Proceedings of the 2006 IEEE
Conference on Computer Aided Control Systems Design
Munich, Germany, October 4-6, 2006

A Computational Approach for Estimating Stability Regions

T. John Koo*† and Hang Su*

*Department of Electrical Engineering and Computer Science, Vanderbilt University
Nashville, TN, USA
†Department of Computer Science, Shantou University
Shantou, Guangdong, China

Abstract— **In this paper, we propose a computational approach for estimating the stability region of an asymptotically stable equilibrium point. The stability region is estimated through an iterative process specified as an algorithm. Reachable sets are used in the estimation algorithm for checking the invariant property of the initial estimate of a stability region and for representing the enlarged stability regions. The convergence of the estimation algorithm can be shown by considering the sequence properties of the reachable sets. Level set methods are used for representing reachable sets and tracking the evolution of the boundary of a reachable set since they can be used to effectively represent complex continuous sets and, furthermore, there exist efficient computation methods for computing the evolution of reachable sets for nonlinear systems. The proposed approach allows natural extension to higher dimensional systems and enables the computation to be carried out in a parallel manner. ReachLab, a model-based tool, is developed to enable rapid prototyping of the algorithm, and to allow the use of various computation methods for implementing the algorithm on a cluster of parallel computing machines. The accuracy of the proposed approach is compared with another accurate approach. The computation results for three nonlinear systems are presented.**

I. Introduction

Determining the stability region (or region of attraction) for nonlinear systems is an important problem in control from both theoretical and practical viewpoints. The problem has been studied extensively, for example in [1], [2], [3], [6], [18]. Numerous methods have been proposed for determining the stability region and in [1] they are divided in two classes: those using Lyapunov functions and Non-Lyapunov function methods.

Many methods belonging to the Lyapunov function approach, such as [2], [3], [18], are mainly based on La Salle's extension of Lyapunov theory [9], [15]. The Lyapunov functions are parameterized in forms, such as quadratic functions and piecewise quadratic functions, for some classes of nonlinear systems. In some cases, the computation of stability regions can be performed by using convex optimization techniques. Recently, Sum-Of-Square (SOS) polynomials [14] are used in [18] for parameterizing Lyapunov functions for extending the approach to nonlinear systems with highly nonlinear polynomial vector fields and SOS programming [14] is used for finding provable stability regions. However, due to the fact that the Lyapunov functions are parameterized, the stability region estimated by these methods can only be a subset of the true stability region. Another method, belonging to the Lyapunov function approach, is the Zubov method[9], [15]. This method in theory can provide true stability regions by solving an associated partial differential equation. Unfortunately, the partial differential equation seldom admits a closed-form solution and only approximation solution can be obtained, when solvable.

For the Non-Lyapunov function approach, such as [1], [6], the estimation of the stability region is synthesized from a number of system trajectories obtained by integrating the nonlinear system equations. For a fairly large class of nonlinear time-invariant systems, considered in [1], the stability boundary of a stable equilibrium point is shown to consist of the stable manifolds of all the equilibrium points (and/or closed orbits) on the stability boundary and a computational method is presented for determining the exact stability region. In [6], the trajectory reversing method is shown to be effective for enlarging the stability region of an asymptotically stable equilibrium point from an initial estimate of the stability region. These methods can be used to synthesize exact stability regions. However, these methods heavily rely on solving differential equations from finitely many points from the boundary of a stability region so the methods work well for lower dimensional systems. For higher dimensional systems, the stability regions can exhibit complex boundary shape and it is hard to capture the shape by just solving differential equations from finitely many points on the boundary.

In this paper, we propose a computational approach for estimating the stability region of an asymptotically stable equilibrium point. The stability region is estimated through an iterative process specified as an algorithm. Since all possible trajectories of a nonlinear system staring from an initial set for a fixed period of time can be captured by the concept of a reachable set, reachable sets are used in the estimation algorithm for checking the invariant property of the initial estimate of a stability region and for representing the enlarged stability regions. The convergence of the estimation algorithm can be shown by using the result presented in [6]. For certain classes of reachable sets, they can be used to effectively represent complex continuous sets and, furthermore, there exist efficient computation methods for computing the evolution of reachable sets for nonlinear systems. In particular, we consider the use of Level Set Methods (LSM) [12], [13], [16] for representing reachable sets and tracking the evolution of the boundary of the set. LSM embeds the boundary of a reachable set, called interface, as a zero level set of a higher dimensional implicit function and the evolution of the implicit function is governed by the associated partial differential equation, called a level set equation (LSE). Due to the use of the implicit function, any changes to the interface topology such as pinching and merging due to highly nonlinear behaviors of a system can be handled nicely in this so called *Eulerian* formulation. The LSE is solved numerically by using a *consistent* and *stable* finite difference approximation to the LSE over a fixed Cartesian grid in order to ensure the convergence of the solution. The accuracy of the proposed approach is compared with the accurate results produced by using the approach developed in [1] for lower dimensional systems. The proposed approach allows natural extension to higher dimensional systems and enables the computation to be carried out in a parallel manner. ReachLab [4], a model-based tool, is developed to enable rapid prototyping of the algorithm, and to allow the use of various computation methods for implementing the algorithm on a cluster of parallel computing machines.

