

A fast sampling method for estimating the domain of attraction

Najafi, Esmaeil; Babuska, R.; Delgado Lopes, G.A.

DOI

[10.1007/s11071-016-2926-7](https://doi.org/10.1007/s11071-016-2926-7)

Publication date

2016

Document Version

Final published version

Published in

Nonlinear Dynamics: an international journal of nonlinear dynamics and chaos in engineering systems

Citation (APA)

Najafi, E., Babuska, R., & Delgado Lopes, G. A. (2016). A fast sampling method for estimating the domain of attraction. *Nonlinear Dynamics: an international journal of nonlinear dynamics and chaos in engineering systems*, 86(2), 823-834. <https://doi.org/10.1007/s11071-016-2926-7>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

A fast sampling method for estimating the domain of attraction

Esmaeil Najafi  · Robert Babuška · Gabriel A. D. Lopes

Received: 17 November 2015 / Accepted: 22 June 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Most stabilizing controllers designed for nonlinear systems are valid only within a specific region of the state space, called the domain of attraction (DoA). Computation of the DoA is usually costly and time-consuming. This paper proposes a computationally effective sampling approach to estimate the DoAs of nonlinear systems in real time. This method is validated to approximate the DoAs of stable equilibria in several nonlinear systems. In addition, it is implemented for the passivity-based learning controller designed for a second-order dynamical system. Simulation and experimental results show that, in all cases studied, the proposed sampling technique quickly estimates the DoAs, corroborating its suitability for real-time applications.

Keywords Domain of attraction · Lyapunov function · Optimization · Sampling method

E. Najafi (✉) · R. Babuška · G. A. D. Lopes
Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
e-mail: e.najafi@tudelft.nl

R. Babuška
e-mail: r.babuska@tudelft.nl

G. A. D. Lopes
e-mail: g.a.delgadolopes@tudelft.nl

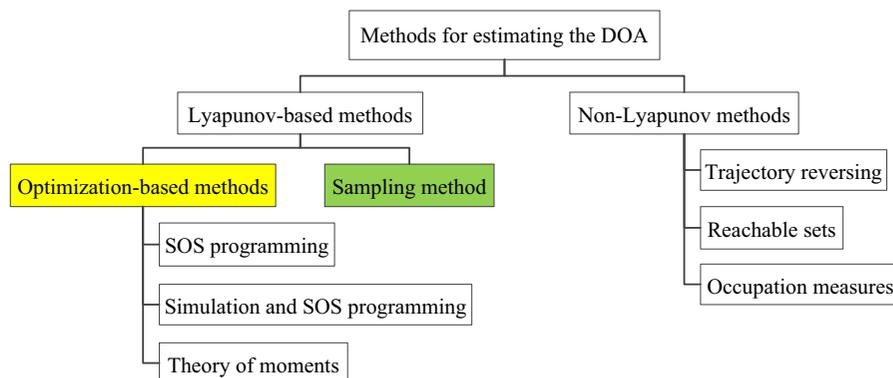
E. Najafi
Department of Mechatronic Systems Engineering,
Faculty of New Sciences and Technologies, University of Tehran,
Tehran 14399-57131, Iran

1 Introduction

The domain of attraction (DoA) of a stable equilibrium in a nonlinear system is a region of the state space from which each trajectory starts and eventually converges to the equilibrium itself. In the literature, the DoA is also known as the region of attraction or basin of attraction [1, 33]. The DoA of an equilibrium and its computation is of main importance in control applications. However, in most cases, computation of the DoA is quite costly. This paper aims to approximate the DoAs of nonlinear systems in real time by introducing a sampling approach.

Several techniques have been proposed in the literature to compute an inner approximation for the DoA [8], which can broadly be classified into Lyapunov-based and non-Lyapunov methods [11]. Lyapunov-based approaches include, for instance, sum of squares (SOS) programming [4], methods that apply both simulation and SOS programming [32], procedures that use theory of moments [13]. In this approach, first, a candidate Lyapunov function is chosen to show asymptotic stability of the system within a small neighborhood of the equilibrium. Next, the largest sublevel set of this Lyapunov function, in which its time derivative is negative definite, is computed as an estimate for the DoA [25]. Non-Lyapunov methods include, for instance, trajectory reversing [11, 24], determining reachable sets of the system [2], and occupation measures [14, 18]. Figure 1 illustrates a broad classification of the existing techniques for estimating the DoA.

Fig. 1 A broad classification of the existing techniques for estimating the DoA. This paper proposes a sampling approach and makes a comparison with optimization-based methods



Although Lyapunov-based techniques have been successfully implemented for estimating the DoAs of various nonlinear systems [8], there are still two main issues with using these approaches. The first is that most of the existing methods are limited to polynomial systems [12,31]. As such, in the case of non-polynomial systems, first, the equations of motion are approximated by using the Taylor's expansion and then the DoA is computed based on the approximated polynomial equations. The second is that the available methods are usually computationally costly and time-consuming which make them unsuitable for real-time applications [7].

This paper proposes a fast sampling approach for Lyapunov-based techniques to estimate the DoAs of various nonlinear systems. This method is computationally effective and is beneficial for real-time applications. In this procedure, once a candidate Lyapunov function is chosen, a sampling algorithm searches for the largest sublevel set of the Lyapunov function such that its time derivative is negative definite throughout the obtained sublevel set. The proposed sampling approach is applied to approximate the DoAs of several nonlinear systems, which have been already investigated in the literature, to validate its capability in comparison with the existing methods. In addition, we go beyond these examples and implement it to compute the DoAs of the passivity-based learning controller [28] designed for an inverted pendulum.

This paper is organized as follows. Section 2 reviews the process of estimating the DoAs of nonlinear systems using Lyapunov-based techniques. Section 3 describes the sampling approach and provides a comparison between the estimated DoAs computed by the sampling method and by the existing optimization-based methods. Section 4 illustrates the DoAs approxi-

imated for the controller learned for an inverted pendulum. Finally, Sect. 5 concludes the paper after a short discussion on the properties of the sampling algorithm.

2 Estimating the domain of attraction using Lyapunov-based methods

Consider the dynamical system

$$\dot{x} = f(x, u) \quad (1)$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state vector, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, and $f: \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^n$ is the system dynamics. For a specific state-feedback controller $\Phi(x)$ the closed-loop system is described by

$$\dot{x} = f(x, \Phi(x)) = f_c(x). \quad (2)$$

If x^* is a stable equilibrium of the closed-loop system and $x(t, x_0)$ denotes the solution of (2) at time t with respect to the initial condition, the DoA of controller Φ is defined by the set

$$\mathcal{D}(\Phi) = \left\{ x_0 \in \mathcal{X} : \lim_{t \rightarrow \infty} x(t, x_0) = x^* \right\}. \quad (3)$$

An analytical method to approximate the DoA is defined via Lyapunov stability theory as follows [6,16].

