into a finite number of points for the LIM computation [24]. For each parameter value we use the same number of points to represent these steady states. Thus, in the same way we defined the critical point of the BoA boundary, we can define a critical point for the periodic solutions as well, which is the closest to the boundary of its basin, and from which the LIM is computed. For periodic orbits of periodically excited systems this critical point of the solution would define a critical time section too. During the prediction phase both actions are computed using the nearest point of the orbit to the point of the previous solution, from which the final LIM value was obtained.

Once the prediction phase is completed, the algorithm proceeds with the corrector phase, according to the algorithm described in details in [23, 24]. The most important steps of the algorithm are recalled here. First, (i) boundaries of the phase space are defined; trajectories crossing the boundary, even if temporarily, are considered divergent. Then, (ii) the phase space is divided into small cells. After that, (iii) trajectories are computed one by one, and their convergence properties are automatically calculated leveraging the cell subdivision, as detailed in [23]. Cells containing points of a trajectory are classified as converging or diverging depending on the convergence of the trajectory they belong to. If a trajectory reaches a cell already classified, its computation is interrupted and it is classified as convergent or divergent depending on the reached cell classification. (iv) After each diverging trajectory, the LIM is estimated as the minimal distance between the steady-state solution under study and any point of the diverging trajectory. Initial conditions of each trajectory are computed either through a bisection method, aiming at initiating them as close as possible to a BoA boundary [25], or through a minimal distance algorithm, which tries to provide a homogeneous exploration of the phase space [23]. While the former method is more accurate and faster, the latter one reduces the probability of overlooking other solutions or regions of basin erosion. In both cases, initial conditions are chosen withing the hypersphere marked by the last LIM estimate, or in its close vicinity. The procedure continues until a predefined number of iterations is reached. We note that also the initial trajectories, computed for the prediction of the initial hypersphere, are classified according to the cell subdivision, providing a substantial reduction in computational time. Importantly, we highlight that, unlike the cell-mapping method [16], only a small portion of the cells are classified, which prevents the algorithm from having significant memory issues in large dimensional systems.

While the original `DynIn` algorithms [23–25] compute the LIM for one of the coexisting solutions at the time, we modified the procedure such that information obtained for one solution is used for the others. In particular, the algorithm stores the convergent cells of all identified stable solutions and, when analyzing one specific solution, reclassifies the convergent cells of the other solutions as divergent, since from the perspective of the investigated solution those trajectories are effectively divergent and the base DynIn algorithm can distinguish only between convergent and divergent cells for one solution at a time. Initially only the cells of the solutions themselves are known; however, with each solution examined, the number of already explored cells increases, leading to a much faster computation. In particular, one of the more time consuming tasks of the original `DynIn` algorithm was exactly the identification of other solutions, which instead is performed much more efficiently by the numerical continuation algorithm. If solutions are not identified by the continuation algorithm—for example in the case of detached branches—they will be discovered by the corrector algorithm, as illustrated in the case studies of Section 3.2. If this happens, our algorithm can numerically continue those branches as well, if the user request it, thus exploring the phase space in even more depth. However, it must be noted that the algorithm might overlook stable solutions, especially if they do not directly limit the LIM of the investigated steady state, or they appear in a parameter region, where no other stable solutions are known, and no analysis is initiated; nevertheless, in this case the solutions are not the most critical for the system integrity, and overlooking them is indeed computationally more efficient than analyzing them, from a purely dynamical integrity perspective.

Algorithm 1 Predictor-corrector continuation of dynamical integrity

Require: Continued solutions $S_{i,j}$ at parameter values λ_i, $i = 1 \dots N$, $j = 1 \dots m_i$
Ensure: LIM: $\mathrm{L}_{i,j}$ for each $S_{i,j}$
for $i = 1$ to N **do**
  Classify cells of all solutions: $C_{i,j} \leftarrow S_{i,j}$
  for $j = 1$ to m_i **do**
    Convergent and divergent cells:
    $c \leftarrow C_{i,j} \qquad d \leftarrow C_{i,k}, k \neq j$
    if $S_{i,j}$ is unstable **then**
      $\mathrm{L}_{i,j} \leftarrow 0$
    else if $S_{i-1,j}$ does not exist **then**
      $[\mathrm{L}_{i,j}, p_{i,j}, c] \leftarrow \mathrm{DynIn}(S_{i,j}, c, d)$
      $C_{i,j} \leftarrow c$
    else
    Prediction phase I
      Previous critical point: $x_0 \leftarrow p_{i-1,j}$
      while x is convergent **do**
        $x \leftarrow$ Trajectory from x_0
        $k{+}{+}$
        $x_0 \leftarrow x_0 + \frac{(x_0 - S_{i,j})(1+k)}{10}$
      end while
      $p_{i,j}^{(0)} \leftarrow x$
      $\mathrm{L}_{i,j}^{(0)} = ||x - S_{i,j}||$
    Prediction phase II
      Extrapolate $\mathrm{L}_{i,j}^{\mathrm{pr}}$ from $\mathrm{LIM}_{1 \text{ to } i-1,j}$
      Compute first initial condition:
      $x_{IC} \leftarrow S_{i,j} + \mathrm{L}_{i,j}^{\mathrm{pr}} \cdot \frac{p_{i-1,j} - S_{i-1,j}}{||p_{i-1,j} - S_{i-1,j}||}$
    Correction phase
      $\mathrm{pr} \leftarrow [p_{i,j}^{(0)}, \mathrm{L}_{i,j}^{(0)}, x_{IC}]$
      $[\mathrm{L}_{i,j}, p_{i,j}, c] \leftarrow \mathrm{DynIn}(S_i, \mathrm{pr}, c, d)$
      $C_{i,j} \leftarrow c$
      if New solution found: $S_{i,j+1}$ **then**
        Continue $S_{i,j+1}$ for λ_i to λ_N
      end if
    end if
  end for
end for

This concludes the outline of the developed continuation algorithm, which, for better understanding, is provided as a pseudocode in Algorithm 1. There, we denote the steady states as $S_{i,j}$, the parameter values as λ_i, the LIM values as $L_{i,j}$, the critical points with $p_{i,j}$ and the convergent cells of each solution as $C_{i,j}$, with i being the index of the parameter values, and j being the index of the coexisting solutions for a parameter set. The results of the prediction (pr) are the initial LIM $L^0_{i,j}$, the point from which it was computed $p^0_{i,j}$, and the first initial condition x_{IC}, while c and d are the input convergent and divergent cells to DynIn.

The algorithm will be tested on three different nonlinear systems in Section 3, to prove its efficiency and potential limitations in the parametric study of system robustness.

2.2 Basin of attraction identification

Describing the robustness of a system through a scalar value like the LIM has great advantages when it comes to computational effectiveness. However, it fails to provide detailed information about the system's global dynamics, which, in case of specific parameter sets, can be required. Thus, the identification of the BoA becomes a necessity, regardless of its computational costs. Additionally, to analyze specific parameter sets in detail, computing the BoA, or at least sections of it, can also serve as a way to validate the LIM estimation of the continuation algorithm. Therefore, we included a feature in the classical DynIn that allows the user to compute sections of the BoA, and from that, the exact LIM value.

The developed algorithm defines a grid of initial conditions on a selected phase plane section. From each initial condition trajectories are initiated. However, for characterizing the trajectories the algorithm does not wait for them to reach one of the steady-state solutions, but exploits the cell subdivision used in the classical DynIn framework, i.e., if a trajectory reaches an already classified cell, the simulation is interrupted. This allows for a significant reduction in computational time, especially in large dimensional systems. In fact, only few trajectories will have to reach the steady state. We note that the size of the grid of initial conditions affect the resolution of the BoA, while its accuracy is related to the cell dimension, and the two are independent from each other. In some sense, the approach uses ideas of the cell mapping method, as each cell is interpreted as a system's state; however, it does not require to investigate all the cells, which would lead to insurmountable memory issues for large dimensional systems.

An exemplification result is illustrated in Figure 4a, where the BoA of the Duffing oscillator, discussed later in Section 3.1, is presented. In this case, the cell resolution is 10 times greater than the one of the grid (100 points in each direction), leading to accurate results in a reasonable time. Additionally, Figure 4b shows that the computational time increases only moderately with the increased cell resolution. Moreover, if we only increase the number of the cells, while keeping their size constant, the computational time increase is even less significant.

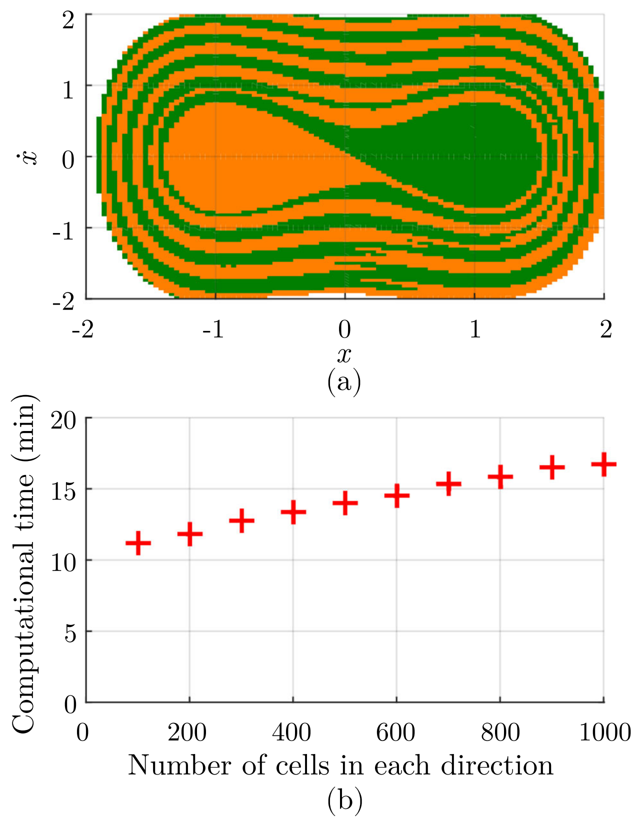


Fig. 4 A time section of the BoA computed for the Duffing oscillator ($\zeta = 0.05$, $F = 0.02$, $\omega = 1.1$) and the computational time as a function of the cell discretization's resolution

We remark that, for systems of dimension 3, the computation can be further improved by plotting also intersections of the trajectories with the section under study. However, this cannot be done for systems of larger dimension. For continuous system of dimension 2, BoA boundaries are fully defined by separatrices [43], usually making the computation trivial. The algorithm can be applied to larger dimensional sections; however, large computational cost and difficulty of visualization makes it rarely useful. It should be noted that our approach makes parallelization of the computation process more complicated, as new trajectories use information of previous ones. In view of using GPUs, this can be an important drawback. Still, a compromise can be reached, by performing groups of simulations in parallel and proceeding group by group.

2.3 Dynamical integrity assessment of chaotic attractors

Strange attractors are relatively common in nonlinear systems. While usually avoiding such solutions is the goal, in some instances, like for thermal pulse combustors [44], or fluid mixers [45], keeping the systems in a chaotic regime can be beneficial from an engineering standpoint. In these cases, assessing the robustness of the attractors is important to ensure adequate operation. For such problems, the generalized cell mapping method is often applied [18, 46]; however, unaffordable memory requirements still limit its widespread use [47]. Hence, we aim to enhance the set of algorithms in DynIn, to compute the dynamical integrity of chaotic solutions, without expensive hardware requirements.

