AUTOMATIC GENERATION OF LYAPUNOV FUNCTIONS USING GENETIC PROGRAMMING

Benyamin Grosman and Daniel R. Lewin†

PSE Research Group, Wolfson Department of Chemical Engineering,
Technion–IIT, Haifa 32000, Israel

Abstract: The selection of a suitable Lyapunov function is a critical step in the stability analysis of nonlinear systems. The appropriate choice can lead to the possibility of guaranteeing increased regions of attraction in the vicinity of a desired stable equilibrium point. Typically, its selection is limited to well-known functions and on the intuition of the researcher, which can lead to suboptimal functions. This paper describes the preliminary results obtained using a novel method to identify Lyapunov functions using genetic programming.

Keywords: Nonlinear dynamics; Stability; Radius of convergence; Lyapunov functions; Genetic programming.

1. INTRODUCTION

The formal definition of system stability is at the focus of differential and integral analysis, having engaged the attention of leading mathematicians and physicists including Torricelli, Laplace, Lagrange and others. However, it was only in 1892 that a clear criterion established, with the publication of the work of the Russian mathematician, Lyapunov (Lyapunov, 1907). He defined a scalar function inspired by a classical energy function (Lyapunov's direct method), which has three important properties that are sufficient for establishing the domain of attraction of a stable equilibrium point: (a) It must be a local positive definite function, (b) it must have continuous partial derivatives, and (c) its time derivative along any state trajectory must be negative semi-definite (Slotine and Li, 1991). While Lyapunov theory provides powerful guarantees concerning a system's stability once an appropriate function is identified, it regrettably provides no guidance on how to select it.

This paper introduces a novel approach for the automated generation of a Lyapunov function for the analysis of a given dynamic system using genetic programming (GP). The genetic program, which is an optimization method inspired by natural evolution, evolves an improved Lyapunov function, driven by its required properties as described above, in such a way that the resulting region of attraction is maximized. The paper is structured as follows. First, we review the formal definition of Lyapunov functions. Then, the approach used to generate Lyapunov function using genetic programming is described in some detail. Three typical autonomous nonlinear systems are then analyzed using the approach.

2. LYAPUNOV STABILITY PRINCIPLES

This study concerns stability analysis of autonomous systems, of the general form:

$$\dot{\underline{x}} = \underline{f}(\underline{x}) \qquad (1)$$

where $\underline{x} = R^n$. A system is said to be autonomous if f does not depend explicitly on time. An equilibrium point of the system of Eq. (1), $\underline{x} = \underline{x}^*$, is one that satisfies:

$$f(\underline{x}^*) = 0 \qquad (2)$$

An equilibrium point is said to be stable in the sense of Lyapunov if for any n-dimensional ball of radius $\varepsilon > 0$ there exists an n-dimensional ball of radius $\delta(\varepsilon)$, such that for any trajectory $\underline{x}(t_0,\underline{x}_0)$, starting in δ, then $\|\underline{x}(t, x_0)\| < \varepsilon$ for any $t > 0$. Otherwise, the equilibrium point is unstable. These conditions are stated formally as:

---

† Author for correspondence. Email: dlewin@tx.technion.ac.il, URL: http://pse.technion.ac.il

$$\left\| \underline{x}_0 \right\| < \delta(\varepsilon) \Rightarrow \left\| \underline{x}(t) \right\| < \varepsilon \ \forall t \geq 0 \qquad (3)$$

This definition binds an equilibrium point to its domain of attraction without requiring it to be asymptotic stable. An equilibrium point is also said to be asymptotically stable if it is stable, and if in addition there exists some $r > 0$ such that $\left\| \underline{x}_0 \right\| < r$ implies that $\underline{x}(t, x_0) \to 0$ as $t \to \infty$.