Theorem 1 [16] *A closed set $\mathcal{M} \subset \mathbb{R}^n$, including the origin as an equilibrium, can approximate the DoA for the origin of system (2) if:*

1. \mathcal{M} is an invariant set for system (2);
2. A positive definite function $V(x)$ can be found such that $\dot{V}(x)$ is negative definite within \mathcal{M} .

If the equilibrium is nonzero, without loss of generality, we can replace the variable x by $z = x - \bar{x}^*$, where \bar{x}^* is the nonzero equilibrium. As such, we can study

the stability of the associated zero equilibrium [1]. The conditions of Theorem 1 ensure that the approximated set \mathcal{M} is certainly contained in the DoA.

The choice of a candidate Lyapunov function is not a trivial task and the DoA approximation relies on the shape of the Lyapunov function’s level sets. A procedure to find an appropriate Lyapunov function has been proposed in [10], where gradient search algorithms are implemented to compute a candidate Lyapunov function. Moreover, using composite polynomial Lyapunov functions [29] and rational Lyapunov functions instead of quadratic ones might lead to better approximations, since these have a richer representation power (see, e.g., [9,34]). Quadratic Lyapunov functions restrict the estimates to ellipsoids which are quite conservative [30]. A rational Lyapunov function is written in the form

$$V(x) = \frac{N(x)}{D(x)} = \frac{\sum_{i=2}^{\infty} R_i(x)}{1 + \sum_{i=1}^{n-2} Q_i(x)} \tag{4}$$

where $R_i(x)$ and $Q_i(x)$ are homogeneous polynomials of degree i , which are constructed by solving an optimization problem [34]. The sublevel set $\mathcal{V}(c)$ of the Lyapunov function $V(x)$ is defined by

$$\mathcal{V}(c) = \{x \in \mathcal{X} : V(x) \leq c\}. \tag{5}$$

According to Theorem 1, any sublevel set of a candidate Lyapunov function that satisfies the locally asymptotic stability of the equilibrium can be an estimate for the DoA if the time derivative of the Lyapunov function is negative everywhere within the sublevel set. Since the largest sublevel set provides a more accurate estimate, the problem of approximating the DoA is converted to the problem of finding the largest sublevel set of a given Lyapunov function [15]. To attain the largest estimate for the DoA, one needs to find the maximum value $c \in \mathbb{R}$ for $\mathcal{V}(c)$ such that the computed set satisfies the conditions of Theorem 1.

Theorem 2 [8] *The invariant set $\mathcal{V}(c_*)$, which is a sublevel set of the Lyapunov function $V(x)$, is the largest estimate of the DoA for the origin of system (2) if*

$$\begin{cases} c_* = \max c \\ \text{s.t. } \mathcal{V}(c) \subseteq \mathcal{H}(x), \\ \mathcal{H}(x) = \{0\} \cup \{x \in \mathbb{R}^n : \dot{V}(x) < 0\}. \end{cases} \tag{6}$$

This can be approached as an optimization problem that has been solved by using SOS programming, methods that apply both simulation and SOS programming, and methods that use theory of moments. However, these techniques are typically restricted to systems

and Lyapunov functions described by polynomial equations. In this paper, we present an alternative approach using the sampling approach.

3 Sampling method for estimating the domain of attraction

The sampling approach presented in this paper has the same goal as the Lyapunov-based optimization approaches have: Find the largest sublevel set of a candidate Lyapunov function to approximate the DoA. We explicitly evaluate the conditions stated in Theorem 1 for a given Lyapunov function with respect to a randomly chosen state x_i . The level sets associated with the sample x_i with positive derivative of the Lyapunov function are discarded. We propose two sampling methods, memoryless and with a memory, designed to achieve tighter estimates.

3.1 Memoryless sampling

This method searches for the upper bound of the parameter c_* in (6). First, a state x_i is randomly chosen within \mathcal{X} or its user-defined subset and the conditions of Theorem 1 are checked for $V(x_i)$ and $\dot{V}(x_i)$. If these conditions are not satisfied, the upper bound of c_* , denoted \hat{c}_* , is decreased to the value $\hat{c}_* = V(x_i)$ and the sublevel set $\mathcal{V}(\hat{c}_*)$ is computed as an overestimation for the DoA. At the beginning of the algorithm, \hat{c}_* is initialized at $\hat{c}_* = \infty$. As the sampling proceeds for a large number of samples (n_s) throughout the state space, the value of \hat{c}_* converges to c_* from above and the obtained largest sublevel set $\mathcal{V}(\hat{c}_*)$ will be very close to $\mathcal{V}(c_*)$. Since this procedure just focuses on the upper bound of c_* , the achieved estimates are not tight enough and the condition of $\dot{V}(x) < 0$ may not be satisfied for some regions of the attained sublevel set as the computed value \hat{c}_* is actually larger than the real value c_* . This algorithm may exceptionally not exclude very small regions where $\dot{V}(x) \geq 0$ from the DoA approximated. However, the empirical evidence arising from extensive simulations suggests that, in practice, the proposed algorithm converges to the exact level set for a sufficiently large number of samples.

Based on the practical results, we found that this technique is very fast and its result is very close to the reported estimates in the literature for various classes of

systems. Moreover, it does not require computer memory to save the results computed since once a new value is computed for \hat{c}_* , its current value is replaced by the new value. Algorithm 1 summarizes this method for estimating the DoA of a given stable equilibrium.