The idea behind the algorithm is similar to the one employed in the case of periodic solutions [24]. Let us first consider the case of periodically excited systems. First a simulation is initiated from a user provided point of the chaotic attractor, generating a relatively long trajectory which cover most of the areas touched by the chaotic attractor. Points of the trajectory are collected according to the different phases of the excitation force, obtaining stroboscopic Poincaré sections of the attractor. These points and the cells they lie in are then classified as convergent.

Utilizing all the collected points for the LIM computation would be computationally too expensive. In fact, for each diverging trajectory, in each phase, it is required to compute the minimal distance between any of these points and any point of the trajectory. Hence, we represent the attractor by a subset of these points in each time section, which are chosen such that they homogeneously cover each section of the chaotic attractor. Details of the algorithm for the point selection are provided in Appendix A. Then, we use only these sampled points for estimating the LIM and selecting the next initial condition after a trajectory's convergence has been determined.

Apart from this, the procedure is the same as for periodic orbits [24], with the difference that for each considered time phase we have a set of point and not a single one. The LIM is estimate as the minimal distance between any of the selected points and any point of diverging trajectories, computed for each phase. An example of a Poincaré section of a chaotic attractor, obtained for the system in Section 3.1, is represented in Fig. 5. While the black dots (5000 points) mark all points initially computed for the section of the chaotic attractor, only red points (40 points) are used for the LIM computation. The purple point indicates the competing periodic solution. Green and orange regions are the BoA of the two solutions, respectively, and the blue cross identifies the critical point of the BoA boundary.

The sampling points' ability to represent the attractor even with such low numbers for the LIM estimation is clearly visible from Fig. 5. This is further proven by the low sensitivity of the estimated LIM value to variations of the sampling point number, as illustrated in Table 1.

In the case of autonomous systems, there is no need to organize the points marking the chaotic attractor according to Poincaré sections, similarly to the case of limit cycle oscillations studied in Section 3.2, as described in [24]. Also in this case, a small number of homogeneously spaced points of the attractor are used for the LIM computation. Although this case is not covered by the case studies presented in Section 3, it does not present any technical difficulties with respect to chaotic attractors ensuing in periodically excited system, as the ones studied in Section 3.1.

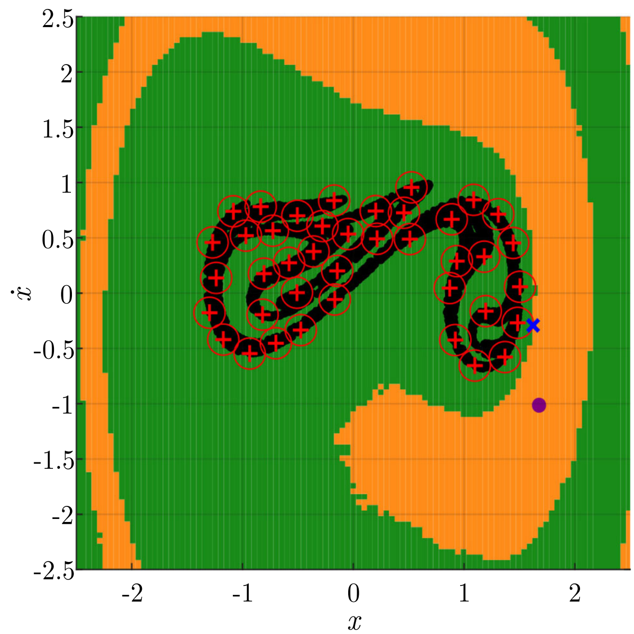


Fig. 5 The BoA (green region), the sampling points (red crosses) and the critical point (blue cross) of the chaotic attractor (black dots) found in the forced Duffing oscillator, computed by the BoA identification algorithm (Sec. 2.2), for parameters: $\zeta = 0.05$, $F = 0.24$, $\omega = 1.1$

Table 1 Effect of the number of sampling points used for chaotic attractors on the LIM value

Sampling points	20	40	60	80	100
LIM	0.259	0.246	0.246	0.244	0.244

3 Case studies

In order to evaluate the capabilities of the developed algorithm, we tested it on three different dynamical systems. The analyses were mostly carried out on periodic solutions, as their algorithmic treatment presents more difficulties than fixed points. However, also more challenging cases, such as chaotic solutions, were considered. First, we tested the algorithm on a non-autonomous system, namely a harmonically forced Duffing oscillator with a negative linear stiffness term. The goal of this analysis was, on the one hand, to compare the performance—in terms of computational time and accuracy—of the proposed continuation algorithm against that of the original algorithm in [23, 24], without continuation. On the other hand, we tried to assess the ability of the method to process a relatively large amount of attractors at the same time. The second analyzed system consists of a self-excited autonomous system, namely a Van der Pol-Duffing oscillator coupled with a nonlinear vibration absorber. Here, the ability of the method to assess several limit cycles at the same time was again examined, and the engineering relevance of parameter studies regarding robustness was demonstrated. Finally, in order to show how the method behaves when applied to realistic, moderately large dimensional systems, and to further reinforce the legitimacy of such analysis in engineering practice, the synchronous motion of a self-synchronizing vibrating screen model was analyzed. For all of the analyses a single core of a computer with an Intel i9-9900, 3.10GHz CPU and 32GB RAM was used.

3.1 Forced Duffing oscillator with negative linear stiffness

The forced Duffing oscillator, which is a non-autonomous, externally excited system, often plays the role of a trial system in scientific studies due to its simple governing equations and rich dynamics. This is especially true when the linear stiffness term is considered to be negative, which can be described by the following equation of motion [48].

$$\ddot{x} + 2\zeta\dot{x} - x + x^3 = F\cos(\omega t) \tag{1}$$

Previous studies regarding the dynamical integrity of similar systems also exist [40, 48–50], making it an optimal test subject for our analysis. A typical bifurcation diagram of the investigated periodic orbits, varying the forcing amplitude (F), can be observed in Fig. 1. In this instance, as well as during the whole investigation of the system, the damping coefficient is fixed at $\zeta = 0.05$; thus, from now on, only the other parameters will be specified. In Fig. 1, the saddle-node bifurcations, computed by the numerical continuation routine, are denoted by *snA*, *snB*, and *snC*, while the period-doubling bifurcation is indicated by *pd*. The stable periodic orbits, visualized by blue lines, which exist symmetric to the origin in the phase space as well, are the in-well non-resonant (INR), the in-well resonant (IR), and the out-of-well resonant (OR) steady states, while the unstable solutions are marked by red dashed lines. For clearer understanding of the different solutions, they are also represented in the phase space in Fig. 6.

Overall, the algorithm discovers eight different periodic solutions, three of which are unstable. The dynamical integrity analysis of all stable solutions was performed in a single run of the continuation algorithm. This enables a significant reduction in computational time, as information is

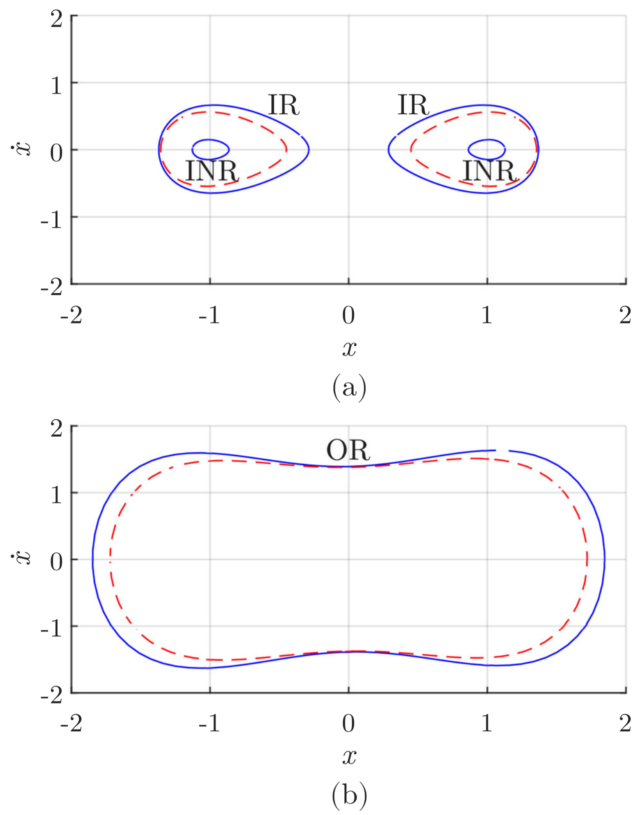


Fig. 6 Existing periodic solutions of the forced Duffing oscillator in Eq. (1) for parameters: (a) $F = 0.1$, $\omega = 1.1$. (b) $F = 0.3$, $\omega = 1.1$. Solid lines: stable solution; dashed lines: unstable solutions

shared between the analysis of the different coexisting solutions. The obtained results can be seen in Fig. 7, where the LIM was calculated for four solutions (two INR and two IR), but due to the symmetry only two of them are shown. For the analysis, the used parameter values are specified in Table 2, while their meaning is thoroughly discussed in [23, 24] and in `DynIn`'s tutorial. The continuation algorithm computed results for values of F between 0.004 and 0.2, with a step size of 0.004, so overall 50 parameter sets were analyzed. Obtained results are consistent with those presented in [40], manifesting the same sudden drop in dynamical integrity at the appearance of the INR steady states due to bifurcation *snA*. The OR solution is not shown on the diagrams due to the fact that when it is born from the *snC* bifurcation, there are no other stable solutions, thus the algorithm does not initiate the dynamical integrity analysis, hence it impossible to discover it.

