Model-free control of Lorenz chaos using an approximate optimal control strategy

Shuai Li, Yangming Li, Bu Liu, Timmy Murray

Abstract

In this paper, we are concerned with model-free control of the Lorenz chaotic system, where only the online input and output are available while the mathematic model of the system is unknown. The problem is formulated from an optimal control perspective and solved using an iterative method. The convergence of the iteration and the stability of the control law are proven in theory. Simulations validate theoretical conclusions and demonstrate the effectiveness of the proposed method.

1. Introduction

Starting from the discovery of Lorenz chaos [1], the phenomena of chaos are widely found in many different fields, such as physics [2], economics [3], engineering [4], biology [5], etc. In some applications, the effect of chaos is harmful and is undesirable. With this consideration, regulating the evolution trajectory from chaos to a fixed operating point or within a small neighborhood of the fixed operating point, which is often refereed as control of chaos, has attracted intensive studies in recent years [6–9].

Chaotic systems are very sensitive to noises and their dynamic evolution may deviate far from the nominal one even though they are perturbed by a small noise. This fact makes it difficult to distinguish chaotic evolutions from stochastic processes due to the seemingly randomness demonstrated by chaotic processes although the chaotic system is indeed deterministic. Resulting from such a super sensitivity of the chaotic system to noises including the modeling error, most existing work on controlling chaos design control laws based on a precisely known dynamic model. In practice, an off-line system identification step has to be firstly considered to obtain the mathematic model of the dynamic chaotic system and then the control law is designed based on the identified model. However, the modeling error is unavoidable in most real implementations. Such an error may lead to chaotic deviations from the desired operating point for the chaotic system with the elaborately designed control input although the control input works well in ideal situations. In other words, there is a dilemma between the super sensitivity of a chaotic system to modeling errors and the unavoidability of the modeling error. Actually, compared with others, it is relatively difficult to identify the system model of a chaotic system [10], not to mention the simultaneous estimation and control of the chaotic model. There is few work on controlling chaotic systems purely based on the system output without knowing model information. If any, most of them assume a complete knowledge on the model...
structure while the unknowns are some parameters [11–14]. We make progress in the direction of model-free control of chaotic systems. In this paper, a controlling mechanism is proposed to control the Lorenz chaotic system, where neither the parameters nor the model structure are assumed to be known.

Inspired by the success of fuzzy control, fuzzy controllers for chaotic systems are widely explored [15–18]. In [15], a fuzzy controller is introduced to stabilize Chua’s circuit. The authors in [16] presents a fuzzy control algorithm to control chaotic systems without requiring knowledge on the accurate system model. In [17], an adaptive fuzzy model is presented to control an uncertain Lorenz system. Similar strategies are applied to synchronize chaotic systems in [18] with adaptive fuzzy controllers. It is worth noting that although accurate analytical model of the chaotic system is not required, the construction of the fuzzy rules in the fuzzy controller design requires experience on the chaotic system model. Chaotic systems are essentially nonlinear. The ability of a neural network for universal approximation makes it possible to control the behavior of chaotic systems. In [19], an universal learning network optimized by a generalized learning algorithm is introduced to control the maximum Lyapunov exponent of a chaotic system and therefore to stabilize it. The authors in [20] uses a dynamic neural network as an identifier, uses a local optimal controller to remove the nonlinearity of the chaos based on the identified model. A predictive control law using self-recurrent wavelet neural networks is proposed in [21]. In the proposed framework, the neural network is used as a model predictor for predicting the dynamic property of chaotic systems. A gradient law is applied to train the neural network based predictor and controller. Evolutionary computation provides a way for optimization without model information. Application of evolutionary computation to chaotic systems is able to produce model-free control inputs. In [22,23], the authors apply the minimum entropy control method to the control of chaos and use a particle swarm optimization algorithm for the model-free minimization. In [24], the genetic algorithm with the use of multi-objective fitness functions is employed to design and optimize a local control of chaos. Other works on using evolutionary computation to the control or synchronization of chaos can be found in [25–29] and references therein. By deliberate designs, a chaotic system can be stabilized to a fixed point using the existing methods referred above. However, due to the lack of optimization mechanisms considering the problem along the infinitely long time horizon, globally optimal control cannot be reached. Instead, the proposed method in this paper employs a functional defined on infinite time horizon as the objective function and the solution demonstrates a tendency to take advantage of the inherent favorable dynamics in the chaotic systems.