In the next section, we will present important concepts and properties of stability regions related to nonlinear systems. Existing methods for estimating the stability region of a stable equilibrium point will be presented in Section III. In Section IV, the proposed computational approach for estimating the stability region will be presented. Computation results for three nonlinear systems will be shown in Section V. Finally, we will conclude our work in Section VI.

II. Stability Regions

In this section, some important concepts and properties of stability regions related to nonlinear systems are introduced. For general background on the theory for nonlinear systems the reader is advised to consult the paper [17], the books [15], [9] and the references therein.

Consider a nonlinear time-invariant system $\Sigma_c = (X, f)$ where the state space $X (= \mathbb{R}^n)$, the state vector $x \in X$ and the vector field $f : X \to \mathbb{R}^n$. The evolution of the state is specified by the following differential equations:

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_s, \quad t \geq 0 \qquad (1)$$

with an initial state $x_s \in X$. The solution of the system (1) at time t starting from x_s at $t = 0$ is called the *flow* and is denoted by $\phi(t, x_s)$. If the vector field of (1) f is *Lipschitz* continuous over X, then $\phi(t, x_s)$ exists and is unique in X for $t \geq 0$. Furthermore, $\phi(t, \cdot)$ is a homeomorphism for each t (see [15]).

A point x^* is said to be an *equilibrium point* of (1) if $f(x^*) = 0$. The set of equilibrium points of (1) is denoted by $\Gamma := \{x | f(x) = 0\}$. An equilibrium point $x^* \in \Gamma$ is said to be *hyperbolic* if none of the eigenvalues of the Jacobian matrix $D_x f$ at x^* has zero real part.

We assume that the origin 0 is an isolated equilibrium point and it is asymptotically stable, which means the origin 0 is stable in the sense of Lyapunov and attractive[15]. The *stability region* (or *region of attraction*) of 0, denoted by S, is defined as

$$S = \{x | \lim_{t \to \infty} \phi(t, x) = 0\} \qquad (2)$$

Its boundary and its closure are denoted by ∂S and $\overline{S}$, respectively. As shown in [15], the stability region S has the following topological properties: 1. S is open; 2. S is an invariant set, *i.e.* every trajectory of (1) starting in S remains in S for all t; and 3. the boundary of S, ∂S is invariant as well.

For a fairly large class of nonlinear time-invariant systems, considered in [1], the stability boundary of a stable equilibrium point is shown to consist of the stable manifolds of all the equilibrium points (and/or closed orbits) on the stability boundary. In particular, the stability region can be exactly characterized by using the following theorem stated in [1] when all trajectories converge to one or the other equilibrium points and that there are no closed orbits or other complex ω limit sets:

Theorem 2.1: [1] For the system (1), if 1. all the equilibrium points on the stability region boundary are hyperbolic, 2. the stable and unstable manifolds of the equilibrium points on the stability region boundary satisfies the transversality condition and 3. every trajectory on the stability boundary approaches one of the equilibrium points as $t \to \infty$, then the boundary of the stability region can be characterized as:

$$\partial S = \bigcup_{i=1}^{k} W_s(x_i) \qquad (3)$$

where the stable manifold[1] $W_s(x_i) = \{x | \lim_{t\to\infty} \phi(t, x) = x_i\}$ corresponds to a hyperbolic equilibrium point $x_i \in \Gamma \setminus \{0\}$ for $i = 1, \ldots, k$.

Based on the above characterization of the stability boundary, a computational method is presented in [1] for finding the stability region, and the method, when feasible, can find the exact stability region. The computational method works well for lower dimensional systems. For $n = 2$, the stability boundary can be represented by a collection of curves which can be computed by solving associated differential equations backwards in time starting from some specific points in the neighborhoods of the hyperbolic equilibrium points $x^* \in \Gamma \setminus \{0\}$ on the stability boundary. The specific points are chosen according to the stable eigenvector of the

[1] As shown in [1], [15], by using Hartman-Grobman and stable-unstable manifold theorems, the set W_s can be shown to be a manifold.