Next, we computed dynamical integrity maps of all the stable solutions, considering simultaneous variations of the forcing frequency (ω) and amplitude (F). The continuation algorithm used the same parameter values as in the previous case (Table 2), but calculated the output for 60 forcing amplitude values between 0.01 and 0.6; this procedure was repeated for 31 values of ω between 0.8 and 1.4. The continu-

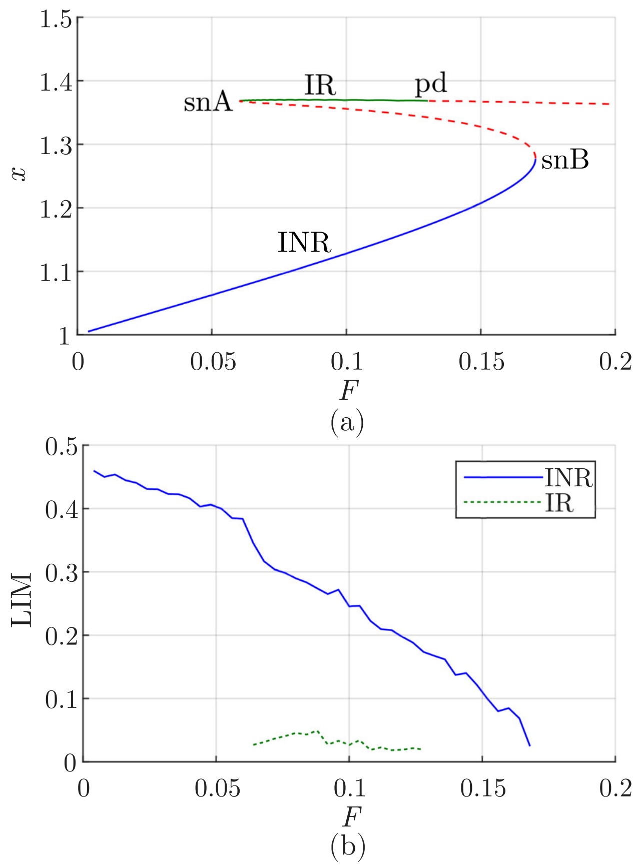


Fig. 7 (a) Bifurcation diagram of an in-well branch of the forced Duffing oscillator ($\omega = 1.1$); solid lines: stable solutions, dashed lines: unstable solutions. (b) Estimated LIM of the stable periodic solutions

Table 2 Algorithm parameters used by the corrector `DynIn`, detailed information about the parameters' meaning can be found in [23, 24]

discr	Cells in each direction	701
spaceboundary	Boundary of the phasespace	[-5,5][a]
Nt	Number of time sections	30
number_of_steps	Number of iteration steps	100

[a]For all coordinates

ation was always performed for increasing F values. Results are shown in Fig. 8a-c, where symmetric solutions are not illustrated twice for the sake of brevity. Nevertheless, further analysis with reversed direction of the continuation (not illustrated here) showed that the direction of the continuation has negligible effect on the results, with the LIM values being slightly less precise in the reversed direction, because the prediction step finds divergent trajectories easier, when the LIM decreases. Also, as a mean to make the diagrams more perceptible, in regions where only one globally stable solution exists, the LIM is capped at value 0.55. In all cases, the trend of the LIM is perfectly consistent with the bifurcations of the system.

In order to make the LIM map more comprehensive, with the algorithm developed for strange attractors described in Section 2.3, we analyzed the out-of-well chaotic attractor of the system. This solution exists in the region defined by the bifurcation *snB*, where the chaotic attractor inherits the BoA of the two INR orbits, the boundary crisis *cr* of the in-well chaotic attractors, which are born at the end of the period-doubling cascade, and the homoclinic bifurcation *hb*, where it disappears [48]. Even in this case, the results (Fig. 8d) are consistent with the bifurcations, which were computed numerically. The region where this chaotic attractor is the sole stable solution is again capped to the LIM value of 0.55. The LIM analysis of the chaotic solutions did not use the continuation scheme, since it presents some technical difficulties related to the algorithms used by continuation methods for chaotic solutions [51, 52]. Additionally, we note that the system presents periodic windows within the chaotic region, related to phase locking, and in-well chaotic attractors [48]. However, their LIM analysis is omitted here, since they exist in very narrow parameter spaces and their practical relevance is minimal.

In the following, we compare the performance of the algorithm proposed here, with the one presented in [23, 24], using the classical `DynIn` algorithm, i.e. without predictor-corrector continuation. For the diagrams in Fig. 8, 100 iterations of the corrector were used; however, the main advantage of our continuation algorithm is the possibility of reducing the iteration numbers needed to achieve accurate results. Therefore, the performance—in terms of computational time and accuracy of the estimated LIM value—was compared for either 20, 40 or 100 iterations. Reference LIM values were obtained from the identification of the whole BoA from an initial condition grid of 100 points in each direction, while 30 time sections were considered for each period of excitation. In each case, the analysis reported in Fig. 7 was repeated ten times, and the mean values of the LIM error and computational time over the examined forcing amplitude interval were attained. Results are summarized in Table 3. One can see that, while for large iteration numbers the precision difference is negligible, with a slightly faster computation with the continuation method, there is a clear advantage of using continuation when a rapid dynamical integrity analysis is needed, resorting to a low number of iterations. In the latter case, both the computational time, as well as the error in the results are significantly lower with the proposed continuation algorithm than with the classical approach.

However, it must be mentioned that even with the continuation, the LIM error might not be negligible for very low number of iterations. This is due to the fact that, when the basin boundary is fractal, like in the presence of the in-well chaotic attractors, the prediction of the critical point becomes much less reliable, thus requiring more iterations for the cor-

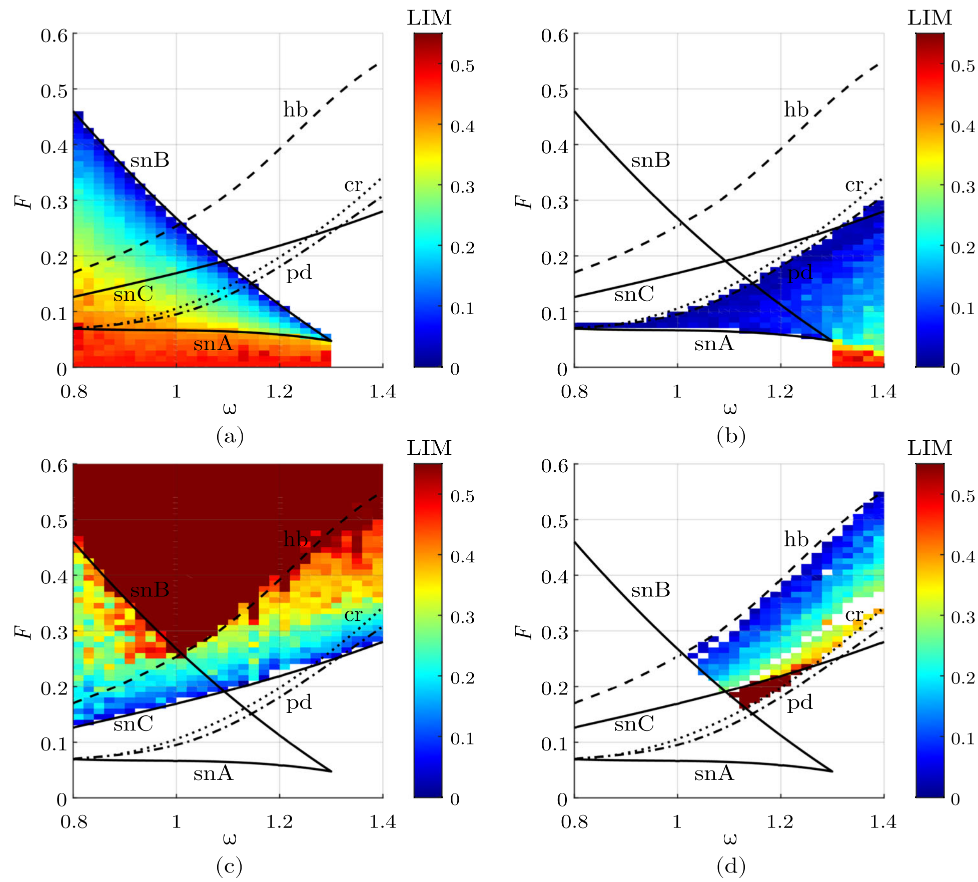


Fig. 8 Robustness maps of the stable periodic orbits of the forced Duffing oscillator: (a) INR. (b) IR. (c) OR. (d) Out-of-well chaotic attractor

Table 3 Comparison between error of LIM results and computational time between the continuation algorithm and the `DynIn` toolbox

Analysis type	INR LIM error (%)	Time (s)	IR LIM error (%)	Time (s)
Continuation, 20 iterations	1.63	1.515	11.13	1.002
No continuation, 20 iterations	232.45	2.634	181.13	2.498
Continuation, 40 iterations	1.03	3.496	4.72	2.539
No continuation, 40 iterations	130.26	5.392	124.25	4.932
Continuation, 100 iterations	0.69	15.308	1.03	13.506
No continuation, 100 iterations	0.71	16.687	1.56	15.481

rector. This is clearly observable in Fig. 9, where a sudden increase in the LIM error is noticeable after the IR solution goes through the period-doubling bifurcation. It is easy to see, however, that even in this region the continuation method still performs significantly better than the classical `DynIn`. This is further highlighted in Figure 10, where the two analyses, which produced the smallest LIM during the investigation in Table 3 for parameter values where the basin boundaries are fractal, are compared with the critical time section of the BoA.

From an engineering practice perspective, where typically a quick parametric analysis is needed, our continuation method can provide, in very short time, results, which are both qualitatively and quantitatively significant, and describe the evolution of the LIM effectively. Another important advantage of the continuation algorithm is that it can assess all periodic solutions simultaneously, while the classical `DynIn` toolbox could only analyze one solution at a time, reducing the amount of setup time required for the parametric analysis.

3.2 Van der Pol-Duffing oscillator coupled with a nonlinear vibration absorber

In order to test the continuation algorithm on the limit cycles of an autonomous system, a vibration mitigation problem was selected, specifically the Van der Pol-Duffing oscillator (VdPD) coupled with a nonlinear tuned vibration absorber

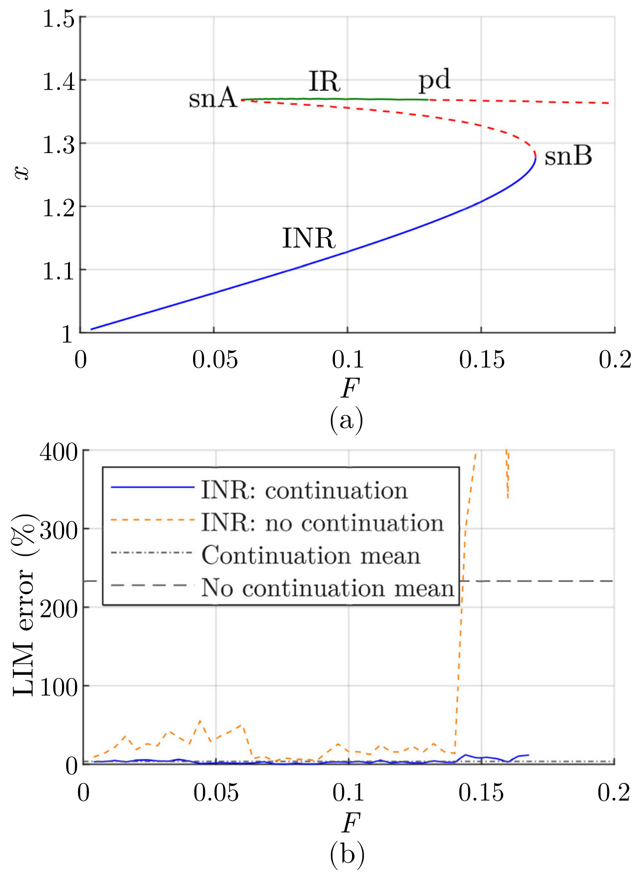


Fig. 9 Comparison of the error in the computed LIM values of the INR attractor between the continuation algorithm and the classical `DynIn` for 20 iterations ($\omega = 1.1$): (a) Bifurcation diagram. (b) LIM error diagram