In this paper, we study the problem from an optimal control perspective. In this framework, the goal is to find the input sequence to minimize the cost function defined on infinite horizon under the constraint of the system dynamics. The solution can be found by solving a Bellman equation according to the principle of optimality [30]. Then an adaptive dynamic programming strategy [31–33] is utilized to numerically solve the input sequence in real time. Note that the optimal control problem, as widely studied in model predictive control, can be numerically solved using finite horizon control techniques by approximately considering the cost function on finite horizon [34,35]. However, due to the curse of dimensionality, such a strategy is computational intensive and is not suitable for the online control of rapid evolving systems [36], such as chaotic systems. More importantly, model information is necessary for the design of a finite horizon controller [37] and therefore this technique cannot be applied to model-free control of chaos directly.

The remainder of this paper is organized as follows: in Section 2, the Lorenz system with a single input is presented. In Section 3, the problem is formulated as a constrained optimization problem. In Section 4, the model-free control strategy is presented and analyzed in theory. Section 5 considers issues in the practical implementation and introduces the critic model and the action model to facilitate the control. In Section 6, simulations are given to show the effectiveness of the proposed method. The paper is concluded in Section 7.

2. Lorenz chaotic attractor

We are concerned with the following Lorenz chaotic model with a single input,  
\[
\begin{align*}
\dot{x}_1 &= \sigma(x_2 - x_1) \\
\dot{x}_2 &= \rho x_1 - x_2 - x_1 x_3 + u \\
\dot{x}_3 &= -\beta x_3 + x_1 x_2
\end{align*}
\]  
(1a)
(1b)
(1c)
where \(\sigma > 0\), \(\rho > 0\) and \(\beta > 0\). Under the parameter setup \(\sigma = 10\), \(\rho = 97\) and \(\beta = 8/3\), the uncontrolled system with \(u = 0\) exhibits chaotic behavior. Note that in system (1a), the input \(u\) enters the system through the dynamic Eq. (1a) (b). The control problem can be similarly handled in the cases with input entering the system through Eq. (1a) or Eq. (1c). For system (1a), the input \(u\) acts on the dynamic evolution of \(x_2\) directly while has no direct actions on the state variables \(x_1\) and \(x_3\). For example to the variable \(x_1\), the effect of \(u\) on \(x_1\) takes place indirectly through its action on \(x_2\). This indirect control channel makes it more challenging to control system (1a) than to control the Lorenz system with three inputs acting on \(x_1\), \(x_2\) and \(x_3\) directly.

3. Problem formulation

Our goal is to stabilize system (1a) to a given operating point by state feedback \(u = u(x_1, x_2, x_3)\). In this section, we formulate such a control problem from the optimal control perspective.

---

S. Li et al. / Commun Nonlinear Sci Numer Simulat 17 (2012) 4891–4900
Without losing generality, we set the origin as the desired operating point, i.e., we consider the problem of controlling the state of the system (1a) to the origin or a small neighborhood around it. For the case with other desired operating points, the problem can be equivalently transformed to the one with the origin as the operating point by shifting the coordinates. At each sampling period, the state values $x_1, x_2$ and $x_3$ can be used to evaluate the one step performance. Specifically, we define the following utility function associated with the one-step cost at the $i$th sampling period,

$$U_i = U(x_1(i), x_2(i), x_3(i))$$

(2)

To regulate the system to the origin, the utility function is designed to be a function with a large value if the state has a large deviation from the origin and a small value if the state value is close to the origin. For simplicity, we choose the following function for $U(x_1, x_2, x_3)$,

$$U(x_1, x_2, x_3) = \begin{cases} 0 & |x_1| \leq \delta_1, |x_2| \leq \delta_2 \text{ and } |x_3| \leq \delta_3 \\ 1 & \text{otherwise} \end{cases}$$