A function $v(x)$ is said to be a Lyapunov function if it satisfies two main properties: (a) $v(x)$ must be locally positive definite, meaning that it is bounded from below by a constantly increasing function equal to zero at the origin; (b) the derivative of $v(x)$ with respect to time:

$$\frac{dv(x)}{dt} = \frac{\partial v(x)}{\partial x} f(x), \qquad (4)$$

must be semi-negative definite in the domain of attraction. This type of function must exist around a stable equilibrium point (Shankar, 1999) encapsulated by an attraction domain in the ball B_r, of radius r, where the above properties are satisfied. Furthermore, for systems of the type of Eq. (1), asymptotic stability is guaranteed if the function's derivative (Eq. 4) is locally negative definite (Slotine and Li, 1991).

3. SELECTION OF LYAPUNOV FUNCTIONS USING GENETIC PROGRAMMING

Genetic programming (GP) is an optimization method inspired by the principles of Darwinian evolution (Koza, 1992). The GP is based on simple rules that imitate biological evolution. Unlike conventional optimization techniques that manipulate the parameters of an initial estimate of the solution, GPs maintain a population of potential models, each structured in a tree-like fashion, with basis functions linking nodes of inputs and constants, as illustrated by the example chromosome in Figure 1. The probability of a given model surviving into the next generation dependent on the performance of each individual, evaluated using a fitness function, with the most successful (efficient) chromosomes having a higher probability to reproduce. In synthetic evolution, biological reproduction is mimicked by operators such as crossover (pairing) and mutation, which create a generation of offspring solutions. Crossover generates new features in the solution space by combining genetic information, while mutation accomplishes this by adding random perturbations. Fitness-proportional selection, combined with these genetic operators produces generation after generation of offspring solutions. Since the more appropriate solutions are given higher probabilities to reproduce, one would expect improved solutions over generations.

The work reported here is based on the GP code developed in a previous study (Grosman and Lewin, 2004), who demonstrated that it has the capacity to generate compact nonlinear models that accurately predict the input-output system behavior without requiring the user to specify the model complexity in advance. This is achieved mainly due to the adaptive nature of the fitness function adopted. This and additional features described by (Grosman and Lewin, 2004) make the code superior to generic genetic programming codes (e.g., McKay et al., 1997).

The approach is harnessed to the automated generation of Lyapunov functions. First, the candidate model tree is further specified by addition of constants that fully define the models, which then become the principal degrees of freedom that are adjusted to optimize the model performance. For example, the model in Figure 1 has two degrees of freedom, requiring the insertion of two parameters, and is sent to the optimizer as: $y = u_3 \times (\theta_2 u_1 + \theta_1 u_2)$. This procedure is important to ensure that a good candidate model structure is not dropped because of a possible poor fit resulting from insufficient "tuning" parameters. After this procedure is over the derivative function (Eq. 4) has to be created for each candidate model. This is accomplished analytically using symbolic computation of $v(x)$ and $\dot{v}(x)$ using MATLAB®.

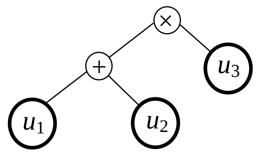


Fig. 1. Tree structure for model: $y = u_3 \times (u_1 + u_2)$. The multiplication functional is positioned at the root of the tree, and u_3 and $u_1 + u_2$ are its branches.

The candidate Lyapunov functions and their derivatives are required to satisfy the two Lyapunov properties summarized in Section 2. First, the degree to which these properties are satisfied in a small ball in the vicinity of the origin are checked (in all three examples described below, the radius of this initial ball radius was set at 10^{-3}). The candidate function score is computed using a discretized grid positioned around the origin, and graded according to:

$$f_i = -r_i \cdot 2^{-r_i \cdot (D_+ + V_- - 1)} \qquad (5)$$

where r_i is the radius of the ball for candidate i, D_+ is the number of semi-positive time derivates on the grid excluding zero derivate at the origin, and V_- is the number of negative candidate function values on the grid including the origin. For example, a grid, in the vicinity of the origin, that includes fifteen compounds points, where two of them are positive Lyapunov function derivates and three of them contain negative Lyapunov function values, is evaluated as: $f_i = -10^{-3} \cdot 2^{-10^{-3} \cdot (2+3-1)} = -9.97 \times 10^{-4}$, i.e., its value is 99.7% of the maximum possible score on the given radius. This penalization is sufficient to eliminate this candidate from further consideration.