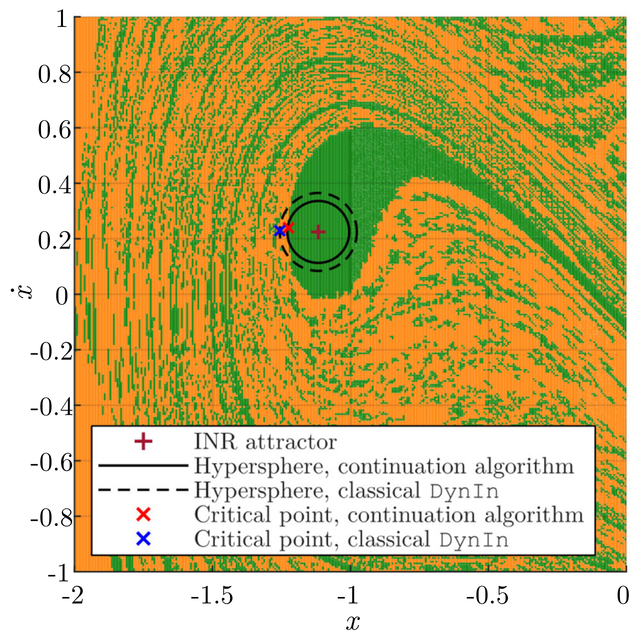


Fig. 10 Comparison of LIM results in the critical time section, when the BoA boundary is fractal for the INR attractor. The BoA section was computed using the algorithm in Sect. 2.2, with 200 initial conditions in each direction. Parameters are: $F = 0.15$, $\omega = 1.1$

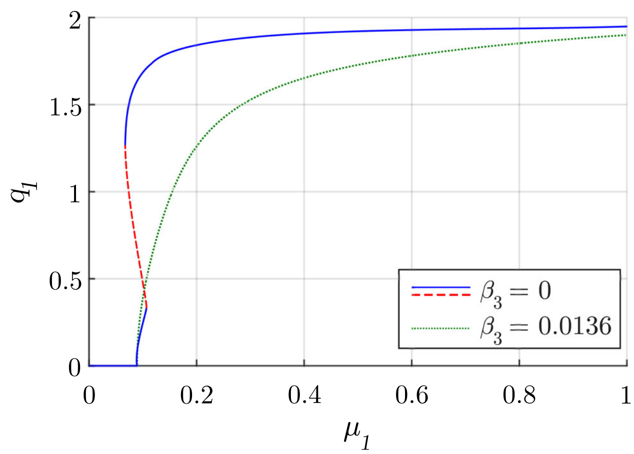


Fig. 11 Bifurcation diagrams of the system in Eq. (2)-(3). Solid lines: stable solutions, dashed lines: unstable solutions. ($\mu_2 = 0.12, \alpha_3 = 0.3$, $\varepsilon = 0.05$ and $\gamma = 0.985$)

(NLTVA), studied in [41]. The system was chosen due to its engineering relevance in the study of limit cycle oscillations, which can often lead to safety risks [6]. Another reason was its two degrees of freedom, which would imply that traditional methods to assess robustness, especially over an interval of parameters, are challenging from a computational cost perspective. The equations of motion derived in [41] are the following.

$$m_1\ddot{q}_1 + c_1(q_1^2 - 1)\dot{q}_1 + k_1 q_1 + k_{\mathrm{nl1}} q_1^3 + \\ + c_2\dot{q}_\mathrm{d} + k_2 q_\mathrm{d} + k_{\mathrm{nl2}} q_\mathrm{d}^3 = 0, \quad (2)$$

$$m_2\ddot{q}_2 - c_2\dot{q}_\mathrm{d} - k_2 q_\mathrm{d} - k_{\mathrm{nl2}} q_\mathrm{d}^3 = 0, \quad (3)$$

where q_1, m_1, c_1, k_1 and k_{nl1} are the oscillator's displacement, mass, damping and the coefficients of the linear and cubic springs, respectively. While q_2, m_2, c_2, k_2 and k_{nl2} denote the same for the NLTVA, and $q_\mathrm{d} = q_1 - q_2$. To make the investigation more universal and following [41], the following dimensionless parameters were introduced to characterize the self-excitation term and the nonlinear stiffness of the NLTVA:

$$\mu_1 = \frac{c_1}{2\sqrt{k_1 m_1}}, \quad \beta_3 = \frac{k_{\mathrm{nl2}} m_1}{k_1 m_2}, \\ \mu_2 = \frac{c_2}{2\sqrt{k_2 m_2}}, \quad \alpha_3 = \frac{k_{\mathrm{nl1}}}{k_1}, \quad (4) \\ \varepsilon = m_2/m_1, \quad \gamma = \frac{\omega_{\mathrm{n2}}}{\omega_{\mathrm{n1}}} = \sqrt{\frac{k_2 m_1}{k_1 m_2}}.$$

In the original study about this system [41], the objective was threefold. The linear absorber parameters (γ and μ_2) were tuned to postpone the Hopf bifurcations to the largest

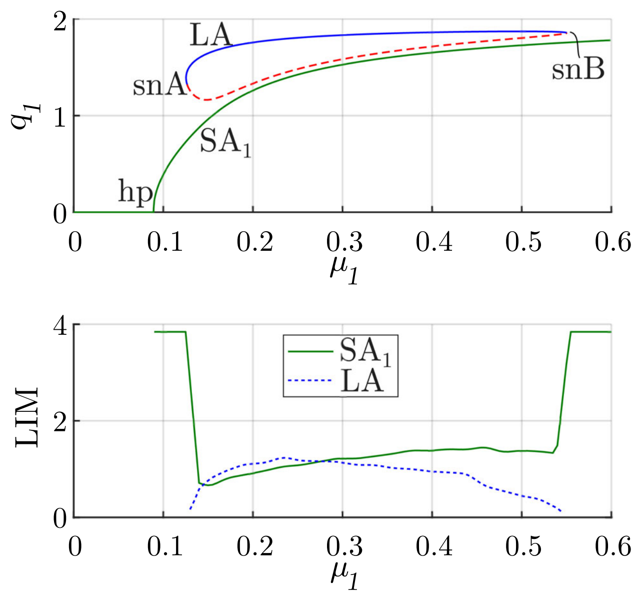


Fig. 12 The behavior of the VdPD-NLTVA system, which lead to the discovery of the isola: (a) Bifurcation diagram, where snA and snB mark saddle-node bifurcations and hp the Hopf bifurcation, with the unstable solutions depicted by the dashed line. (b) LIM diagram. ($\mu_2 = 0.12$, $\alpha_3 = 0.3$, $\varepsilon = 0.05$, $\gamma = 0.985$ and $\beta_3 = 0.0136$)

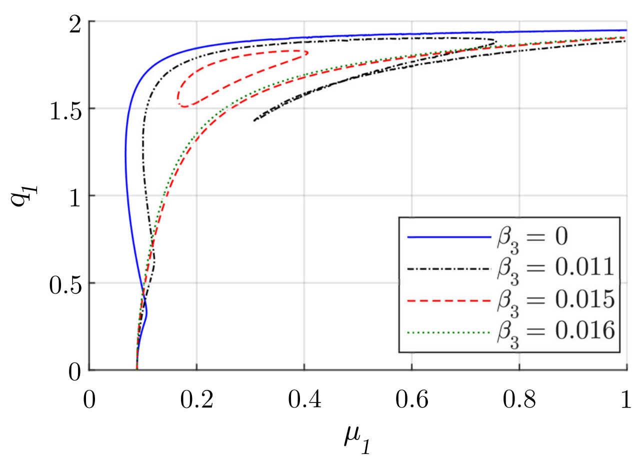


Fig. 13 Bifurcation diagram of the VdPD-NLTVA system along μ_1 and β_3. ($\mu_2 = 0.12$, $\alpha_3 = 0.3$, $\varepsilon = 0.05$, $\gamma = 0.985$)

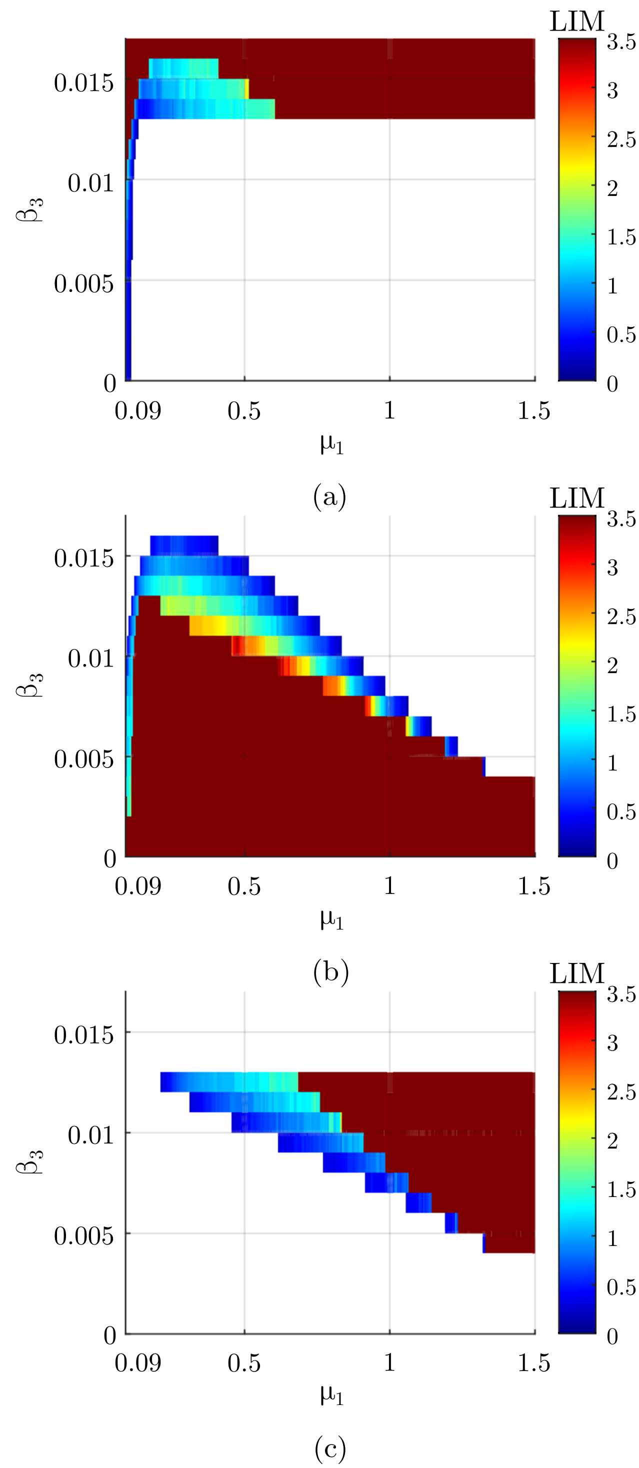


Fig. 14 LIM diagram of the VdPD-NLTVA system along μ_1 and β_3: (a) SA$_1$. (b) LA. (c) Another smaller amplitude vibration, which appears from a fold bifurcation (SA$_2$), see Fig. 13. ($\mu_2 = 0.12$, $\alpha_3 = 0.3$, $\varepsilon = 0.05$, $\gamma = 0.985$)

possible μ_1 value; then, the nonlinear absorber stiffness (β_3) was chosen such that the Hopf bifurcation is supercritical and the emerging branch of periodic solutions has amplitude as small as possible, without other undesired solutions present. After analytically identifying the optimal parameter tuning, validation was mostly done through a classical numerical continuation in `MatCont` [35], as can be seen in Fig. 11.