Algorithm 1 Memoryless sampling method for estimating the DoA

Require: $V(x)$, $\dot{V}(x)$, n_s

```

1: Initialize  $\hat{c}_* = \infty$ 
2: for  $i = 1 : n_s$  do
3:   Pick a random state  $x_i$  within the state space
4:   if  $\dot{V}(x_i) \geq 0$  and  $V(x_i) < \hat{c}_*$  then
5:      $\hat{c}_* = V(x_i)$ 
6:   end if
7: end for
8: return  $\hat{c}_*$ 

```

As an example, consider a pendulum described by the following nonlinear dynamic equations

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\sin(x_1) - 0.5x_2 \end{cases} \quad (7)$$

where x_1 is the angle of the pendulum measured from the vertical axis and x_2 is the angular velocity. The state vector is defined by $x = [x_1 \ x_2]^T$. We use the sampling method with a uniform distribution to approximate the DoA of the stable equilibrium $x = (0, 0)$. To compute a candidate Lyapunov function, first the dynamic Eq. (7) are linearized around the equilibrium and then the candidate Lyapunov function is computed in the form $V(x) = x^T P x$, where P is the solution of the Lyapunov equation $A^T P + P A + Q = 0$ with the identity matrix Q . In this example, the candidate Lyapunov function is obtained as

$$V(x) = 2.25x_1^2 + x_1x_2 + 2x_2^2. \quad (8)$$

Figure 2 illustrates the evolution of \hat{c}_* of the sampling approach with $n_s = 500$ samples. The real value c_* for the candidate Lyapunov function (8), calculated by solving the optimization problem (6), is $c_* = 9.287$ and the value computed by our method is $\hat{c}_* = 9.702$.

3.2 Sampling with memory

This method updates both the lower and the upper bounds of c_* denoted \underline{c}_* and \bar{c}_* , respectively. Together,

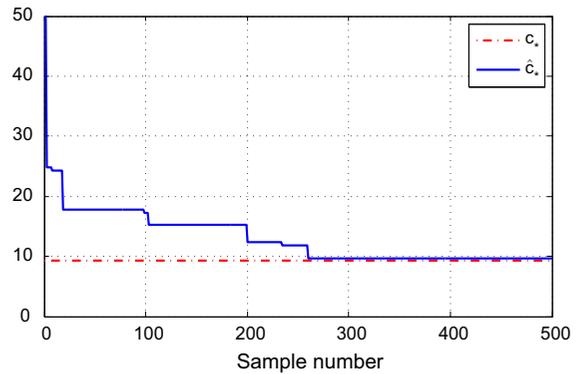


Fig. 2 Evolution of \hat{c}_* using the memoryless sampling method for the pendulum example

these bounds yield a more accurate estimate for the DoA. At the beginning of the algorithm, the lower bound of c_* is set to $\underline{c}_* = 0$ and its upper bound to $\bar{c}_* = \infty$. If for a randomly chosen state x_i we have $\dot{V}(x_i) < 0$ and $\underline{c}_* < V(x_i) < \bar{c}_*$, then the value of \underline{c}_* is replaced by the value of its associated Lyapunov function, that is $\underline{c}_* = V(x_i)$. Otherwise, if $\dot{V}(x_i) \geq 0$ and $V(x_i) < \bar{c}_*$, then the value of \bar{c}_* is replaced by $V(x_i)$. As the sampling proceeds, after a large number of samples, the value of \underline{c}_* increases, but not necessarily monotonically. Eventually it converges to c_* and the largest sublevel set $\mathcal{V}(\underline{c}_*)$ is obtained. Moreover, the value of \bar{c}_* monotonically decreases and converges to c_* from above.

When the conditions of Theorem 1 are satisfied for state x_i , the value of $V(x_i)$ is stored in an array as a possible estimate for c_* . This is required to guarantee that the approximated DoAs computed by the lower bound of c_* always verify the conditions of Theorem 1. This leads to tighter estimates. The array, denoted \mathcal{E} , contains 0 initially. The length of this array, without counting its initial element, is in the worst case $n_s - 1$. When $\dot{V}(x_i) < 0$ and $V(x_i) < \bar{c}_*$, the value of $V(x_i)$ is stored in an array \mathcal{E} as $\mathcal{V}(V(x_i))$ is a potential estimate for the DoA. In the case $\dot{V}(x_i) \geq 0$ and $V(x_i) < \bar{c}_*$, if $\underline{c}_* \geq \bar{c}_*$ then the algorithm looks for a new lower bound \underline{c}_* among the values stored in the array \mathcal{E} . The maximum value of \underline{c}_* is chosen from \mathcal{E} such that $\underline{c}_* < \bar{c}_*$. Selecting a previously stored lower bound satisfies the condition $\dot{V} < 0$ for the obtained sublevel set $\mathcal{V}(\underline{c}_*)$. In the worst-case scenario, $\underline{c}_* = 0$.

Although the sampling algorithm with memory is a conservative method, it may exceptionally overesti-

mate the DoA, for instance, when the region described by $\dot{V}(x) < 0$ is not simply connected. In such a case, the algorithm may not exclude small holes inside the region in which $\dot{V}(x) \geq 0$. A formal guarantee for convergence of this algorithm does not exist yet, but the empirical result attained from extensive simulations and experiments illustrates that the sampling technique converges to the exact level set for a sufficiently large number of samples, in practice. Algorithm 2 describes the sampling method with memory for estimating the DoA.

Algorithm 2 Sampling method with memory for estimating the DoA

```

Require:  $V(x), \dot{V}(x), n_s$ 
1: Initialize  $\underline{c}_* = 0, \bar{c}_* = \infty, \mathcal{E} = \{0\}$ 
2: for  $i = 1 : n_s$  do
3:   Pick a random state  $x_i$  within the state space
4:   if  $\dot{V}(x_i) < 0$  and  $V(x_i) < \bar{c}_*$  then
5:     store  $V(x_i)$  in  $\mathcal{E}$ 
6:     if  $V(x_i) > \underline{c}_*$  then
7:        $\underline{c}_* = V(x_i)$ 
8:     end if
9:   else if  $\dot{V}(x_i) \geq 0$  and  $V(x_i) < \bar{c}_*$  then
10:     $\bar{c}_* = V(x_i)$ 
11:    if  $\underline{c}_* \geq \bar{c}_*$  then
12:       $\underline{c}_* = \arg \max\{c \in \mathcal{E} : c < \bar{c}_*\}$ 
13:    end if
14:  end if
15: end for
16: return  $\underline{c}_*$ 
    
```

We apply this approach with a uniform distribution sampling to approximate the DoA for the equilibrium of the pendulum example. Figure 3 illustrates the values of the lower and upper bounds of c_* throughout the sampling process with 500 samples where $\underline{c}_* = 9.174$.

Figure 4 depicts the approximated DoA of the equilibrium. The black ellipsoid represents the DoA estimate with $\underline{c}_* = 9.271$, the dashed blue line, which determines the boundary of the light blue area, represents the region in which $\dot{V}(x) < 0$, and the arrows represent the system trajectories. If the trajectories start inside the DoA estimate, they certainly converge to the origin. The randomly chosen sampling states, which are 500 samples in this example, are represented by red points throughout the state space.