Note that the optimizer attempts to minimize Eq. (5). Thus, the candidate function that satisfies the conditions in the initial ball gradually increases its radius until one of the conditions is violated. Candidates that violate the conditions in the initial ball are graded by their relative successful grid points using (Eq 5). Clearly, the candidates that satisfy the Lyapunov conditions in the initial ball are graded according to their largest convergence region, satisfying the Lyapunov conditions. The fitness of such models always exceeds those that violate the conditions in the initial ball. Furthermore, note that Eq. (5) evaluates the cost function as $-r_i$, while D_+ is equal to zero and V_- is equal to one, thus satisfying the two Lyapunov conditions. Consequently, the function enables the detection of the maximum radius that satisfies the Lyapunov conditions. Finally, it is noted that the function as defined previously is designed to analyze asymptotical stable equilibrium points, which are usually of interest in most control systems analyses. However, it can be easily transformed to examine stability in the general Lyapunov sense.

The score recorded for each candidate is actually the volume of the ball attained in each case. However, the score appropriate for a GP is a fitness value in the range between zero and unity. This is mapped using the function:

$$\hat{f}_i = \frac{1}{1+1/V_i} \qquad (6)$$

where $V_i = r_i^2$ is the volume of the region of attraction obtained with the candidate function i. Evidently, candidates that exhibit large regions of attraction will achieve fitness values approaching unity, while those with small regions of attraction will score poorly. As described previously, our GP penalizes candidates with excessively complex models, by implementing the following correction:

$$F_i = \frac{\hat{f}_i}{1+\exp(\gamma[n_b-(M_c+\beta)])} \qquad (7)$$

In the above formulation, n_b, sums the number of branches in the model tree; the model in Figure 1, for example, shows a model with four branches. The fitness function in Eq. (7) with n_b is penalized if it contains significantly more branches than that of the best model in the parent population, M_c. Thus, M_c is continuously reset to the number of branches of the best detected model in each new generation. Moreover, the additional adjustable parameters, γ and β, are used in shaping the penalizing sigmoid, such that Eq. (7) favours simpler models. In this way, M_c moves the centre of the sigmoid each generation, while γ controls its slope and β its intercept (the larger β is, the larger number of branches greater than M_c that are accepted).

4. EXAMPLES

The approach was tested on three second order dynamical systems, all of which have stable equilibrium points at the origin and exhibit an unstable limit cycle demarcating the domain of the attraction.

4.1 Example 1 (Slotine and Li, 1991, pg. 64).

The state equations for this system are:

$$\begin{aligned} \dot{x}_1 &= x_1(x_1^2+x_2^2-2)-4x_1x_2^2 \\ \dot{x}_2 &= x_2(x_1^2+x_2^2-2)+4x_2x_1^2 \end{aligned} \qquad (8)$$

This has a steady equilibrium point at the origin and an unstable limit cycle at a radius of $\sqrt{2}$, which is actually the boundary of the stable domain of attraction, as shown in Figure 2.

Five distinct GP runs were carried out on this system, each manipulating populations of 25 solutions, running for 50 generations. The best Lyapunov function was identified as:

$$v(x) = x_1^2 + x_2^2 \qquad (9)$$

Although this is the trivial “energy” function, it is noteworthy that it was identified by the GP despite the fact that all five runs were initiated from random starting conditions, without any a priori information. Furthermore, the optimization computed $\sqrt{2}$ as the radius of the domain of attraction.