Jacobian matrix $D_x f$ at x^*. However, it is hard to derive the stable manifolds for higher dimensional systems by solving differential equations numerically since only finitely many points on a manifold can be considered.

III. Estimating Stability Regions

There are numerous methods in the literature referenced for estimating the stability regions of asymptotically stable equilibrium points by iteratively enlarging some known subsets of the stability regions. In [6], the trajectory reversing method is shown to be effective for enlarging stability regions. Consider an initial arbitrarily small estimate of the stability region for the equilibrium point 0 of (1) and the region is denoted as $S_0 \subset S$. The trajectory reversing method enlarges the given stability region S_0 by using backward integration. The following theorem provides sufficient conditions to such an enlargement.

Theorem 3.1: [6] Given the system (1), if the origin, 0, is an asymptotically stable equilibrium point, then the stability region S may be approximated arbitrary well by means of a convergent sequence of simply connected domains generated by the backward integration technique, starting from an initial stability region estimation.

The proof in [6] considers the fact that given the asymptotically stable equilibrium point 0 and the initial estimate of the stability region S_0 defined at time t_0, there *exits* a positive definite Lyapunov function $V(x)$ such that 1) $S_0 = \{x|V(x) < \alpha_0\}$ is simply connected; 2) $\dot{V}(x) < 0$, $\forall x \in \{x|V(x) \leq \alpha_0\}$. The set of predecessor points of S_0 at time $t_1(> t_0)$, denoted as S_1, can be specified by the following equation:

$$S_1 = \bigcup_{x \in S_0} \bigcup_{t \in [t_0, t_1]} \phi^{-1}(t, x) \tag{4}$$

By considering the existence and uniqueness properties of differential equations and the properties of the Lyapunov function, several important results are derived in [6]: 1. S_1 is simply connected; 2. $S_0 \subset S_1$; and 3. $S_1 \subset S$. Then, by using the same argument repeatedly, it can be shown that there exist a monotonic increasing sequence $\{t_i\}$ with $t_0 < \cdots < t_{i-1} < t_i$ such that the sequence $\{S_i\}$ satisfies $S_0 \subset \cdots \subset S_{i-1} \subset S_i \subset S$ and every set in $\{S_i\}$ is simply connected. Furthermore, the theorem guarantees that the sequence converges to the true stability region. In [6], backward integration is used to implement the trajectory reversal method for deriving the stability region but it could only work for lower dimensional systems since stability regions can exhibit complex boundary shape and it is hard to capture the shape by just solving differential equations from finitely many points on the boundary of S_0.

Theorem 3.1 provides the foundation for justifying the use of iterative methods for enlarging a given estimate of the stability region of an asymptotically stable equilibrium point. In this paper, we propose to use a reachable set for representing the estimate S_i at every stage of the enlargement process since all possible trajectories of the nonlinear system (1) staring from a set for a fixed time period can be captured by a reachable set. For certain classes of reachable sets, they can be used to effectively represent complex continuous sets and, furthermore, there exist efficient computation methods for computing the evolution of reachable sets for nonlinear systems.

Here, the definitions of reachable set are provided. There are backward and forward reachable sets. The backward (forward) reachable set, denoted as $Pre_I(P)$ ($Post_I(P)$), is a collection of states in X which can reach (can be reached by) some state of a set $P \subseteq X$ in some time specified by a time set $I \subseteq \mathbb{R}_+$. The sets are defined as

$$\begin{aligned} Pre_I(P) &= \{x'|\exists x \exists t\ x \in P \wedge t \in I \wedge x' = \phi^{-1}(t, x)\}, \\ Post_I(P) &= \{x'|\exists x \exists t\ x \in P \wedge t \in I \wedge x' = \phi(t, x)\}. \end{aligned}$$

Thus, the sequence $\{S_i\}$ can be generated by using $S_i = Pre_{[t_{i-1}, t_i]}(S_{i-1})$ iteratively starting from S_0. Since the system of interest (1) is time-invariant, we have $S_i = Pre_{[0, \tau_i]}(S_{i-1})$ or simply $Pre_{\tau_i}(P)$ where $\tau_i = t_i - t_{i-1}$. Hence, $S_i = Pre_{\sum_{j=1}^{i} \tau_j}(S_0)$.

Therefore, by using the concept of reachable sets and the result presented in Theorem 3.1, given a simply connected estimate of stability region S_0 of the asymptotically stable equilibrium point 0, we can compute a sequence of simply connected stability regions $\{S_i\}$ and furthermore the sequence converges to the true stability region.