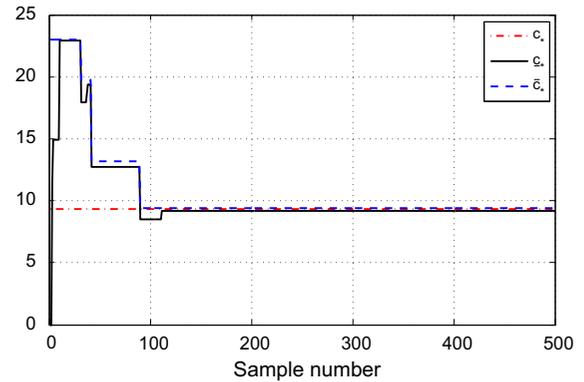


Fig. 3 Evolution of \underline{c}_* and \bar{c}_* using the sampling method with memory for the pendulum example

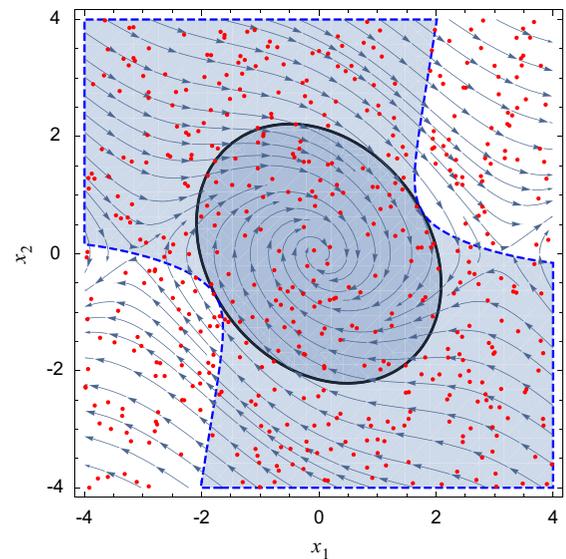


Fig. 4 Approximated DoA for the pendulum example using a uniform distribution for sampling. The *black ellipsoid* represents the DoA estimate, the *dashed blue line* (the boundary of the *light blue area*) represents the region in which $\dot{V}(x) < 0$, the *arrows* represent the system trajectories, and the *red points* represent the randomly chosen sampling states. (Color figure online)

3.3 Repeatability of the sampling method

To check the repeatability of the proposed sampling approach, we run various instances of the process of estimating the DoA for the equilibrium of the pendulum example. Figure 5 illustrates the mean value of \underline{c}_* and \bar{c}_* (i.e., $(\underline{c}_* + \bar{c}_*)/2$) and its standard deviation by a black line and green bars, minimum of \underline{c}_* and maximum of \bar{c}_* by blue dashed lines at each sample in a simulation where the sampling method runs 1000 iterations each

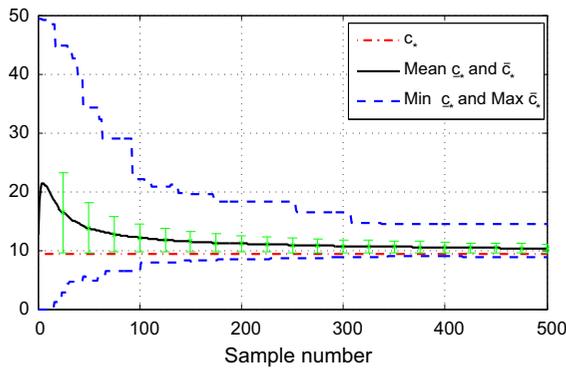


Fig. 5 Evolution of mean value of c_* and \bar{c}_* and its standard deviation, minimum value of c_* , and maximum value of \bar{c}_* for the sampling method in the pendulum example. The real value of c_* is represented by the *dotted red line*. (Color figure online)

with 500 samples. The real value of $c_* = 9.287$ is represented by a dotted red line. While sampling proceeds, the mean, minimum and maximum values converge to the real value of c_* and the value of the standard deviation decreases. These results validates the repeatability of the proposed sampling technique for this particular model.

3.4 Directed sampling

In the pendulum example, we used a uniform distribution for sampling the state space or its subset. However, if the structure of the level sets of the Lyapunov function are known, other distributions can be used to avoid sampling in areas of the state space which are already known not belong to the DoA. It is desirable to sample inside the largest level set found so far, specially in its boundary.

In general, sampling with an arbitrary distribution is a challenging problem. Two main approaches exist in the literature: rejection sampling and inverse transform sampling [3]. These techniques focus on sampling the relevant locations of the state space at the cost of computational complexity. In situations where evaluating a particular sample is costly (due to a complicated Lyapunov function or system dynamics), the extra cost incurred by sampling from a complex distribution may be negligible.

To test the trade-off between the speed of convergence and the computational cost, we have applied three different sampling approaches to the pendulum example (7). The uniform sampling on a fixed box

(a subset of the state space) is compared with uniform sampling mapped through polar coordinates to lie inside the largest found valid level set, and with exponential sampling mapped through polar coordinates to lie around the boundary of the largest found valid level set. Figure 6 illustrates the sampling points selected by the three types of distributions. The obtained data corroborate the hypothesis that different sampling leads to different convergence rates. Figure 7 presents the convergence statistics for 1000 iterations with 500 samples each. The exponential polar sampling converges the fastest and has the lowest variation between c_* and \bar{c}_* while converging. This can be explained by observing Fig. 6c that most of the samples are focused around the boundary of the level set. For this particular example, the cost of evaluating the Lyapunov function and its time derivative is low, but the computation time increases with the complexity of the sampling algorithm. Table 1 shows the average computation time of each sampling method with 500 samples, implemented in the Mathematica software on an Intel core i7 2.7 GHz microprocessor.

3.5 Sampling method versus optimization-based methods

Both the sampling and optimization-based methods require a candidate Lyapunov function for estimating the DoA. Table 2 represents six dynamical systems with quadratic Lyapunov functions selected from the literature. The dynamic equations of the first three examples are polynomial and the equations of the last three are non-polynomial. Examples E3 and E6 are third-order systems and the others are second-order systems. For each system, the maximum possible value of c_* computed by the sampling approach with 1000 samples is compared with the result of optimization-based methods, reported in the literature. The estimates attained by the sampling technique are very close to the estimates derived by optimization-based methods. In some cases, such as example E2, the result of the sampling procedure is even more accurate. The last column of Table 2 presents the simulation time for approximating the DoA of each system using the sampling approach, implemented in the MATLAB R2014a software on an Intel core i7 2.7 GHz microprocessor.