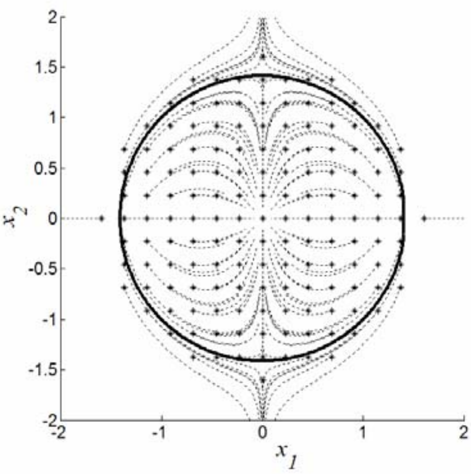


Fig. 2. x_1–x_2 phase plane for Example 1. The bold line indicates the unstable limit cycle that demarcates the region of attraction, and the stars indicate various initial conditions.

4.2 Example 2 (Shankar, 1999, pg. 194).

The state equations for this system are:

$$\begin{aligned} \dot{x}_1 &= x_1\left(x_1^2+x_2^2-1\right)-x_2 \\ \dot{x}_2 &= x_2\left(x_1^2+x_2^2-1\right)+x_1 \end{aligned} \qquad (10)$$

As on the previous case study, the system has a stable equilibrium point at the origin and a limit cycle that separates between the domain of attraction and the unstable region. This limit cycle is located at a unit radius from the origin, as shown in Figure 3.

As in the first example, five GP runs were executed, which identified the intuitive Lyapunov function of Eq. (9), as the function that guarantees the largest radius of convergence. In this case, the maximum radius found by the GP approach was unity, which is the theoretical value. Table 1 illustrates one of the GP runs for Example 2, which graphically demonstrates the power of the method to identify the appropriate Lyapunov function. Note the way the two-dimensional surfaces are adjusted during the evolution until the final “best” Lyapunov function is achieved in generation 9.

The examples this far appears to indicate that the approach merely confirms the intuitive idea that an appropriate selection of Lyapunov functions is the generic form:

$$v(x) = x^T P x, \qquad (11)$$

where P is a positive definite matrix. In a similar vein, Krasovskii's method (Slotine and Li, 1991) suggests Lyapunov functions of the form $v(x) = f^T(x)f(x)$. In general, as will be seen in the next example, the approach can be helpful in identifying appropriate Lyapunov functions, when intuitive judgement leads to sub-optimal results.

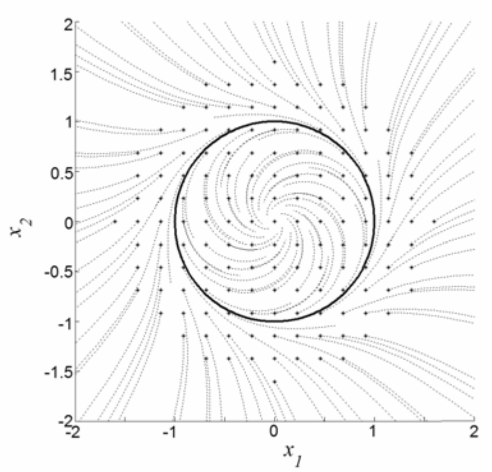


Fig. 3. x_1–x_2 phase plane for Example 2. The bold line indicates the unstable limit cycle that demarcates the region of attraction, and the stars indicate various initial conditions.

4.3 Example 3. Van der Pol's equation.

The state equations for this system are:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -0.5x_2\left(1 - x_1^2 + 0.1x_1^4\right) - x_1 \qquad (12)$$

As shown in Figure 4, this system has a stable equilibrium point at the origin, with a domain of attraction __ surrounded by an unstable limit cycle that is itself surrounded by another stable limit cycle. Note that contrary to the previous two examples, the limit cycles are not circular.

The goal is to find the best Lyapunov function, meaning the one that will guarantee the largest attraction domain. However, as shown in Figure 5, a Lyapunov function of the generic form of Eq. (9) leads to a semi-definite negative derivative in the domain, rather than the desired definite negative derivative. This fact prevents the GP to continue the search outside the initial ball as mentioned in the previous section. Furthermore, such a solution is insufficient for this example, since it is known that it exhibits asymptotic stability at the origin (if the origin was the only stable domain this would be sufficient for asymptotic stability).