IV. Computation Methods

In this section, we are interested in how reachable sets can be numerically computed for the estimation of stability regions. Level set methods (LSM) [13], [16], [12] provide an effective way to represent reachable sets and track the evolution of the interface. LSM embeds an interface ∂P which bounds a region $P \subset \mathbb{R}^n$ as the zero level set of a higher dimensional implicit function $J(x, t)$ at time t, where $x \in \mathbb{R}^n$, and J is related to P in the following way:

$$\begin{aligned} &J(x, t) > 0, \text{ for } x \notin P \\ &J(x, t) \leq 0, \text{ for } x \in P \end{aligned} \tag{5}$$

The interface ∂P can be identified by the implicit function $J(x, t) = 0$. Hence, complex shape of a region can be represented, without any requirement of convexity or connectivity of the region. By differentiating both sides of the implicit function, the interface evolution of P can be depicted by the resulting partial differential equation (PDE) called the level set equation (LSE) as shown in:

$$\dot{J}(x, t) + f(x) \cdot \nabla J(x, t) = 0 \tag{6}$$

where $f(x)$ is the vector field of the nonlinear system (1). Given a region P at time $t = 0$, *i.e.* $P =$

$\{x|J(x,0) \leq 0\}$, the forward reachable set can be represented as $Post_{\Delta t}(P) = \{x|\exists\tau \in [0,\Delta t], J(x,\tau) \leq 0\}$. By solving the corresponding LSE, the reachable set can be computed and tracked.

In LSM, the formulation of interface evolution is *Eulerian*, since the interface is captured by the implicit function J. It makes gross changes to the interface topology, such as merging and pinching easier to handle, as opposed to the *Lagrangian* formulation, which typically requires ad hoc techniques to address mesh connectivity during merging and pinching. Since (6) is a convection equation by using the vector field of the system, this is called the convection method of LSM.

In [13], [16], the LSE is solved numerically by using the method of lines, which assumes that the spatial discretization can be separated from temporal discretization in a semidiscrete manner that allows the temporal discretization of the LSE to be treated independently as an ordinary differential equation (ODE). A Cartesian grid can be applied to discretize the state space. By using a signed distance function, we assign each grid point a value, which is the distance to the interface, in order to represent regions as in (5). Assume the grid size of the Cartesian grid is $\Delta x = [\Delta x_1, \Delta x_2, \cdots, \Delta x_n]^T$, where n is the number of dimensions, and Δx_i is the grid size of the ith dimension. Because of limited storage space for the Cartesian grid to be implemented, the state space X is assumed to be bounded. Let the length of the ith dimension of X be L_i. In total, there are N $(= \prod_{i=1}^{n} \frac{L_i}{\Delta x_i})$ grid points, $\{x_i\}_{i=1}^{N}$, in the Cartesian grid. Therefore, by using the method of lines, there are N differential equations, which are

$$\dot{J}(x_i,t) + f(x_i) \cdot \nabla J(x_i,t) = 0, \text{ for } i = 1, 2, \cdots, N.$$

In order to approximate the spatial derivative $\nabla J(x_i,t)$ at x_i, appropriate finite difference methods, such as HJ-(W)ENO, Hamilton-Jacobi (Weighted) Essentially Non-Oscillatory, can be used by considering the values of its neighboring grid points. The upwind difference method gives first-order accuracy of spatial approximation, while HJ-(W)ENO gives up to fifth order accuracy. Since the spatial approximation of these methods at x_i is tightly coupled, the spatial approximation results in an ODE with N coupled differential equations. The ODE can then be solved by numerical methods such as forward Euler and TVD (Total Variation Diminishing) Runge-Kutta, each with a different order of accuracy.

As described in [13], in order to ensure convergence of the numerical solution, the finite difference approximation to the LSE must be both consistent and stable. The combination of a spatial discretization scheme with a time discretization scheme is a *consistent* finite difference approximation to the PDE, if the approximation error converges to zero as both $\Delta x \to 0$ and $\Delta t \to 0$. Such combination is *stable*, if small errors in the approximation are not amplified as the solution is marched forward in time. Stability can be enforced by the Courant-Friedreichs-Lewy (CFL) condition, which restricts the relationship of spatial discretization Δx and temporal discretization Δt by the following equation:

$$\Delta t \max(\sum_{i=1}^{n} \frac{|f_i(x)|}{\Delta x_i}) = \alpha$$

where $\alpha \in (0,1)$, n is the number of dimensions, and $f_i(x)$ is the ith component of vector field. For a fixed Cartesian grid, in which Δx is fixed, according to the CFL condition, the upper bound of Δt can be determined. And according to the Lax-Richtmyer equivalence theorem [13], a finite difference approximation to a linear partial differential equation is *convergent*, *i.e.* the correct solution can be obtained as $\Delta x \to 0$ and $\Delta t \to 0$, if and only if it is both consistent and stable. In particular, the combination of the upwind difference scheme with the forward Euler time discretization that conforms to the CFL condition is convergent, as shown in [13], as is the combination of the fifth order HJ WENO scheme and third order TVD Runge-Kutta. Therefore, we can use LSM to numerically compute the forward reachable set, $Post_{\Delta t}(P)$, for the evolution of set P within time Δt. Then, the backward reachable set, $Pre_{\Delta t}(P)$, can be obtained by computing the forward reachable set with vector field on reversed flow direction.

We propose an algorithm for estimating the stability region of an asymptotically stable equilibrium point of the nonlinear system (1). Given a dynamic system and a bounded analysis region X with grid size Δx, we are interested in finding the stability region for an asymptotically stable equilibrium point 0 of this system. The algorithm is shown in Table I.

TABLE I

STABILITY REGION ESTIMATION ALGORITHM

1) Initial set selection:
   a) Set an initial set P as a circle around the given equilibrium point 0 with radius $r = r_0$.
   b) Set $R_0 = P^c$, as the complement of P.
   c) Repeat $k = 0, 1, 2, \cdots$
      Compute $R_{k+1} = R_k \cup Prec_{\Delta t}(R_k)$
      Until $R_{k+1} \subseteq R_k$.
   d) Set $P = F^c$.
2) Backward reachable set computation:
   a) Set $S_0 = P$. Set $t = 0$.
   b) Repeat $k = 0, 1, 2, \cdots$
      Compute $S_{k+1} = S_k \cup Prec_{\Delta t}(S_k)$
      Until $S_{k+1} \subseteq S_k$ or $k\Delta t > T$.
   c) Set $\hat{S} = S_{k+1}$, which gives the estimation of the stability region of the given equilibrium point p.

In the algorithm, step 1) is used to determine the portion of an initial ball P that is inside the stability region. If the P is completely inside the stability region, the loop in 1)c) will terminate after the first iteration. Otherwise, the loop iteratively removes the points from P that eventually leave the set P. The loop terminates until a fixed point is reached so that all the points outside the stability region but in P are removed. In step 2), the stability region is expanded from S_0, the initial guess provided in the step 1), by iteratively computing the backward reachable set of S_0 for an integer multiple of Δt. It has two termination conditions, either the computation has reached a fixed point solution, or the total amount of time exceeds the specified limit T.

Although it has been shown in Theorem 3.1 that there exists a monotonically increasing sequence such that the sequence of stability region estimates will converge to the true stability region, the algorithm may terminate due to several reasons. Firstly, if the number of iterations exceeds the predefined limit, the algorithm will terminate due to the first condition. When this occurs, for any point on the estimated boundary, $\hat{S}$, it would take at most T time to reach some point on the boundary of S_0. Secondly, if the true stability region is bounded and lies within the bounded analysis region, the algorithm is guaranteed to terminate at a fixed point solution, *i.e.* $S_{k+1} \subseteq S_k$ for some k, due to the finite representation of reachable sets on a Cartesian grid. Finally, if the true stability region is only partially included in the analysis region, the algorithm terminates and a fixed point solution is reached for the same reason, but the computed stability region is only the part that lies inside the analysis region but without any successor of it being outside the analysis region.

V. Computation Results

A. Computation Platform: ReachLab

Due to the intensive requirement for storage space and computation time for solving the LSE by using LSM, we have implemented LSM on a cluster of parallel computing machines at Advanced Computing Center for Research & Education (ACCRE), Vanderbilt University. This facility provides a fast ethernet interface such as Myrinet and more than 1000 processors to meet high-speed communication and data-intensive computation needs. In order to handle various precision and speed requirements, different spatial discretization schemes, such as finite difference and HJ-(W)ENO, and temporal discretization schemes, such as forward Euler and TVD Runge-Kutta, are implemented (in C language). A library, called the parallel level set computation kernel (PLS), is constructed to include all necessary functions of spatial/temporal discretization, initial condition, and set operation.

PLS provides mechanisms to allow flexible configuration of system parameters, such as grid size and approximation schemes. By dividing the Cartesian grid into topologically hypercube partitions and distributing the computation of each partition into a single computation node, PLS parallelizes the computation process. Computation nodes of neighboring partitions exchange boundary information every internal time step to perform spatial approximation. Due to its workload distribution possibility, PLS can solve larger problems, which may require larger memory space to store the information or longer computation time, than a single machine.