Similarly, Table 3 illustrates three dynamical systems with rational Lyapunov functions selected from

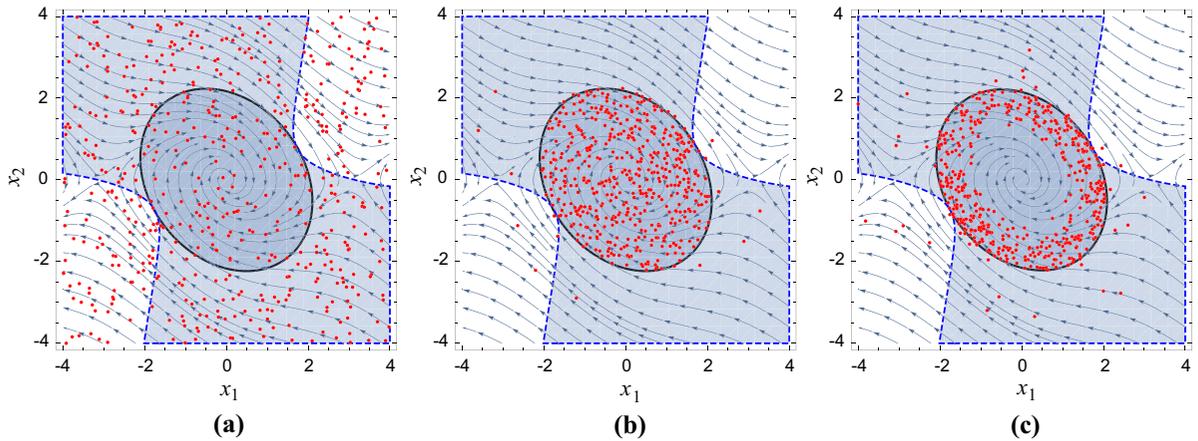


Fig. 6 Approximated DoAs for the pendulum example using a **a** uniform, **b** polar uniform, and **c** polar exponential distribution for sampling. In the plots, the *black ellipse* represents the DoA estimate, the *dashed blue line* represents the region in which

$\dot{V}(x) < 0$, the *arrows* represent the system trajectories, and the *red points* represent the randomly chosen sampling states. (Color figure online)

Fig. 7 Evolution of the mean value of \underline{c}_* and \bar{c}_* , minimum value of \underline{c}_* , and maximum value of \bar{c}_* for the sampling technique implemented for the pendulum example with a uniform, polar uniform, and polar exponential distribution. The sampling method runs 1000 iterations each with 500 samples. The real value of c_* is represented by a *dashed black line*

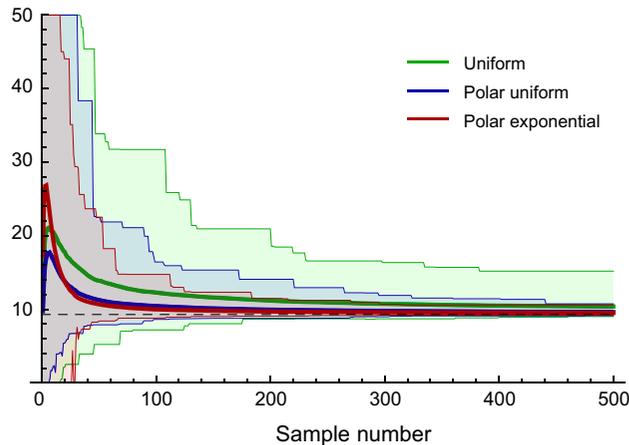


Table 1 Computation time statistics of the sampling methods with various distributions for estimating the DoA of the pendulum example

Sampling method	Time [ms]
Uniform in a box	7.4
Uniform in polar coordinates	17.1
Exponential in polar coordinates	27.4

the literature. Example E7 is a second-order polynomial system, E8 is a second-order non-polynomial system, and E9 is a third-order polynomial system. Table 4 presents their corresponding rational Lyapunov functions based on (4). The maximum possible value of

c_* obtained by the sampling approach with 1000 samples is compared with the result of optimization-based methods, reported in the literature. The result of this comparison validates the proposed sampling technique particularly for non-polynomial systems. The simulation time for approximating the DoA of each system using the sampling procedure is given in the last column of Table 3. Figure 8 depicts the approximated DoAs obtained by the sampling method for the origins of examples E1–E9.

Based on the obtained results, it is concluded that the proposed sampling approach is suitable for estimating the DoAs of both polynomial and non-polynomial systems. It is computationally effective and computes the DoA estimate considerably fast. Although the sampling

Table 2 Dynamical systems with quadratic Lyapunov functions

Example	Systems dynamics	Lyapunov function	Optimization (c_*)	Sampling (\underline{c}_*)	Time [ms]
E1 [12,30]	$\dot{x}_1 = -2x_1 + x_1x_2$ $\dot{x}_2 = -x_2 + x_1x_2$	$x_1^2 + x_2^2$	4.0804	4.112	6.6
E2 [12,34]	$\dot{x}_1 = -x_2$ $\dot{x}_2 = x_1 - x_2 + x_1^2x_2$	$1.5x_1^2 - x_1x_2 + x_2^2$	2.09	2.318	6.7
E3 [13,31]	$\dot{x}_1 = -x_1 + x_2x_3^2$ $\dot{x}_2 = -x_2 + x_1x_2$ $\dot{x}_3 = -x_3$	$x_1^2 + x_2^2 + x_3^2$	4.9188	4.971	8.4
E4 [5,27]	$\dot{x}_1 = -\frac{1}{4}x_1 + \ln(1 + x_2)$ $\dot{x}_2 = -\frac{3}{8}x_1 - \frac{1}{5}x_1x_2 + \left(\frac{1}{8}x_1 - x_2\right)\cos x_1$	$x_1^2 + x_2^2$	0.2737	0.278	8.3
E5 [7]	$\dot{x}_1 = x_2$ $\dot{x}_2 = -0.2x_2 + 0.81 \sin x_1 \cos x_1 - \sin x_1$	$x_1^2 + x_1x_2 + 4x_2^2$	0.6990	0.708	7.2
E6 [5]	$\dot{x}_1 = 1 + x_3 + \frac{1}{8}x_3^2 - \exp(x_1)$ $\dot{x}_2 = -x_2 - x_3$ $\dot{x}_3 = -x_2 - 2x_3 - \frac{1}{2}x_1^2$	$x_1^2 + x_2^2 + x_3^2$	2.655	2.887	8.6