Figures 6 and 7 confirm the power of genetic programming in finding appropriate Lyapunov functions. Both figures describe example quadratic functions generated by the GP, which would be hard to arrive at intuitively. The one that is plotted in Figure 6 is the following function:

$$v(x) = 1.122(x_2{}^2 + 0.85694x_1(1.2364x_1 + x_2)), \qquad (13)$$

which gives a convergence radius of 1.11. In contrast, the solution plotted in Figure 7:

$$v(x) = 1.007u_1{}^2 + 1.0219(u_1 - 1.0063u_2 - 1.9265Sin(0.77835u_1))^2, \qquad (14)$$

involves $\sin(x_1)$. As shown in Figure 4, this solution guarantees a larger attraction domain (of radius 1.96) than that of Eq. (11) and of the trivial semi-negative domain of Eq. (8), with both of the latter regions of attraction being equal. In contrast, the intuitive method of Krasovskii gives a radius of convergence of only 0.31.

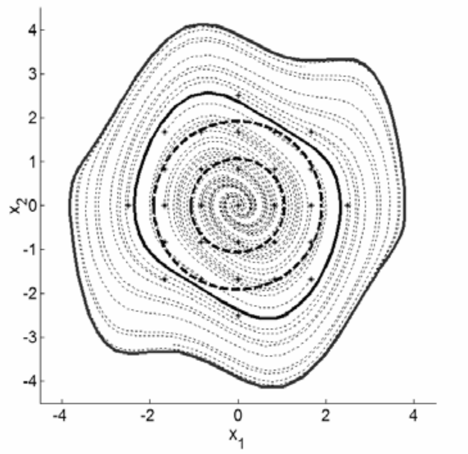


Fig. 4. x_1–x_2 phase plane for Example 3. The solid inner line indicates the unstable limit cycle that demarcates the region of attraction, the solid outer line is the stable limit cycle, and the dotted lines indicate two stable domains of attraction detected by the GP. The stars indicate various initial conditions.

Figure 8 displays the evolution of the radius of attraction for Van der Pol's equation, as the GP searches for the appropriate functional form of the Lyapunov function. Note that all of the candidates identified are quadratic in form, which is not essential but expected, seeing as that function most be locally positive definite.

5. CONCLUSIONS

This paper has presented a novel approach for the automatic generation of Lyapunov functions suitable for stability analysis of nonlinear systems. More specifically, the capability of genetic programming has been demonstrated to disclose the 'best' Lyapunov function for three two-dimensional dynamical systems. While on first two examples the genetic programming identifies the well-known quadratic Lyapunov function, a result that could have been done without the power of genetic programming, the third example demonstrates the ability of the GP to detect complex structures for Lyapunov function, which could have been identified by hand only by chance. It is believed that the described methodology could give rise to many important applications in the field of stability and process control. This ability has yet to be tested on more challenging systems, with the ultimate goal of our research being to design nonlinear stable controllers by enlarging the domain of the attraction using genetic programming.

REFERENCES

Grosman, B. and D. R. Lewin. (2004). Adaptive Genetic Programming for Steady-state Process Modeling, *Comput. Chem. Eng.,* **28**(12), 2779-2790.

Grosman, B. and D. R. Lewin (2002). Automated Nonlinear Model Predictive Control using Genetic Programming, *Comput. Chem. Eng*, **26**(4-5), 631-640.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.

McKay, B., M. Willis, and G. W. Barton (1997). Steady-State Modeling of Chemical Process Systems using Genetic Programming, *Comput. Chem. Eng.,* 21(9), 981-996.

Lyapunov, M. A. (1907). Problème général de la stabilité du mouvement, *Ann. Fac. Sci. Toulouse*, **9**, 203-474. (Translation of the original paper published in 1892 in *Comm. Soc. Math. Kharkow* and reprinted as Vol. 17 in *Ann. Math Studies*, Princeton University Press, Princeton, N.J., 1949.)