In order to automate the design and implementation process of analysis algorithms for dynamic systems, and utilize the PLS more efficiently and effectively, we have developed a Model-Integrated Computing (MIC) based computation platform, ReachLab [4], implemented on Generic Modeling Environment (GME) [8]. The ReachLab design enables that analysis algorithms can be reused and constructed hierarchically, and furthermore the design process can be highly automated, in which executable code for different computation kernels, such as PLS, can be automatically generated by ReachLab interpreters. Therefore, system models and analysis algorithms designed in ReachLab can be reused with different computation kernels to serve various computation needs.

B. Examples

In the following, the computation results based on the proposed computational approach for estimating the stability regions are presented by considering three nonlinear systems with asymptotically stable equilibrium points. Two of these three nonlinear systems are 2 dimensional and they satisfy the conditions specified in Theorem 2.1. Hence, accurate boundaries of the stability regions can be computed by using the algorithm presented in [1] for validating the results produced by the proposed approach.

Our proposed stability region estimation algorithm has been designed on ReachLab, and executable code has been automatically generated by the interpreter for PLS. The same executable code has been used on all of the following examples, and the only change to the code is the system dynamics of each example, and corresponding computation parameters, such as grid size. For all of the following examples, HJ-WENO and third-order TVD Runge-Kutta schemes are used. We also assume uniform grid sizes in these examples. The computation parameters for the following examples are listed in Table III at the end of this section.

1) Example 1: The first example is a 2D nonlinear system defined as ODE:

$$\begin{aligned} \dot{x}_1 &= -2x_1 + x_1 x_2 \\ \dot{x}_2 &= -x_2 + x_1 x_2 \end{aligned}$$

This system has been studied in [1], [7], [11]. It has an asymptotically stable equilibrium point (0,0), and its stability region has been estimated by our proposed algorithm. The result is shown in Fig. 1. The initial set is selected as a circle centered at the stable equilibrium point with radius $r = 0.5$, and the stability region is given by $\hat{S}$ in the algorithm. The boundary of the stability region is computed by using the method in [1] at the type-one equilibrium point (1,2), shown as the solid curve in the figure.

The result in Fig. 1 uses the grid size $\Delta x = 0.025$. Our experiments have shown that if the grid size increases, the accuracy of estimated stability region will become lower. We use area loss $(1 - \frac{\text{area of } \hat{S}}{\text{area of } S})$ as our error estimation, and Table II lists the error estimation of this example for different grid sizes. In the table, area is computed directly by integration. It shows that the estimation error of stability region decreases as grid size becomes smaller.

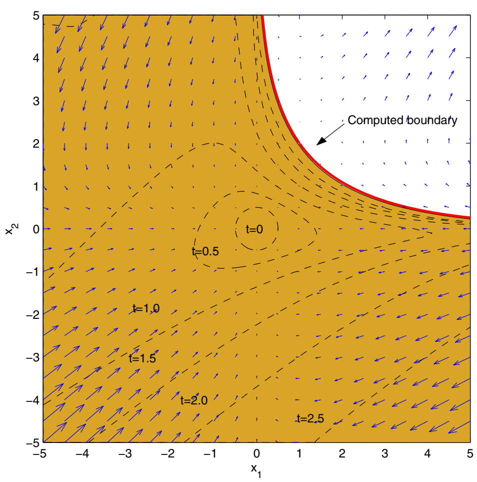


Fig. 1. Results for Example 1 with $\Delta x = 0.025$. The shadow region is the estimated stability region $\hat{S}$. The central dashed circle is the initial set. The region enclosed by each dashed boundary is S_{k+1} in the algorithm, for $k = 0, 1, 2, 3, 4$. The solid curve is the computed boundary by using the method in [1]. And the phase portrait of the system is also drawn.

TABLE II

ERROR ESTIMATION OF EXAMPLE 1 FOR DIFFERENT GRID SIZES

Δx	0.4	0.2	0.1	0.05	0.025
Area of $\hat{S}$	79.71	80.46	80.85	81.06	81.27
Area of S	81.43	81.43	81.43	81.43	81.43
Area loss	2.12%	1.19%	0.72%	0.45%	0.20%

2) Example 2: The second example is a 2D nonlinear speed-control system as ODE [1], [5], [6]:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_2 - x_1 - 6x_1^2(x_1 + x_2 + 1)$$

It has an asymptotically stable equilibrium point (0,0) (the other stable equilibrium point (-0.78865,0) has a stability region that is complement of the stability region of (0,0)), and its stability region has been estimated by our proposed algorithm, as shown in Fig. 2. The radius of the initial set is $r = 0.15$, and again, and the stability region is given by $\hat{S}$ in the algorithm. The boundary of the stability region is computed by using the method in [1] at the type-one equilibrium point (-0.21135,0), shown as the solid curve in the figure. As we can see from the figure, some part of the estimated stability region is inside the computed boundary, while some part is outside the boundary (although very little). This shows that the LSM does not necessarily give under approximation results. It only gives results that converge to viscosity solution as the grid size Δx and time step Δt go to zero. Therefore, the result shown in the figure can be further refined with smaller Δx.