Table 3 Dynamical systems with rational Lyapunov functions

Example	Systems dynamics	Optimization (c_*)	Sampling (\underline{c}_*)	Time [ms]
E7 [19,26]	$\dot{x}_1 = -x_2$ $\dot{x}_2 = x_1 - x_2 + x_1^2x_2$	5.3133	5.131	14.0
E8 [7,19]	$\dot{x}_1 = -x_1 + x_2 + 0.5(\exp(x_1) - 1)$ $\dot{x}_2 = -x_1 - x_2 + x_1x_2 + x_1 \cos x_1$	1.2251	1.218	14.6
E9 [13,19]	$\dot{x}_1 = -x_1 + x_2x_3^2$ $\dot{x}_2 = -x_2 + x_1x_2$ $\dot{x}_3 = -x_3$	1.320	1.318	16.6

method may offer less accurate estimates for the DoA at times, it is very useful for real-time applications. It is also beneficial for the control schemes applying the controllers' DoAs such as online sequential composition approaches [21–23].

4 Experimental results

Consider the inverted pendulum, shown in Fig. 9, which is modeled by the nonlinear differential equation

$$J\ddot{q} = mgl \sin(q) - \left(b + \frac{K^2}{R}\right)\dot{q} + \frac{K}{R}u \quad (9)$$

where q is the angle of the pendulum measured from the upright position, J is the pendulum inertia, m is the mass, l is the pendulum length, and b is the viscous mechanical friction. Moreover, K is the motor constant, R is the electrical motor resistance, and u is the control input in Volts which is saturated at ± 3 V. The state vector of the system is defined by $x = [q \ p]^T$ with $p = J\dot{q}$ the angular momentum. Table 5 presents the values of the physical parameters of the pendulum. These values have been found partly by measuring and partly estimated using nonlinear system identification.

The algebraic interconnection and damping assignment actor-critic (A-IDA-AC) algorithm, proposed in

Table 4 Rational Lyapunov functions for the systems of Table 3

Example	Numerator terms of the Lyapunov function (4)	Denominator terms of the Lyapunov function (4)
E7 [19,26]	$R_2(x) = 1.5x_1^2 - x_1x_2 + x_2^2$ $R_3(x) = 0$ $R_4(x) = -0.3186x_1^4 + 0.7124x_1^3x_2 - 0.1459x_1^2x_2^2 + 0.1409x_1x_2^3 - 0.03769x_2^4$	$Q_1(x) = 0$ $Q_2(x) = -0.2362x_1^2 + 0.31747x_1x_2 - 0.1091x_2^2$
E8 [7, 19]	$R_2(x) = x_1^2 + 1.3333x_1x_2 + 1.1667x_2^2$ $R_3(x) = -0.2272x_1^3 - 0.1396x_1^2x_2 + 0.3785x_1x_2^2 - 0.1798x_2^3$ $R_4(x) = 0.0136x_1^4 - 0.2864x_1^3x_2 + 0.1918x_1^2x_2^2 - 0.0530x_1x_2^3 + 0.0172x_2^4$	$Q_1(x) = -0.5605x_1 - 0.7255x_2$ $Q_2(x) = 0.3254x_1^2 + 0.0910x_1x_2 + 0.1015x_2^2$
E9 [13, 19]	$R_2(x) = 0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2$ $R_3(x) = -0.0739x_1^3 + 0.2594x_1x_2^2 - 0.0739x_1x_3^2$ $R_4(x) = -0.0301x_1^4 + 0.0573x_1^2x_2^2 - 0.0301x_1^2x_3^2 + 0.2501x_1x_2x_3^2 - 0.03x_2^4 - 0.03x_2^2x_3^2$	$Q_1(x) = -0.1478x_1$ $Q_2(x) = -0.0602x_1^2 - 0.06x_2^2$

[20], is implemented to obtain swing-up and stabilization of the pendulum at the desired upper equilibrium $x_d = (q_d, p) = (0, 0)$. The goal of this algorithm is to find a proper control input after a number of learning trials. Monitoring the DoA of the learned controller at every trial provides a stopping criterion to terminate learning once the DoA is large enough to fulfill the control objective. This leads to learning in a short amount of time. The parameterized control policy is given by

$$\hat{\pi}(x, \vartheta) = -\vartheta^T \Psi(x) \gamma(q - q_d) - mgl \sin(q) \quad (10)$$

where $\vartheta \in \mathbb{R}^n$ is a parameter vector, $\Psi \in \mathbb{R}^n$ is a user-defined basis function vector, and γ is a unit conversion factor with the value of one. The parameter vector ϑ is updated using the actor-critic reinforcement learning (RL) method by following the procedure described in [20]. Consequently, the saturated control input of the A-IDA-AC algorithm is computed at each time step by

$$u_k = \text{sat}(\hat{\pi}(x_k, \vartheta_k) + \Delta u_k) \quad (11)$$

where Δu_k is a zero-mean Gaussian noise, as an exploration term.

The desired system Hamiltonian is chosen in the quadratic form

$$H_d(x) = \frac{1}{2} \gamma (q - q_d)^2 + \frac{p^2}{2J}. \quad (12)$$

We exploit the desired system Hamiltonian as a candidate Lyapunov function to approximate the DoAs of the learned controllers at each learning trial [17]. Figure 10 illustrates the approximated DoAs of the learned controllers computed by the sampling approach at seven specific trials, where the trial numbers are also given. While learning is in progress, the DoAs of the learned controllers typically enlarge centered at the up equilibrium, but not necessarily monotonically. In this example, after 35 trials, the DoA of the controller is large enough to cover the initial state; hence, the learning process can be terminated sooner instead of running for all the scheduled trials. As such, the sampling method speeds up the process of learning controllers.