Shankar, S. (1999). *Nonlinear Systems: Analysis, Stability and Control*. Springer Verlag, New York.

Slotine, J. J. E. and W. Li (1991). *Applied Nonlinear Control*. Prentice Hall Inc., Englewood Cliffs, New Jersey.

Vidyasagar, M. (1993). *Nonlinear Systems Analysis*. Second Edition, Prentice Hall, Inc. Englewood Cliffs, New Jersey.

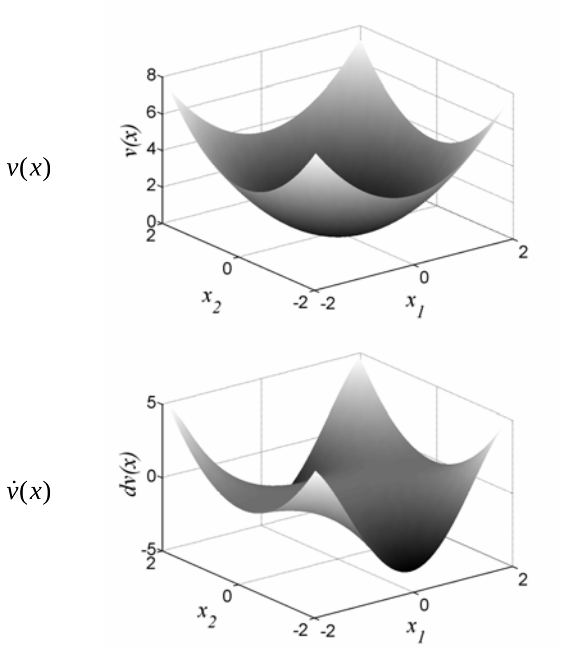


Fig 5. Surface representation of the generic Lyapunov function of Eq. (9) for Example 3.

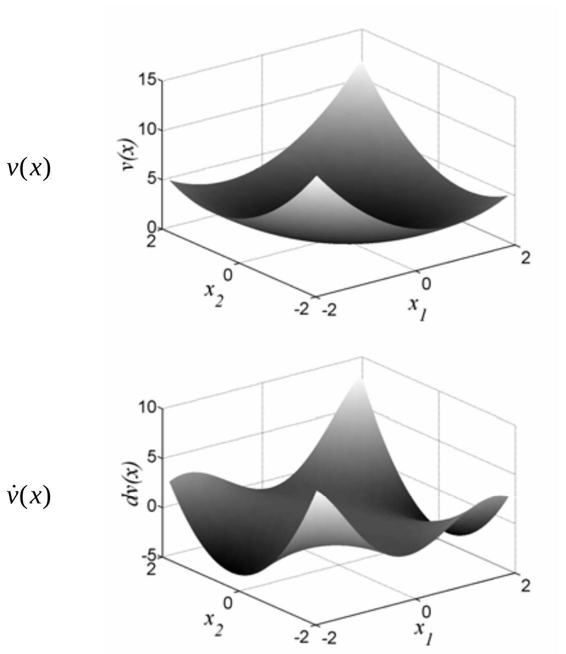


Fig 6. Surface representation of the Lyapunov function of Eq. (13) for Example 3. In this case, the surface of the derivative is negative except the origin, guaranteeing asymptotic stability.