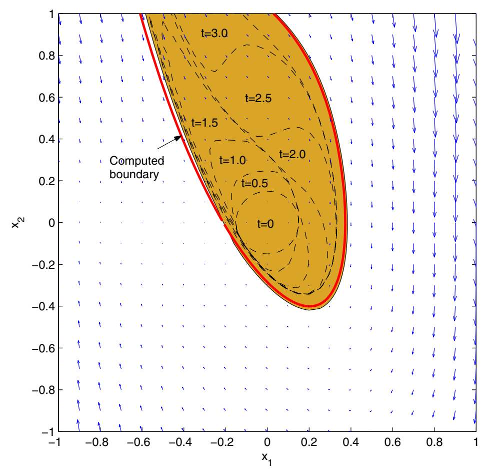


Fig. 2. Results for Example 2. The shadow region is the estimated stability region $\hat{S}$. The central dashed circle is the initial set. The region enclosed by each dashed boundary is S_{k+1} in the algorithm, for $k = 0, 1, 2, 3, 4, 5$. The solid curve is the computed boundary by using the method in [1]. And the phase portrait of the system is also drawn.

3) Example 3: The last example is a 3D nonlinear servomechanism as ODE [10], [3], [2]:

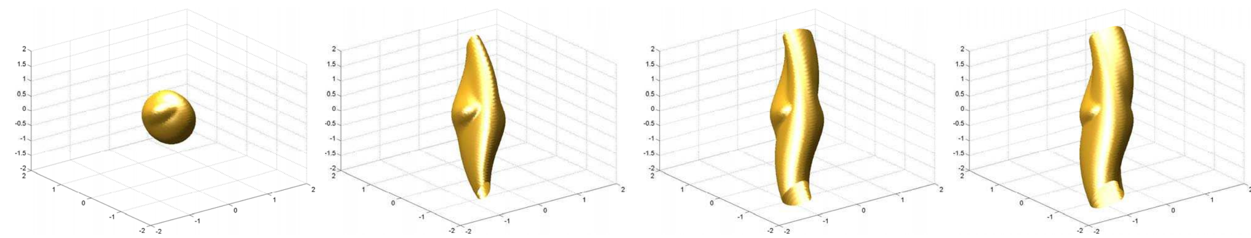
$$\dot{x}_1 = -x_2$$
$$\dot{x}_2 = -x_3$$
$$\dot{x}_3 = -0.915x_1 + (1 - 0.915x_1^2)x_2 - x_3$$

The equilibrium point at (0,0,0) is asymptotically stable, and its stability region has been estimated by our proposed algorithm. The initial set is selected as a sphere centered at the stable equilibrium point with radius $r = 0.5$, and the stability region is given by $\hat{S}$ in the algorithm. Fig. 3 gives a sequence of the computation results for different t. This example shows the scalability of our proposed approach for estimating stability regions for higher dimensional systems.

TABLE III

COMPUTATION PARAMTERS

Example	1	2	3
Figure	Fig. 1	Fig. 2	Fig. 3
Analysis Set X	[-5,5]×[-5,5]	[-1,1]×[-1,1]	[-2,2]×[-2,2]
Initial Set S_0	$r = 0.5$	$r = 0.15$	$r = 0.5$
Grid Size Δx	0.1	0.05	0.1
# of Grid Points	10^4	1.6×10^3	6.4×10^4
Time Step Δt	0.5	0.5	0.5
Bounded Time T	4.0	8.0	3.5
Computation Time	≈ 20 min	≈ 65 min	≈ 25 min
Machine	4 nodes	4 nodes	8 nodes



(a) (b) (c) (d)

Fig. 3. Result of $\hat{S}$ in Example 3 at (a) $t = 0.5$; (b) $t = 1.5$; (c) $t = 2.5$; (d) $t = 3.5$.