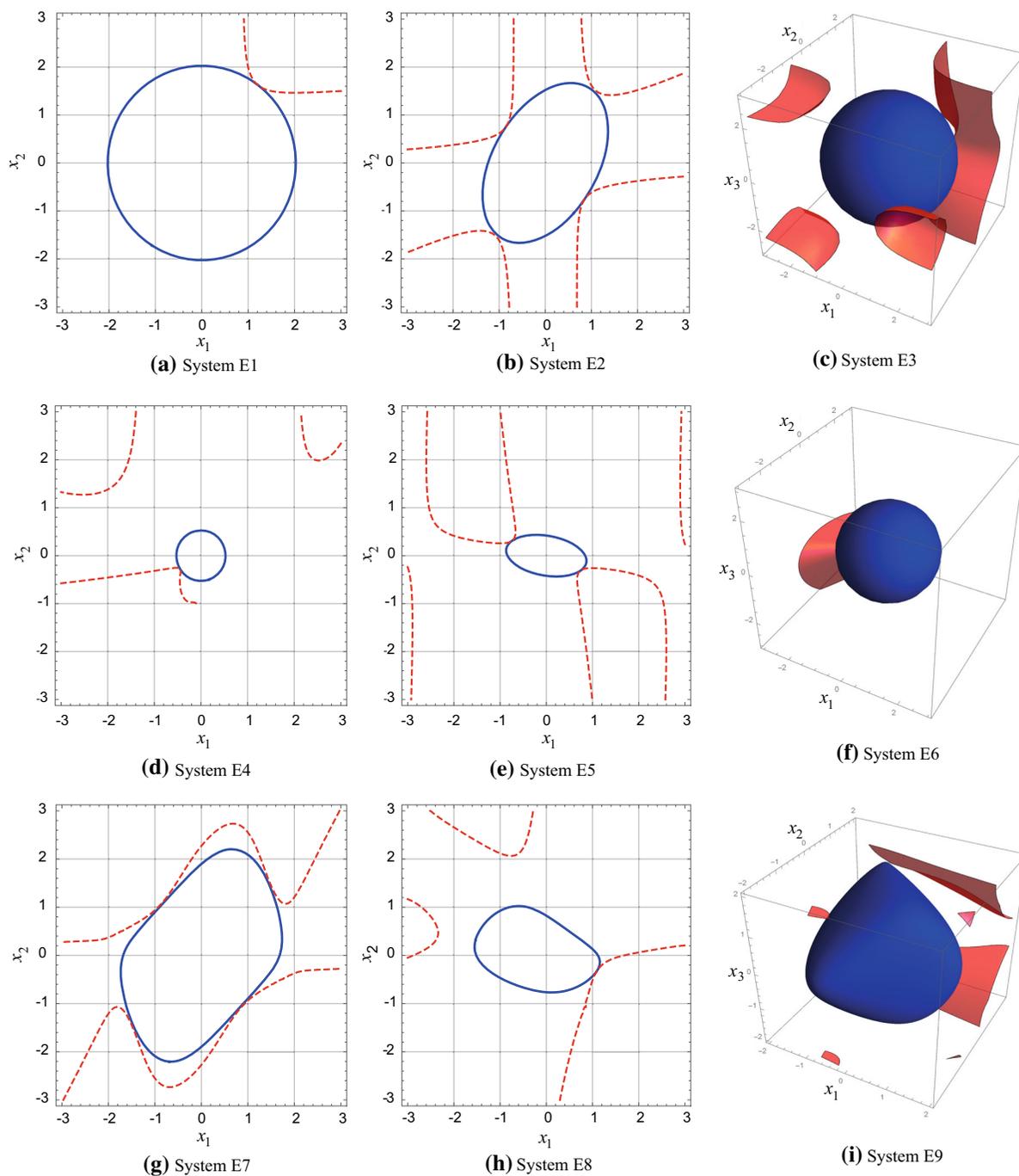


Fig. 8 Approximated DoAs for the origins of examples E1–E9 described in Tables 2 and 3 using the sampling method

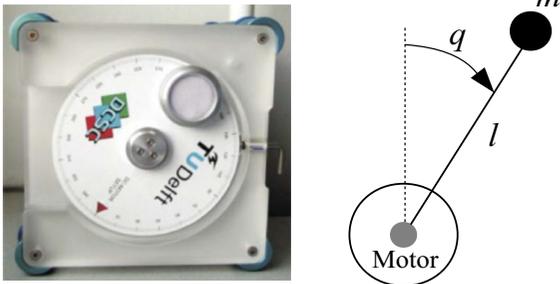


Fig. 9 Inverted pendulum and its schematic representation

Table 5 Physical parameters of the inverted pendulum

Physical parameter	Symbol	Value	Unit
Pendulum inertia	J	1.91×10^{-4}	kg m ²
Pendulum mass	m	6.8×10^{-2}	kg
Gravity	g	9.81	m s ⁻²
Pendulum length	l	4.20×10^{-2}	m
Damping in joint	b	3×10^{-6}	Nm s
Torque constant	K	5.36×10^{-2}	Nm A ⁻¹
Rotor resistance	R	9.5	Ω

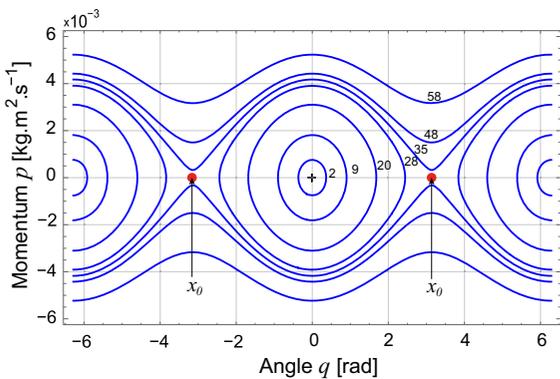


Fig. 10 Approximated DoAs of the learned controllers at seven specific trials for the inverted pendulum, where the trial numbers are also indicated

5 Conclusions

This paper has proposed a fast sampling approach for estimating the DoAs of nonlinear systems in real time. The approximated DoAs computed by this technique have been compared with the estimates derived by optimization-based methods. It is concluded that the sampling approach is fast and computationally effective in comparison with optimization-based methods

and it can be used for real-time applications. Although a formal guarantee for convergence does not exist yet, the empirical evidence arising from extensive simulations suggests that in practice this approach always converges to the exact level set for a sufficiently large number of samples. Moreover, the rate of convergence depends on the distribution function selected for sampling as well as the exploring regions of the state space. Using a more sophisticated distributed function can speed up convergence of the sampling procedure since it can avoid sampling in areas of the state space which are already known not belong to the DoA. As such, there is a trade-off between the speed of convergence and the computational cost imposed by the complexity of the sampling distribution function.

In addition, the sampling approach has been applied to approximate the DoAs of a passivity-based learning controller, designed for an inverted pendulum system, at every learning trial. This online approximation can be used as a stopping criterion for the learning process. This allows learning to be terminated as soon as the controller’s DoA is sufficiently large to satisfy the control objective. Thus, the proposed sampling method enables learning in a short amount of time.