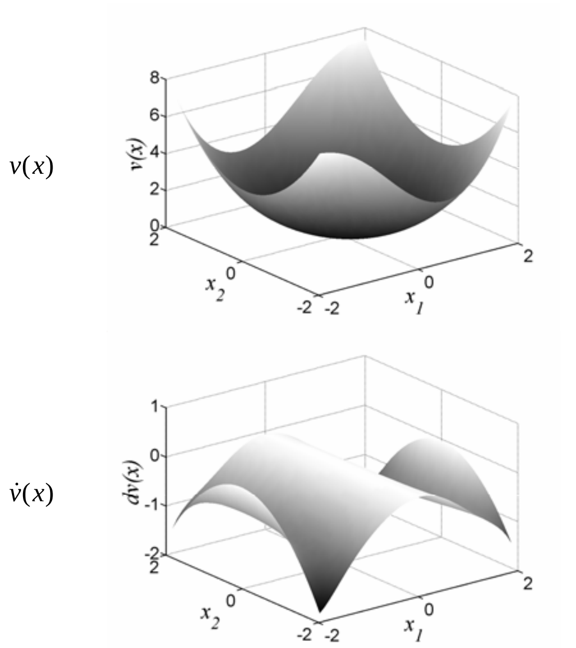


Fig 7. Surface representation of the Lyapunov function of Eq. (14) for Example 3. This functional form exhibits a negative derivative, thus guaranteeing asymptotic stability, as well as a larger attraction domain.