From Fig. 11, it seems that the chosen parameter values can effectively achieve the sought result, as no undesired solution is visible for $\beta_3 = 0.0136$. However, once the dynamical integrity continuation algorithm was applied to the system's desired small amplitude vibrations (SA$_1$), we observed a sudden drop in the LIM for a given parameter range. This led us to the discovery of an isolated branch of large amplitude periodic solutions (LA), visible in Fig. 12,

Table 4 Algorithm parameters used by the corrector DynIn, detailed information about the parameters' meaning can be found in [23, 24]

discr	Cells in each direction	701
spaceboundary	Boundary of the phasespace	[-10,10][a]
Nt	Points of limit cycle	50
number_of_steps	Number of iteration steps	100

[a]For all coordinates

which undermines the performance of the NLTVA. This finding clearly highlights the importance of global dynamics and dynamical integrity assessment for nonlinear system analysis.

This result motivated further investigating of the bifurcation diagram for different values of β_3, with the outcomes illustrated in Fig. 13. According to our results, for low β_3 values (black line, $\beta_3 = 0.011$) the branch of periodic solutions bands, forming a sort of peninsula, which then detaches from the main branch for larger β_3 values (red line, $\beta_3 = 0.015$). This detached branch then shrinks for increasing β_3, until it eventually disappears (green line, $\beta_3 = 0.016$).

This rather involute bifurcation scenario makes the dynamical integrity investigation more intriguing and potentially informative. Results of the analysis, carried out with the algorithm discussed above, are presented in Fig. 14. For all analyses, the parameters for the computation are specified in Table 4. The simulation time step of $\mathtt{Ts} = 0.1$ time units was applied. The LIM was calculated with a step size of 0.005 in the μ_1 direction, between the values 0.09, point of the Hopf bifurcation, and 1.5. This procedure was repeated for 18 values of β_3 between 0 and 0.017, when the isolated branch has already disappeared. The areas where a solution was globally stable had their LIM values capped to 3.5 for better visualization, the same way as it was done in Section 3.1. From an engineering perspective, these results suggest that the optimal β_3 value is not 0.0136, as indicated in [41], but 0.016 or larger.

These results are again proof that our algorithm is able to handle several branches of steady-state solutions at the same time, without the need for user intervention. We note that, for the system under study, in some cases a single branch underwent up to four bifurcations for variations of the single parameter μ_1, which highlight the challenging character of the task. The findings also further promote the necessity of robustness assessment in risk mitigation analyses, where the goal is to ensure sufficiently safe operation.

3.3 Self-synchronizing vibrating screen

In this last case study, we test the continuation algorithm on a realistic industrial problem. The main obstacle engineers face in the case of such systems is their moderately large dimensionality, which complicates analysis, especially regarding dynamical integrity, where typically the so-called curse of dimensionality dominate computational and memory cost [2]. Hence, our main aim here is to prove the effectiveness of the continuation algorithm in conducting robustness assessment over multiple parameters on a 5 degrees of freedom self-synchronizing vibrating screen model, which was chosen due to our previous work on it [53]. The mechanical model is represented on Fig. 15, with the generalized coordinates shown in red.

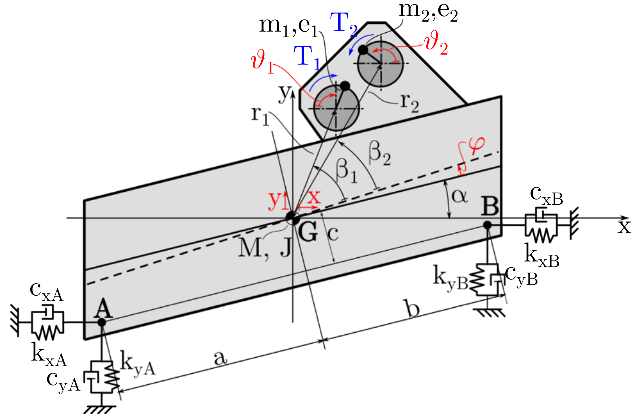


Fig. 15 Mechanical model of a self-synchronizing vibrating screen

The equations of motion of the system and the meaning of each parameter can be found in Appendix B, where the equations were developed for a DC motor drive, with the supplied voltages being the input variables. The parameter values used later are also shown in Appendix B, in Tables 7 and 8.

These machines exhibit a very interesting behavior, where even if the unbalanced rotors are not directly connected, given the right parameters, they will synchronize their angular velocities during operation, creating a translational excitation on the sieve body, which can be used to power horizontal screening machinery. The criterion for synchronization and the stability of the synchronous motion have already been extensively studied in scientific literature [54–58]. However, these results only examined the local stability of the synchronous motion, which does not always guarantee global stability, meaning that large enough perturbations can force the machine into the undesirable non-synchronous motion, making robustness assessment a necessity. The stability map of the system, derived numerically through a sweep-up and sweep-down procedure of the voltage difference between the

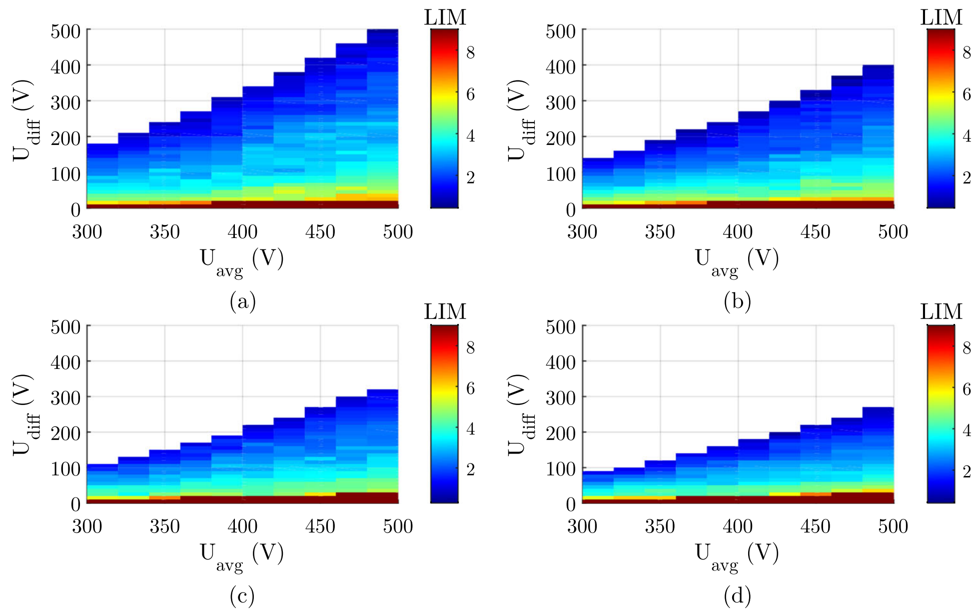


Fig. 16 Dynamical integrity maps of the self-synchronizing vibrating screen, as a function of the mean supplied voltage and the voltage difference between rotors, for friction coefficients: (a) $c_{1,2} = 1.2$ Nms/rad. (b) $c_{1,2} = 1.6$ Nms/rad. (c) $c_{1,2} = 2.0$ Nms/rad. (d) $c_{1,2} = 2.4$ Nms/rad. Other coefficients are marked in Tables 7 and 8

Table 5 Algorithm parameters used by the corrector `DynIn`, detailed information can be found in [23, 24]

`discr`	Cells in each direction	701
`Nt`	Points of limit cycle	10
`number_of_steps`	Number of iteration steps	100

rotors U_{diff}, exhibits a large bistable region, for parameters previously considered safe [53].

Since the synchronous motion is a periodic solution of the system [55], the continuation algorithm could again be applied, in order to run parametric analysis aiming to optimize parameters for robustness against external perturbations. These are visible in Fig. 16. The continuation of the solution branch was done in the direction of U_{diff} for 57 values between 0 V and 560 V, for 25 mean voltage (U_{avg}) values between 300 V and 540 V. This was repeated for four friction coefficient values of the rotors. The algorithm parameters used for the computation, are indicated in Table 5. The boundary of the examined phase space was selected to be $[-0.1, 0.1]$ m and $[-5, 5]$ m/s for the x- and y- directional displacements and velocities, $[-0.5, 0.5]$ rad and $[-5, 5]$ rad/s for the φ swing angle and angular velocity. The limits of the ϑ_i, which are the rotational angles of the rotors, were $[0, 2\pi]$ rad, and the limits of their angular velocities were $[-20, 20]$ rad/s, around the synchronous angular velocity $\overline{\omega}_m$ computed during the continuation of the solution branch. The time step of each numerical simulation was also chosen to be $\mathtt{Ts} = 2\pi/(\mathtt{Nt}\overline{\omega}_m)$, where `Nt` is the number of sampled points for the limit cycle.

The results in Fig. 16 are in line with those attained by simply using the classical `DynIn`, presented in [53]. However, the computational time was reduced by 21.8% on average for each parameter set investigated with 100 iterations, while LIM results were smaller by 7.485%. A small number of iteration would significantly enhance the difference in the computational time. Nevertheless, the relatively large dimension of the system suggests to use a higher number of iterations than for the systems studied previously.

From the point of view of the system safety, results illustrate that by increasing the friction coefficient we lose linearly stable area for the synchronous motion; however, the globally stable region remains approximately the same (capped to LIM value 9 in the figure). This suggests that the increase of the friction coefficient does not reduce operational safety, considering that the bistable region is unfit for prolonged usage, because of its low dynamical integrity.

In this case, since the system is ten dimensional, the computation of the whole BoA is unfeasible even with our BoA identification method, which limits the possibility of validating the results. However, since sections of the BoA can be computed, comparison can be made between the LIM and the BoA section, which is estimated to be the critical one, as seen in Fig. 17. Here, ϑ is the average phase angle of the rotors, and 2ξ is the phase difference between them, since the rotational angle of the rotors can be assumed as $\vartheta_i = \vartheta \pm \xi$ [57]. The BoA is projected onto the section, where the only nonzero coordinates are the 2ξ phase difference, its angular velocity and the mean angular velocity of the rotors, which is fixed at the synchronous angular velocity: $\vartheta = \overline{\omega}_m$. The simulations were initiated from an initial condition grid that had 75 elements in both directions.