Acknowledgments The first author gratefully acknowledges the support of the Ministry of Science, Research and Technology of the Islamic Republic of Iran through his scholarship. Moreover, we thank Subramanya P. Nagesh Rao for his help with parts of the code.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Amato, F., Cosentino, C., Merola, A.: On the region of attraction of nonlinear quadratic systems. *Automatica* **43**(12), 2119–2123 (2007)
2. Baier, R., Gerdts, M.: A computational method for non-convex reachable sets using optimal control. In: *Proceedings of the European Control Conference*, pp. 23–26. (2009)
3. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Berlin (2006)
4. Chesi, G.: Estimating the domain of attraction for uncertain polynomial systems. *Automatica* **40**(11), 1981–1986 (2004)

5. Chesi, G.: Domain of attraction: estimates for non-polynomial systems via LMIs. In: Proceedings of the 16th IFAC World Congress. (2005)
6. Chesi, G.: Estimating the domain of attraction via union of continuous families of Lyapunov estimates. *Syst. Control Lett.* **56**(4), 326–333 (2007)
7. Chesi, G.: Estimating the domain of attraction for non-polynomial systems via LMI optimizations. *Automatica* **45**(6), 1536–1541 (2009)
8. Chesi, G.: Domain of Attraction: Analysis and Control via SOS Programming. Springer, Berlin (2011)
9. Chesi, G.: Rational Lyapunov functions for estimating and controlling the robust domain of attraction. *Automatica* **49**(4), 1051–1057 (2013)
10. Chesi, G., Garulli, A., Tesi, A., Vicino, A.: LMI-based computation of optimal quadratic Lyapunov functions for odd polynomial systems. *Int. J. Robust Nonlinear Control* **15**(1), 35–49 (2005)
11. Genesio, R., Tartaglia, M., Vicino, A.: On the estimation of asymptotic stability regions: state of the art and new proposals. *IEEE Trans. Autom. Control* **30**(8), 747–755 (1985)
12. Hachicho, O.: A novel LMI-based optimization algorithm for the guaranteed estimation of the domain of attraction using rational Lyapunov functions. *J. Frankl. Inst.* **344**(5), 535–552 (2007)
13. Hachicho, O., Tibken, B.: Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments. In: Proceedings of the 41st IEEE International Conference on Decision and Control, pp. 3150–3155. (2002)
14. Henrion, D., Korda, M.: Convex computation of the region of attraction of polynomial control systems. *IEEE Trans. Autom. Control* **59**(2), 297–312 (2014)
15. Johansson, M., Rantzer, A.: Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Autom. Control* **43**(4), 555–559 (1998)
16. Khalil, H.K., Grizzle, J.: *Nonlinear Systems*. Prentice Hall, Upper Saddle River (2002)
17. Lopes, G.A., Najafi, E., Nagesh Rao, S.P., Babuska, R.: Learning complex behaviors via sequential composition and passivity-based control. In: Busoniu, L., Tamas, L. (eds.) *Handling Uncertainty and Networked Structure in Robot Control*, pp. 53–74. Springer, Berlin (2015)
18. Majumdar, A., Vasudevan, R., Tobenkin, M.M., Tedrake, R.: Convex optimization of nonlinear feedback controllers via occupation measures. *Int. J. Rob. Res.* **33**, 1209–1230 (2014)
19. Matallana, L.G., Blanco, A.M., Bandoni, J.A.: Estimation of domains of attraction: a global optimization approach. *Math. Comput. Model.* **52**(3), 574–585 (2010)
20. Nagesh Rao, S.P., Lopes, G.A., Jeltsema, D., Babuska, R.: Interconnection and damping assignment control via reinforcement learning. In: Proceedings of the 19th IFAC World Congress, pp. 1760–1765. (2014)
21. Najafi, E., Babuska, R., Lopes, G.A.: An application of sequential composition control to cooperative systems. In: Proceedings of the 10th International Workshop on Robot Motion and Control, pp. 15–20. (2015)
22. Najafi, E., Babuska, R., Lopes, G.A.: Learning sequential composition control. *IEEE Trans. Cybern.* (2015) (accepted)
23. Najafi, E., Lopes, G.A., Babuska, R.: Reinforcement learning for sequential composition control. In: Proceedings of the 52nd IEEE International Conference on Decision and Control, pp. 7265–7270. (2013)
24. Najafi, E., Lopes, G.A., Babuska, R.: Balancing a legged robot using state-dependent Riccati equation control. In: Proceedings of the 19th IFAC World Congress, pp. 2177–2182. (2014)
25. Najafi, E., Lopes, G.A., Nagesh Rao, S.P., Babuska, R.: Rapid learning in sequential composition control. In: Proceedings of the 53rd IEEE International Conference on Decision and Control, pp. 5171–5176. (2014)
26. Packard, A., Topcu, U., Seiler, P., Balas, G.: Help on SOS. *IEEE Control Syst. Mag.* **30**(4), 18–23 (2010)
27. Saleme, A., Tibken, B., Warthenpfuhl, S., Selbach, C.: Estimation of the domain of attraction for non-polynomial systems: a novel method. In: Proceedings of the 18th IFAC World Congress, pp. 10,976–10,981. (2011)
28. Sprangers, O., Babuska, R., Nagesh Rao, S., Lopes, G.: Reinforcement learning for port-Hamiltonian systems. *IEEE Trans. Cybern.* **45**(5), 1003–1013 (2015)
29. Tan, W., Packard, A.: Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. *IEEE Trans. Autom. Control* **53**(2), 565–570 (2008)
30. Tesi, A., Villoresi, F., Genesio, R.: On the stability domain estimation via a quadratic Lyapunov function: convexity and optimality properties for polynomial systems. *IEEE Trans. Autom. Control* **41**(11), 1650–1657 (1996)
31. Tibken, B., Hachicho, O.: Estimation of the domain of attraction for polynomial systems using multidimensional grids. In: Proceedings of the 39th IEEE International Conference on Decision and Control, pp. 3870–3874. (2000)
32. Topcu, U., Packard, A., Seiler, P.: Local stability analysis using simulations and sum-of-squares programming. *Automatica* **44**(10), 2669–2675 (2008)
33. Topcu, U., Packard, A.K., Seiler, P., Balas, G.J.: Robust region-of-attraction estimation. *IEEE Trans. Autom. Control* **55**(1), 137–142 (2010)
34. Vannelli, A., Vidyasagar, M.: Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica* **21**(1), 69–80 (1985)