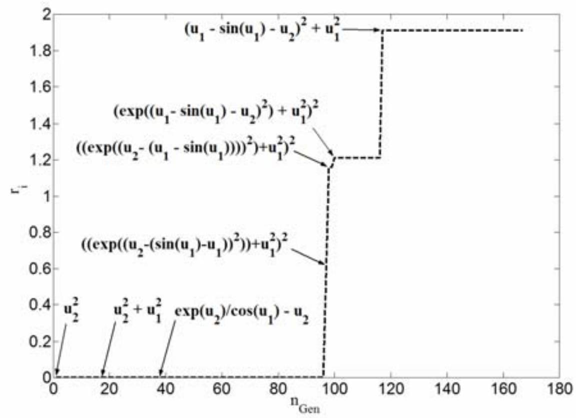
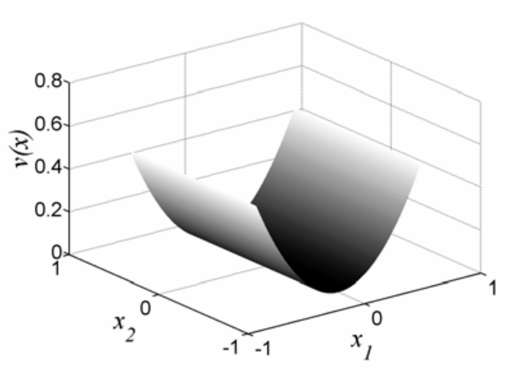
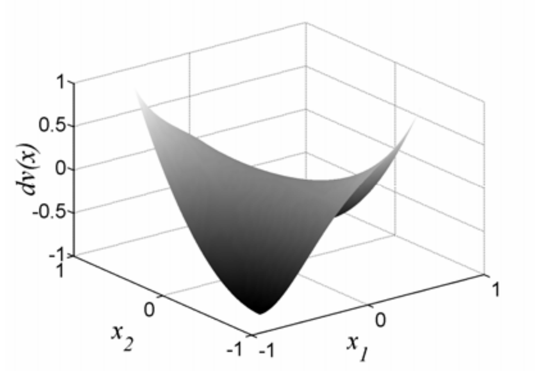
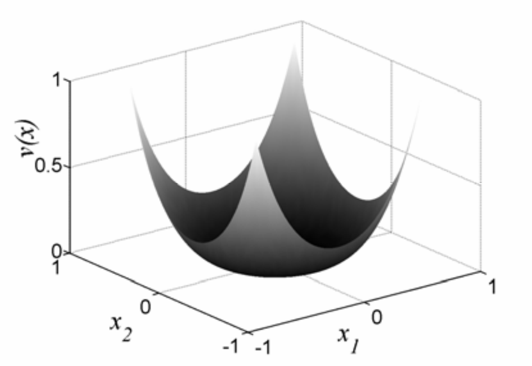
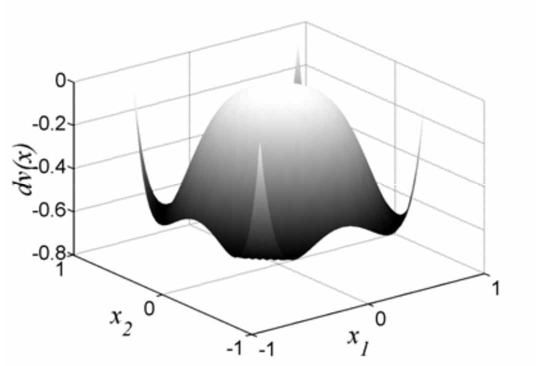
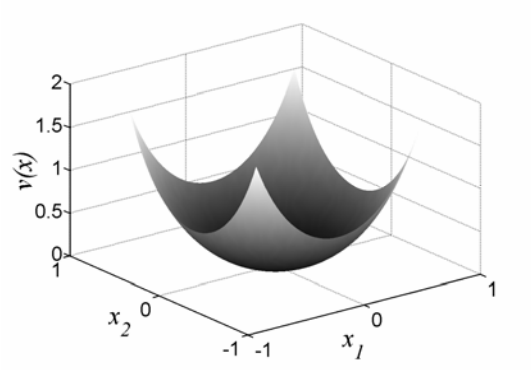
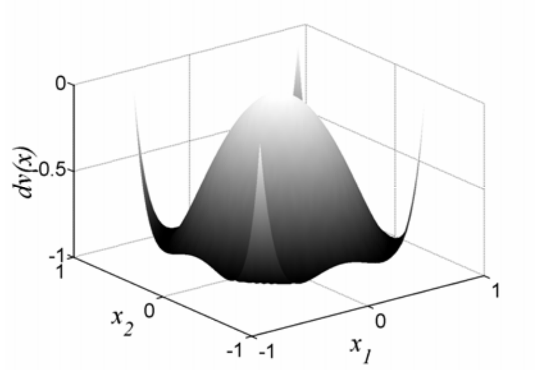
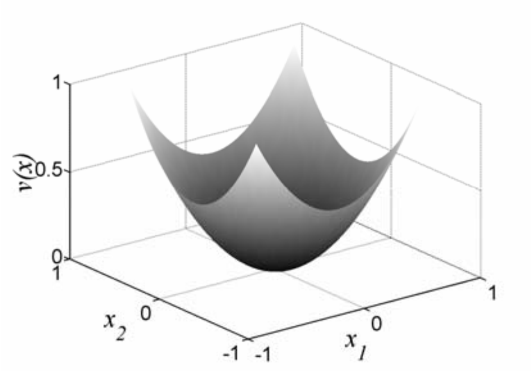
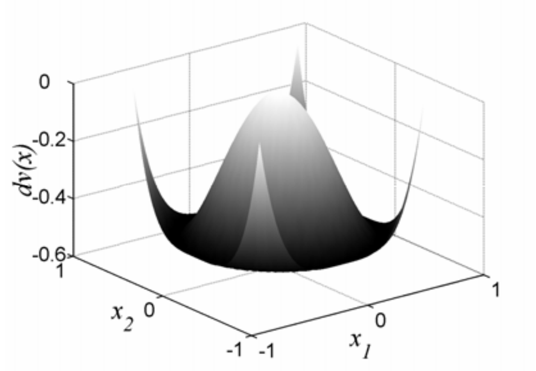


Fig. 8. Evolution of the optimal radius of domain of attraction for Example 3. Note the gradual evolution of the functional form of a successful Lyapunov function, giving Eq. (14) from the 117th generation.

Table 1: A typical evolution of the GP-derived Lyapunov function for Example 2.

<table>
<tr><th>$v(x)$</th><th>$\dot{v}(x)$</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">Generation 1-5: $v = x_1^2$</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">Generation 6-7: $v = (x_1^2 + x_2^2)^2$</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">Generation 8: $v = \exp(x_1^2 + x_2^2)$</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">Generations 9-50: $v = x_1^2 + x_2^2$</td></tr>
</table>