It must be mentioned that we considered solutions (marked by a purple dots) with 2π shifts in the phase difference physically identical; however, we only compute the LIM

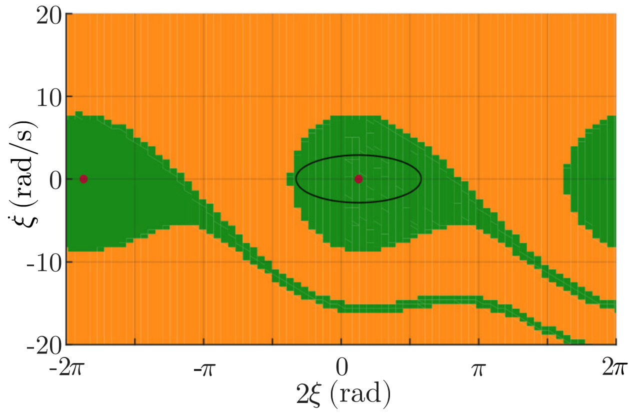


Fig. 17 Basin of attraction section of the synchronous motion (purple dots) of the vibrating screen, with the hypersphere of convergence (black circle), for parameters $U_{\text{avg}} = 400$ V, $U_{\text{diff}} = 100$ V and $c_{1,2} = 1.6$ Nms/rad. Other coefficients are marked in Tables 7 and 8, and other initial conditions are all set to zero

for the solution closest to zero phase difference. Comparing LIM results with the BoA section shows great agreement, as the black circle (marking the LIM) is fully inside the green area. The algorithm generates overestimates of the true LIM; in this case, the black circle does not touches the BoA boundary because the critical point is not included in the plotted sections. Indeed, in this case, our algorithm produced a more precise LIM estimate than the one obtainable by the BoA section. We also note that the section of the hypersphere of convergence, with the LIM as radius, resembles an ellipse, because distances are normalized differently in each direction acknowledging the anisotropy of the phase space. Further detailing of this issue can be found in [23].

4 Conclusions

In this study, for the first time, we proposed a predictor-corrector based continuation algorithm for the computation of the dynamical integrity of steady-state solutions. The method leverages the `DynIn` toolbox [23, 24], which directly computes the local integrity measure (LIM) of a solution, bypassing the full computation of the full basins of attraction (BoA). Our algorithm introduces a prediction step in the procedure, which exploits information about the dynamical integrity of nearby parameter sets to generate an accurate first guess about the LIM, which is then corrected through the same algorithm as in [23, 24]. Coupling this scheme with a standard pseudo-arclength continuation algorithm for steady-state solution identification, our procedure can rapidly generate comprehensive robustness maps, estimating the LIM of all relevant stable solutions of a system for a large range of parameter values. Additionally, all the computation can be performed in a single run of the algorithm, making it particularly practical. The algorithm was further augmented with the possibility of assessing the dynamical integrity of chaotic solutions, an operation that was never done before, without computing the whole BoA. Moreover, the original algorithm in [23, 24] was extended for the rapid computation of BoA sections, exploiting its framework of cell-subdivision.

The algorithm was tested on three qualitatively different dynamical systems; namely, a harmonically forced Duffing oscillator with negative linear stiffness, a Van der Pol-Duffing oscillator coupled with a nonlinear vibration absorber, and a self-synchronizing vibrating screen. The first two systems are both small dimensional, but one is non-autonomous and the other one is autonomous, presenting, therefore, different challenges, while the third one is an industrial-level larger dimensional system. The major challenges of the first systems are the coexistence of numerous stable solutions, the presence of chaotic attractors and fractal BoA, which were all correctly addressed by the proposed algorithm. The precision of the results, together with computational speed of the algorithm, was compared to the LIM calculated from the whole BoA, and the classical `DynIn` method. It was found that the prediction step drastically reduces the number of iteration needed by the corrector, while keeping precision relatively high. It was also proven that the algorithm can process more than one solution for a given parameter set, and use the convergence data acquired for other steady states during the dynamical integrity approximation.

Another test was done on a Van der Pol-Duffing oscillator coupled with a nonlinear vibration absorber. Here, we demonstrated the ability of the method for limit cycles of autonomous systems. The robustness assessment also led to the discovery of unknown isolated steady states, completely overlooked by previous studies on the same system [41], where analytical and traditional continuation methods were implemented.

Finally, we proved the ability of the method in case of moderately large dimensional problems, by studying a 5-DoF self-synchronizing vibrating screen. We were able to compute the robustness diagrams with significantly less computational costs than alternative methods. Our results also suggested that larger friction coefficients of the driving rotors result in approximately the same globally stable region when the supplied voltages differ between driving motors, even though the locally stable area decreases.

In conclusion, the algorithm was proven to reduce computational cost, while keeping high precision for robustness assessment parametric analysis. It was also able to assess several steady states at the same time, further reducing the need for user intervention, and computational and operational time.

In terms of limitations, we note that the algorithm uses only part of the information obtained for one parameter set for a neighboring one. Exploiting more extensively collected data—for example, convergence properties of all explored cells—could lead to a more advanced continuation algorithm. However, as sudden changes in the phase portrait are hard to anticipate, any prediction regarding the system's global dynamics might be misleading; this causes significant challenges for further improving the method.

Additionally, although we were able to address a relatively large system for global analysis standards (5-DoF), the analysis of large finite element models, encompassing hundreds or thousands of DoF, remain prohibitive with the current algorithm, and requires resorting to model reduction techniques [26]. In this respect, the combination of parametric model reduction with the proposed algorithm might reveal computationally extremely effective.

In terms of phase space exploration, our algorithm—as also the classical `DynIn`—utilizes only the position of the trajectories, neglecting their velocity and acceleration of convergence; as recently proved by a series of studies [59–62], this information can be extremely valuable for dynamical integrity analysis. Incorporating this information into our algorithm might significantly reduce computational time and improve accuracy, facilitating a smart exploration of the phase space.

Nevertheless, we note that all these limitations are not intrinsic in the proposed algorithm, but represent possible developments, which will be explored in future studies.

Appendix A Sampling algorithm for strange attractors

In order to homogeneously cover the chaotic attractor or its time sections with sampling points, the points are determined the following way, shown in Fig. 18 and Table 6. First, the centroid of all the points of the attractor is determined. Then the farthest point from it is computed, which would be our first sampling point. Following this, we calculate the farthest point from all the previous sampling points and the centroid as well, by maximizing its minimum distance from them, until the predefined number of sampling points is reached, which is 10 in the illustrated case. The centroid, while not included among the sampling points, still is a part of the distance calculations for all of them.

As the number of sampling points increases, the homogeneity of the points can be increased. However, using too many sampling points would defeat the algorithm's purpose, so it should be avoided.

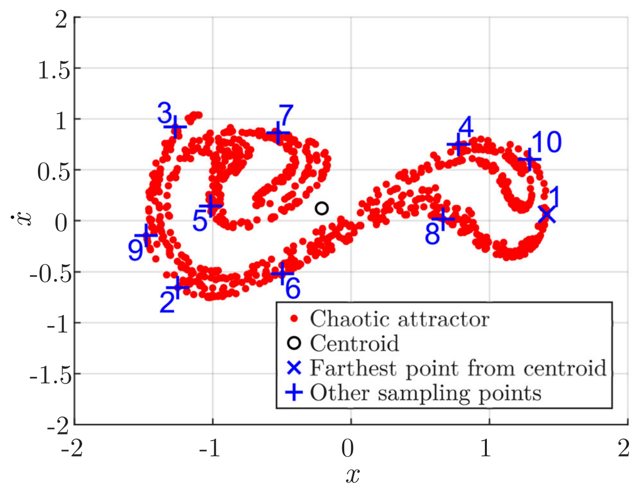


Fig. 18 Sampling of the chaotic attractor of the system discussed in Section 3.1 ($\zeta = 0.05$, $F = 0.24$, $\omega = 1.1$)

Table 6 Distance of the first 4 sampling points from each other and the minimum

Points	1	2	3	4
centroid	1.587	1.379	1.323	1.094
1		2.763	2.820	0.939
2			1.576	2.468
3				2.054
minimum	1.587	1.379	1.323	0.939

Appendix B Equations of motion of the vibrating screen

$$
\begin{aligned}
&(M + m_1 + m_2)\ddot{x} + (c_{xA} + c_{xB})\dot{x} \\
&+ [c(c_{xA} + c_{xB})\cos(\alpha + \varphi) \\
&+ ac_{xA}\sin(\alpha + \varphi) + bc_{xB}\sin(\alpha + \varphi)]\,\dot{\varphi} \\
&+ (bk_{xB} - ak_{xA})\cos(\alpha + \varphi) \\
&+ c(k_{xA} + k_{xB})\sin(\alpha + \varphi) \\
&+ (ak_{xA} - bk_{xB})\cos(\alpha) \\
&+ c(k_{xA} + k_{xB})\sin(\alpha) + (k_{xA} + k_{xB})x = \\
&= m_1 r_1 \dot{\varphi}^2 \cos(\alpha + \beta_1 + \varphi) \\
&+ m_2 r_2 \dot{\varphi}^2 \cos(\alpha + \beta_2 + \varphi) \\
&+ m_1 r_1 \ddot{\varphi} \sin(\alpha + \beta_1 + \varphi) \\
&+ m_2 r_2 \ddot{\varphi} \sin(\alpha + \beta_2 + \varphi) \\
&+ e_1 m_1 \cos(\vartheta_1)\dot{\vartheta}_1^2 - e_2 m_2 \cos(\vartheta_2)\dot{\vartheta}_2^2 \\
&+ e_1 m_1 \sin(\vartheta_1)\ddot{\vartheta}_1 - e_2 m_2 \sin(\vartheta_2)\ddot{\vartheta}_2 \qquad \text{(B1)} \\
&(M + m_1 + m_2)\ddot{y} + (c_{yA} + c_{yB})\dot{y} \\
&+ \big[c(c_{yA} + c_{yB})\sin(\alpha + \varphi)
\end{aligned}
$$

$$\begin{aligned}
&- ac_{yA}\cos(\alpha+\varphi) + bc_{yB}\cos(\alpha+\varphi)\big]\dot{\varphi} \\
&+ (bk_{yB} - ak_{yA})\sin(\alpha+\varphi) \\
&- c(k_{yA}+k_{yB})\cos(\alpha+\varphi) \\
&+ (ak_{yA} - bk_{yB})\sin(\alpha) \\
&+ c(k_{yA}+k_{yB})\cos(\alpha) + (k_{yA}+k_{yB})y = \\
&= m_1r_1\dot{\varphi}^2\sin(\alpha+\beta_1+\varphi) \\
&+ m_2r_2\dot{\varphi}^2\sin(\alpha+\beta_2+\varphi) \\
&- m_1r_1\ddot{\varphi}\cos(\alpha+\beta_1+\varphi) \\
&- m_2r_2\ddot{\varphi}\cos(\alpha+\beta_2+\varphi) \\
&+ (M+m_1+m_2)g \\
&+ e_1m_1\sin(\vartheta_1)\dot{\vartheta}_1^2 + e_2m_2\sin(\vartheta_2)\dot{\vartheta}_2^2 \\
&- e_1m_1\cos(\vartheta_1)\ddot{\vartheta}_1 - e_2m_2\cos(\vartheta_2)\ddot{\vartheta}_2
\end{aligned} \tag{B2}$$