VI. Conclusions

In this paper, a computational approach for estimating the stability region of an asymptotically stable equilibrium point is proposed. The stability region is estimated through an iteratively process specified as an algorithm. Reachable sets are used in the estimation algorithm for checking invariant property of the initial estimate of a stability region and for representing the enlarged stability regions. The convergence of the estimation algorithm can be shown by using the result presented in [6]. Level set methods[12], [13], [16] are used for representing reachable sets and tracking the evolution of the boundary of a reachable set. LSM embeds the interface as a zero level set of a higher dimensional implicit function and the evolution of the implicit function is governed by the level set equation. The LSE is solved numerically by using a *consistent* and *stable* finite difference approximation to the LSE over a fixed Cartesian in order to ensure the convergence of the solution. The proposed approach enables natural extension to higher dimensional systems and enables the computation to be carried out in a parallel manner. ReachLab is developed to enable rapid prototyping of algorithms, and to allow the use of various computation methods for implementing the algorithms on a cluster of parallel computing machines. The computation results for three nonlinear systems are presented. The accuracy of the proposed approach is compared with the accurate results produced by using the approach developed in [1] for lower dimensional systems. We have shown that the estimation error of stability region decreases as grid size becomes smaller. In the future, when the efficiency of LSM for high dimensional state space is improved, the proposed approach can be more efficient on estimating stability regions.

VII. Acknowledgement

This work is supported by the National Science Foundation Faculty Early Career Development (CAREER) Program, Award No. 0448234. This work is conducted in part using the resources of the Advanced Computing Center for Research and Education at Vanderbilt University.

References

[1] H.-D. Chiang, M. W. Hirsch, and F. F. Wu. Stability Regions of Nonlinear Autonomous Dynamical Systems, *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 16-27, Jan. 1988.

[2] H.-D. Chiang and J. S. Thorp. Stability Regions of Nonlinear Dynamical systems: A Constructive Methodology. *IEEE Transactions on Automatic Control*, vol. 34, no. 12, pp. 1229-1241, 1989.

[3] E. J. Davison and E. M. Kurak. A Computational Method for Determining Quadratic Lyapunov Functions for Nonlinear Systems, *Automatica*, vol. 7, pp. 627-636, 1971.

[4] A. Dubey, X. Wu, H. Su, T. J. Koo. Computation Platform for Automatic Analysis of Embedded Software Systems Using Model Based Approach. *Third International Symposium on Automated Technology for Verification and Analysis (ATVA)* Lecture Notes in Computer Science, Vol. 3707, pp. 114-128, Springer-Verlag, Taipei, Taiwan, October 4-7, 2005.

[5] F. Fallside and M. R. Patel. Step-response behavior of a speed-control system with a back-e.m.f. nonlinearity. *Proc. IEE.*, London, vol. 112, pp. 1979-1984.

[6] R. Genesio, M. Tartaglia and A. Vicino. On the Estimation of Asymptotic Stability Regions: State of the Art and New Proposals, *IEEE Transactions on Automatic Control*, vol. 30, no. 8, pp. 747-755, Aug. 1985.

[7] R. Genesio and A. Vicino. Some results on the asymptotic stability of second-order nonlinear systems. *IEEE Trans Automat. Cont.*, vol. AC-29, pp. 857-861, Sept. 1984.

[8] G. Karsai, J. Sztipanovits, A. Ledeczi, T. Bapty. Model-Integrated Development of Embedded Software, *Proceedings of the IEEE*, vol. 91, no. 1, pp. 145-164, January, 2003.

[9] H. K. Khalil. *Nonlinear Systems*. Macmillan, 1992.

[10] H. Ku and C. Chen. Stability study of a third-order servomechanism with multiplicative feedback control. *AIEE Transactions, Part 1*, vol. 77, pp. 131-136, 1958.

[11] A. N. Michel, N.R. Sarabudla, and R. K. Miller. Stability analysis of complex dynamical systems some computational methods. *Circuits Syst. Signal Processing*, vol. 1, pp. 171-202, 1982.

[12] I. Mitchell, C. Tomlin. Level sets methods for computation in Hybrid Systems. *Hybrid Systems : Computation and Control*, LNCS 1790, pages 310-323, 2000.

[13] S. Osher, Ronald Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Springer, 2003.

[14] S. Prajna, A. Papachristodoulou, and P. Parrilo. Introducing SOSTools: A General Purpose Sum of Squares Programming Solver. In *Proceedings of IEEE Conference on Decision and Control*, pp. 741–746, 2002.

[15] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer-Verlag, New York, 1999.

[16] J.A.Sethian. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry. Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, 1999

[17] S. Smale, Differentaible Dynamical Systems, *Bull. Amer. Math. Soc.*, vol. 73, pp. 747-817, 1967.

[18] W. Tan and A. Packard. Stability Region Analysis using Sum of Squares Programming. In *Proceedings of the American Control Conference*, 2006.