$$\begin{aligned}
&(J+m_1r_1^2+m_2r_2^2)\ddot{\varphi} + \{c_{xA}(c\cos(\alpha+\varphi) \\
&+a\sin(\alpha+\varphi))^2 + c_{xB}(c\cos(\alpha+\varphi) \\
&-b\sin(\alpha+\varphi))^2 + c_{yA}(a\cos(\alpha+\varphi) \\
&-c\sin(\alpha+\varphi))^2 + c_{yB}(b\cos(\alpha+\varphi) \\
&+c\sin(\alpha+\varphi))^2 + [c_{xA}(c\cos(\alpha+\varphi) \\
&+ a\sin(\alpha+\varphi)) + c_{xB}(c\cos(\alpha+\varphi)) \\
&-c_{xB}(b\sin(\alpha+\varphi))]\,\dot{x} + \\
&+\big[-c\sin(\alpha+\varphi)) + c_{yB}(b\cos(\alpha+\varphi) \\
&+ c\sin(\alpha+\varphi)) - c_{yA}(a\cos(\alpha+\varphi)\big]\,\dot{y}\}\dot{\varphi} \\
&+k_{xA}(c\cos(\alpha+\varphi) - b\sin(\alpha+\varphi)) \\
&\cdot(c\sin(\alpha+\varphi) - a\cos(\alpha+\varphi)) \\
&+k_{xA}(c\cos(\alpha+\varphi) - b\sin(\alpha+\varphi)) \\
&\cdot(a\cos(\alpha) - c\sin(\alpha)) \\
&+k_{xB}(c\cos(\alpha+\varphi) + a\sin(\alpha+\varphi)) \\
&\cdot(c\sin(\alpha+\varphi) + b\cos(\alpha+\varphi)) \\
&-k_{xB}(c\cos(\alpha+\varphi) + a\sin(\alpha+\varphi)) \\
&\cdot(b\cos(\alpha) + c\sin(\alpha)) \\
&+k_{yA}(a\cos(\alpha+\varphi) - c\sin(\alpha+\varphi)) \\
&\cdot(a\sin(\alpha+\varphi) + c\cos(\alpha+\varphi)) \\
&-k_{yA}(a\cos(\alpha+\varphi) - c\sin(\alpha+\varphi)) \\
&\cdot(c\cos(\alpha) + a\sin(\alpha)) \\
&+k_{yB}(b\cos(\alpha+\varphi) + c\sin(\alpha+\varphi)) \\
&\cdot(b\sin(\alpha+\varphi) - c\cos(\alpha+\varphi)) \\
&+k_{yB}(b\cos(\alpha+\varphi) + c\sin(\alpha+\varphi)) \\
&\cdot(c\cos(\alpha) - b\sin(\alpha)) \\
&+ [(k_{xA}+k_{xB})c\cos(\alpha+\varphi) \\
&+ (k_{xA}a - k_{xB}b)\sin(\alpha+\varphi)]\,x \\
&+ \big[(k_{yA}+k_{yB})c\sin(\alpha+\varphi) \\
&- (k_{yA}a - k_{yB}b)\cos(\alpha+\varphi)\big]\,y = \\
&= m_1r_1\sin(\alpha+\beta_1+\varphi)\ddot{x} \\
&+m_2r_2\sin(\alpha+\beta_2+\varphi)\ddot{x} \\
&-m_1r_1\cos(\alpha+\beta_1+\varphi)\ddot{y} \\
&+m_2r_2\cos(\alpha+\beta_2+\varphi)\ddot{y} \\
&-e_1m_1r_1\Big[\sin(\alpha+\beta_1+\varphi-\vartheta_1)\dot{\vartheta}_1^2 \\
&+ \cos(\alpha+\beta_1+\varphi-\vartheta_1)\ddot{\vartheta}_1\Big] \\
&+e_2m_2r_2\Big[\sin(\alpha+\beta_2+\varphi+\vartheta_2)\dot{\vartheta}_2^2 \\
&- \cos(\alpha+\beta_2+\varphi+\vartheta_2)\ddot{\vartheta}_2\Big]
\end{aligned} \tag{B3}$$

$$\begin{aligned}
&m_1e_1^2\ddot{\vartheta}_1 - m_1e_1\sin(\vartheta_1)\ddot{x} + m_1e_1\cos(\vartheta_1)\ddot{y} \\
&+ m_1e_1r_1\cos(\alpha+\beta_1+\varphi-\vartheta_1)\ddot{\varphi} \\
&- m_1e_1r_1\sin(\alpha+\beta_1+\varphi-\vartheta_1)\dot{\varphi}^2 = T_1
\end{aligned} \tag{B4}$$

$$\begin{aligned}
&m_2e_2^2\ddot{\vartheta}_2 - m_2e_2\sin(\vartheta_2)\ddot{x} + m_2e_2\cos(\vartheta_2)\ddot{y} \\
&+ m_2e_2r_2\cos(\alpha+\beta_2+\varphi-\vartheta_2)\ddot{\varphi} \\
&- m_2e_2r_2\sin(\alpha+\beta_2+\varphi-\vartheta_2)\dot{\varphi}^2 = T_2
\end{aligned} \tag{B5}$$

In the case of direct current motors the parameter of the driving torque ($i = 1, 2$) are:

$$T_i = \frac{K_t}{R_a}U_i - \left(\frac{K_tK_E}{R_a} + c_i\right)\dot{\vartheta}_i \tag{B6}$$

Where the parameters M and m_i are the mass of the screen body and the eccentric rotors, e_i is the eccentricity of the rotors, respectively, and J is the mass moment of inertia of the main body. Parameters a, b, c, α, r_i and β_1 denote the geometry of the machine according to Figure 14. Furthermore, c_x, c_y and k_x, k_y refer to the damping and stiffness coefficients of the mounting springs. In case of the DC motor drive, K_t, K_E, and R_a are the torque constant, voltage constant and the armature resistance, respectively, while U_i is the voltage supplied to the motors, and c_i is the resistance coefficient of viscous friction. The motion of the sieve is characterized by the displacement of the center of mass G in x- and y-directions, the swing angle of the machine body φ, and the phase angles of the rotors ϑ_1 and ϑ_2.

The values used for each parameter during the investigation of the robustness, can be seen in Tables 7 and 8.

Table 7 Parameters of the DC motors

n_m	(-)	2
R_a	(Ω)	4.494
K_t	(Nm/A)	3.224
K_E	(V/rad/s)	4.207
$c_{1,2}$	(Nms/rad)	1.6

Table 8 Parameters of the screen body

$m_{1,2}$	M	$e_{1,2}$	J	α	a	b	c
(kg)	(kg)	(m)	(kgm^2)	(°)	(m)	(m)	(m)
300	25000	0.2	60000	0	2	2	0.3
r_1	r_2	β_1	β_2	k_x	k_y	c_x	c_y
(m)	(m)	(°)	(°)	$(\frac{kN}{mm})$	$(\frac{kN}{mm})$	$(\frac{kNs}{m})$	$(\frac{kNs}{m})$
0.25	0.25	180	0	2	5	40	80

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by M. Sz., and supervised by G.H. The first draft of the manuscript was written by M.Sz. and all authors reviewed and commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Budapest University of Technology and Economics. The project was supported by the Doctoral Excellence Fellowship Programme (DCEP), which is funded by the National Research Development and Innovation Fund of the Ministry of Culture and Innovation and the Budapest University of Technology and Economics.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

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

References

1. Soliman, M.S., Thompson, J.M.T.: Integrity measures quantifying the erosion of smooth and fractal basins of attraction. J. Sound Vib. **135**, 453–475 (1989). https://doi.org/10.1016/0022-460X(89)90699-8
2. Lenci, S., Rega, G. (eds.): Global Nonlinear Dynamics for Engineering Design and System Safety. Springer, New York (2019). https://doi.org/10.1007/978-3-319-99710-0
3. Menck, P.J., Heitzig, J., Marwan, N., Kurths, J.: How basin stability complements the linear-stability paradigm. Nat. Phys. **9**, 89–92 (2013). https://doi.org/10.1038/nphys2516
4. Thompson, J.M.T., Soliman, M.S.: Fractal control boundaries of driven oscillators and their relevance to safe engineering design. In: Proc. R. Soc. Lond. A, vol. 428, pp. 1–13. (1990). https://doi.org/10.1098/rspa.1990.0022
5. Lenci, S., Rega, G.: Load carrying capacity of systems within a global safety perspective. Part II. Attractor/basin integrity under dynamic excitations. Int. J. Non-Linear Mech. **46**(9), 1240–1251 (2011). https://doi.org/10.1016/j.ijnonlinmec.2011.05.021
6. Horvath, H.Z., Takacs, D.: Stability and local bifurcation analyses of two-wheeled trailers considering the nonlinear coupling between lateral and vertical motions. Nonlinear Dyn. **107**, 2115–2132 (2022). https://doi.org/10.1007/s11071-021-07120-9
7. Ji, P., Kurths, J.: Basin stability of the Kuramoto-like model in small networks. Eur. Phys. J. Spec. Top **223**, 2483–2491 (2014). https://doi.org/10.1140/epjst/e2014-02213-0
8. Verstraelen, E., Dimitriadis, G., Rossetto, G.D.B., Dowell, E.H.: Two-domain and three-domain limit cycles in a typical aeroelastic system with freeplay in pitch. J. Fluids Struct. **69**, 89–107 (2017). https://doi.org/10.1016/j.jfluidstructs.2016.11.019
9. Fröhlich, F., Sejnowski, T.J., Bazhenov, M.: Network bistability mediates spontaneous transitions between normal and pathological brain states. J. Neurosci. **30**(32), 10734–10743 (2010). https://doi.org/10.1523/JNEUROSCI.1239-10.2010
10. Oakad, H.M., Zur, K.K.: On the snap-through buckling analysis of electrostatic shallow arch micro-actuator via meshless galerkin decomposition technique. Eng. Anal. Bound. Elem **134**, 388–397 (2022). https://doi.org/10.1016/j.enganabound.2021.10.007
11. Rega, G., Lenci, S.: Identifying, evaluating, and controlling dynamical integrity measures in non-linear mechanical oscillators. Nonlinear Anal. Theory Methods Appl. **63**(5), 902–914 (2005). https://doi.org/10.1016/j.na.2005.01.084
12. Levin, A.: An analytical method of estimating the domain of attraction for polynomial differential equations. IEEE Trans. Automat. Contr. **39**(12), 2471–2475 (1994). https://doi.org/10.1109/9.362845
13. Grinberg, I., Gendelman, O.V.: Boundary for Complete Set of Attractors for Forced-Damped Essentially Nonlinear Systems. J. Appl. Mech. **82**(5), 051004 (2015). https://doi.org/10.1115/1.4030045
14. Genda, A., Fidlin, A., Gendelman, O.V.: The level-crossing problem of a weakly damped particle in quadratic potential well under harmonic excitation. Nonlinear Dyn. **111**(22), 20563–20578 (2023). https://doi.org/10.1007/s11071-023-08875-z
15. Settimi, V., Rega, G.: Global dynamics and integrity in noncontacting atomic force microscopy with feedback control. Nonlinear Dyn, vol. 86, pp. 2261–2277. (2016). https://doi.org/10.1007/s11071-016-2620-9
16. Hsu, C.: A Theory of Cell-to-Cell Mapping Dynamical Systems. J. Appl. Mech. **47**(4), 931–939 (1980). https://doi.org/10.1115/1.3153816
17. Sun, J., Xiong, F., Schütze, O., Hernández, C.: Cell Mapping Methods. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-0457-6_6
18. Wang, X., Jiang, J., Hong, L.: Enhancing subdivision technique with an adaptive interpolation sampling method for global attractors of nonlinear dynamical systems. Int. J. Dynam. Control **8**, 1147–1160 (2020). https://doi.org/10.1007/s40435-020-00683-2
19. Belardinelli, P., Lenci, S.: An efficient parallel implementation of cell mapping methods for M DOF systems. Nonlinear Dyn. **86**, 2279–2290 (2016). https://doi.org/10.1007/s11071-016-2849-3
20. Stender, M., Hoffmann, N.: bSTAB: an open-source software for computing the basin stability of multi-stable dynamical systems. Nonlinear Dyn. **107**, 1451–1468 (2022). https://doi.org/10.1007/s11071-021-06786-5
21. Brzeski, P., Wojewoda, J., Kapitaniak, T., Kurths, J., Perlikowksi, P.: Sample-based approach can outperform the classical dynam-

ical analysis - experimental confirmation of the basin stability method. Sci. Rep. **7**, 6121 (2017). https://doi.org/10.1038/s41598-017-05015-7
22. Thompson, J.M.T.: Chaotic phenomena triggering the escape from a potential well. In: Proc. R. Soc. Lond. A, vol. 421, pp. 195–225. (1989). https://doi.org/10.1098/rspa.1989.0009
23. Habib, G.: Dynamical integrity assessment of stable equilibria: a new rapid iterative procedure. Nonlinear Dyn, vol. 106, pp. 2073–2096. (2021). https://doi.org/10.1007/s11071-021-06936-9
24. Patkó, D., Habib, G.: Dynamical integrity of periodic orbits: a computationally efficient approach. Nonlinear Dyn. **113**, 24641–24661 (2025). https://doi.org/10.1007/s11071-025-11424-5
25. Szaksz, B., Stepan, G., Habib, G.: Dynamical integrity estimation in time delayed systems: a rapid iterative algorithm. J. Sound Vib. **571**, 118045 (2024). https://doi.org/10.1016/j.jsv.2023.118045
26. Habib, G., Cenedese, M., Rega, G.: Spectral submanifold reduction of high dimensional nonlinear systems for dynamical integrity analysis. Nonlinear Dyn. **113**, 28983–29005 (2025). https://doi.org/10.1007/s11071-025-11668-1
27. Sharma, S., Coetzee, E.B., Lowenberg, M.H., Neild, S.A., Krauskopf, B.: Numerical continuation and bifurcation analysis in aircraft design: an industrial perspective. Philos. Trans. R. Soc. A **373**(2051), 20140406 (2015). https://doi.org/10.1098/rsta.2014.0406
28. Sracic, M.W., Allen, M.S.: Numerical Continuation of Periodic Orbits for Harmonically Forced Nonlinear Systems. In: Proulx, T. (ed.) Civil Engineering Topics, Volume 4. Conference Proceedings of the Society for Experimental Mechanics Series, vol. 4, pp. 51–69. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-9316-8_5
29. Nayfeh, A.H., Balachandran, B.: Applied Nonlinear Dynamics: Analytical, Computational, and Experimental Methods. John Wiley & Sons Ltd, Hoboken (2008)
30. Kuznetsov, Y.A.: Numerical Analysis of Bifurcations. In: Kuznetsov, Y.A. (ed.) Elements of Applied Bifurcation Theory. Applied Mathematical Sciences, pp. 569–655. Springer, New York (2023)
31. Keller, H.B.: Numerical Methods in Bifurcation Problems. Springer, Berlin (1987)
32. Seydel, R.: Practical Bifurcation and Stability Analysis: from Equilibrium to Chaos, 2nd edn. Springer, Berlin (1994)
33. Peeters, M., Viguié, R., Sérandour, G., Kerschen, G., Golinval, J.-C.: Nonlinear normal modes, Part II: Toward a practical computation using numerical continuation techniques. Mech. Syst. Signal Process. **23**(1), 195–216 (2009). https://doi.org/10.1016/j.ymssp.2008.04.003
34. Doedel, E.: Auto, Software for Continuation and Bifurcation Problems in Ordinary Differential Equations. Accessed 7 October 2025 (1996). http://indy.cs.concordia.ca/auto/
35. Goaverts, W., Kuznetsov, Y.A., Meijer, H.G.E.: MATCONT and CL_MATCONT: Continuation toolboxes in matlab. Accessed 7 October 2025 (2011). http://sourceforge.net/projects/matcont/
36. Schilder, F., Dankowitz, H.: Continuation Core and Toolboxes. Accessed 30 December 2025 (2009). https://sourceforge.net/projects/cocotools/
37. Guillot, L., Cochelin, B., Vergez, C.: Manlab: An interactive path-following and bifurcation analysis software. Accessed 30 December 2025 (2011). https://manlab.lma.cnrs-mrs.fr/spip/
38. Engelborghs, K.: DDE-BIFTOOL, a Matlab package for bifurcation analysis of delay differential equations. Accessed 30 December 2025 (2007). https://twr.cs.kuleuven.be/research/software/delay/ddebiftool.shtml
39. Govaerts, W., Kuznetsov, Y.A., Dhooge, A.: Numerical Continuation of Bifurcations of Limit Cycles in MATLAB. SIAM J. Sci. Comput. **27**(1), 231–252 (2005). https://doi.org/10.1137/030600746
40. Lenci, S.: Dynamical Integrity and Its Background. In: Piqueira, J.R.C., Mazzilli, N., Eduardo, C., Pesce, C.P., Franzini, G.R. (eds.) Lectures on Nonlinear Dynamics, pp. 301–339. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-45101-0_11
41. Habib, G., Kerschen, G.: Suppression of limit cycle oscillations using the nonlinear tuned vibration absorber. Proc. R. Soc. A 471, 20140976 (2015)https://doi.org/10.1098/rspa.2014.0976
42. Golubitsky, M., Stewart: Symmetric Chaos, vol. 200, pp. 241–274. Birkhäuser, (2002). https://doi.org/10.1007/978-3-0348-8167-8_9
43. Guckenheimer, J., Holmes, P.: Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields. Springer, New York (1983). https://doi.org/10.1007/978-1-4612-1140-2
44. In, V., Spano, M.L., Neff, J.D., Ditto, W.L., Daw, C.S., Edwards, K.D., Nguyen, K.: Maintenance of chaos in a computational model of a thermal pulse combustor. Chaos **7**(4), 605–613 (1997). https://doi.org/10.1063/1.166260
45. Shirmohammadi, F., Tohidi, A.: Mixing enhancement using chaos theory in fluid dynamics: Experimental and numerical study. Chem. Eng. Res. Des. **141**, 350–360 (2019). https://doi.org/10.1016/j.cherd.2018.04.007
46. Zou, H., Xu, J.: Improved generalized cell mapping for global analysis of dynamical systems. Sci. China Ser. E-Technol. Sci **52**, 787–800 (2009). https://doi.org/10.1007/s11431-009-0056-x
47. Li, Z., Jiang, J., Li, J.: A subdomain synthesis method for global analysis of nonlinear dynamical systems based on cell mapping. Nonlinear Dyn 95, 715–726 (2019) https://doi.org/10.1007/s11071-018-4592-4
48. Lenci, S., Rega, G.: Forced harmonic vibration in a Duffing oscillator with negative linear stiffness and linear viscous damping. In: Kovacic, I., Brennan, M. (eds.) The Duffing Equation: Nonlinear Oscillators and Their Behaviour, pp. 219–276. John Wiley & Sons Ltd, New York (2011)
49. Lenci, S., Rega, G.: Optimal Control of Nonregular Dynamics in a Duffing Oscillator. Nonlinear Dyn. **33**, 71–86 (2003)
50. Yang, S.: Parameter-induced fractal erosion of the safe basin in a softening Duffing oscillator. J. Vibroeng **18**(5), 3329–3336 (2016). https://doi.org/10.21595/jve.2016.17209
51. Gerlach, R., Ziessler, A., Eckhardt, B., Dellnitz, M.: A set-oriented path following method for the approximation of parameter dependent attractors. SIAM J. Appl. Dyn. Syst. **19**, 705–723 (2020). https://doi.org/10.1137/19M1247139
52. Glendinning, P.A., Simpson, D.J.W.: Robust chaos and the continuity of attractors. Trans. Math. Appl 4, (2020). https://doi.org/10.1093/imatrm/tnaa002
53. Szabó, M., Miklós, A., Habib, G.: Dynamical integrity of synchronous motion for self-synchronizing vibrating screens with two unbalanced exciting rotors. J. Sound Vib. **620**, 119459 (2026). https://doi.org/10.1016/j.jsv.2025.119459
54. Blekhman, I.I., Lavrov, B.P.: On an integral criterion for stability of motion. J. Appl. Math. Mech. **24**(5), 1416–1422 (1960). https://doi.org/10.1016/0021-8928(60)90122-2
55. Blekhman, I.I.: Synchronization in Science and Technology. ASME Press, New York, USA (1988)
56. Blekhman, I.I., Fradkov, A.L., Tomchina, O.P., Bogdanov, D.E.: Self-synchronization and controlled synchronization: general definition and example design. Math. Comput. Simul. **58**(4), 367–384 (2002). https://doi.org/10.1016/S0378-4754(01)00378-0
57. Zhao, C., Zhu, H., Wang, R., Wen, B.: Synchronization of two non-identical coupled exciters in a non-resonant vibrating system of linear motion. Part I: Theoretical analysis. Shock. Vib. **16**, 505–515 (2009). https://doi.org/10.3233/SAV-2009-0484
58. Zhao, C., Zhu, H., Wang, R., Wen, B.: Synchronization of two non-identical coupled exciters in a non-resonant vibrating system of linear motion. Part II: Numerical analysis. Shock. Vib. **16**, 517–528 (2009). https://doi.org/10.3233/SAV-2009-0485

59. J., Ghadami, G.P., Sanches, A., Epureanu, L., Michon, B.I., G.: Data-driven bifurcation analysis of experimental aeroelastic systems using preflutter measurements. AIAA J. **62**(5), 1906–1914 (2024). https://doi.org/10.2514/1.J063736
60. Habib, G.: Predicting saddle-node bifurcations using transient dynamics: a model-free approach. Nonlinear Dyn. **111**(22), 20579–20596 (2023). https://doi.org/10.1007/s11071-023-08941-6
61. Habib, G.: Unstable limit cycle estimation from small perturbations: a minimal-data-driven approach. Nonlinear Dyn. **113**, 1–16 (2025). https://doi.org/10.1007/s11071-025-10989-5
62. Kadar, F., Stepan, G., Habib, G.: Model-free fold bifurcation prediction from pre-bifurcation scenario: experimental validation through wheel shimmy vibrations. Nonlinear Dyn. **113**(10), 11247–11258 (2025). https://doi.org/10.1007/s11071-024-10686-9