(3)

where $|x|$ denotes the absolute value of $x$. The parameters $\delta_1 > 0, \delta_2 > 0, \delta_3 > 0$. At each step, there is a value $U_i$ and the total cost starting from the $k$th step along the infinite time horizon can be expressed as follows,

$$J_k = J(x(k), u(k)) = \sum_{i=k}^{\infty} \gamma^i U_i$$

(4)

where $x(k)$ is the state vector of system (1a) sampled at the $k$th step with $x(k) = [x_1(k), x_2(k), x_3(k)]$, $\gamma$ is the discount factor with $0 < \gamma < 1$, $u(k) = [u_k, u_{k+1}, \ldots, u_n]$ is the control sequence starting from the $k$th step. Note that for the deterministic system (1a), the preceding states after the $k$th step are determined by $x(k)$ and the control sequence $u_k$. Accordingly, $J_k$ is a function of $x(k)$ and $u(k)$ with $J_k = J(x(k), u(k))$. Also note that both the cost function $J_k$ and the utility function $U_k$ are defined based on the discrete samplings of the continuous system (1a). Now, we can define the problem of controlling the chaotic system (1a) in this framework as follows,

$$\min_{u(0), u(1), \ldots, u(\infty) \in U} J_0 = \sum_{i=0}^{\infty} \gamma^i U_i$$

subject to:

$$\begin{align*}
\dot{x}_1(t) &= \sigma(x_2(t) - x_1(t)) \\
\dot{x}_2(t) &= \rho x_1(t) - x_2(t) - x_1(t)x_3(t) + u(t) \\
\dot{x}_3(t) &= -\beta x_3(t) + x_1(t)x_2(t) \\
u(t) &= u(i) \quad \text{for } t \leq t < (i+1)\tau
\end{align*}$$

(5)

where $J_0$ is defined by (2) and (3), $\tau > 0$ is the sampling period, the set $U$ defines the feasible control actions, $J_0$ is the cost function for $k = 0$ in (4). It is worth noting that $J_0$ is a function of $u(0) = [u_0, u_1, \ldots, u_n]$ and $x(0)$ according to (4). The optimization in (5a) is relative to $u(0)$ with a given initial state $x(0)$. Also note that in the optimization problem (5a), the decision variable $u(0), u(1), \ldots, u(\infty)$ are defined in every sampling period. The control action keeps the value in the duration of two consecutive sampling steps. This formulation is consistent with the real implementations of digital controllers.

**Remark 1.** There are infinitely many decision variables, which are $u(0), u(1), \ldots, u(\infty)$, in the optimization problem (5a). Therefore, this is an infinite dimensional problem. It cannot be solved directly using numerical methods. Conventionally, such kind of problem is often solved by using a finite dimensional approximation [38]. For instance, a possible way is to replace the objective function in (5a) with $\min_{u(0), u(1), \ldots, u(\infty) \in U} J_0 = \sum_{i=0}^{\infty} \gamma^i U_i$ and repeat this optimization in every time step with the increase of $k$. This treatment considers the effect of the control actions in the following $l$ steps and omits the control effect in the far future after $l$ steps to obtain an $l$ dimensional approximation. However, a large value of $l$ is often necessary to reach a satisfactory approximation and it often leads to an intensive computation, which is not suitable for online control of fast-varying systems. In addition, note that the dynamic model of the chaotic system appears in the optimization problem (5a) and it will also show up in the finite dimensional relaxation of the problem, which means the resulting solution requires model information and thus is also model-dependent.

4. Model-free control of chaos

In this section, we present the strategy to solve the constrained optimization problem efficiently without knowing the model information of the chaotic system. We first investigate the optimality condition of (5a) and present an iterative procedure to approach the analytical solution. Then, we analyze the convergence of the iterative procedure and the stability with the derived control strategy.
4.1. Optimality condition

Denoting $J^*$ the optimal value to the optimization problem (5a), i.e.,

$$J^* = \min_{u(0),u(1),\ldots,u(\infty) \in \Omega} J_0$$

subject to : (5b), (5c)

According to the principle of optimality [30], the solution of (5a) satisfies the following Bellman equation:

$$J^*(y) = \min_{u(k)} (U_k + \gamma J^*(z)) \quad \forall x \forall k = 0, 1, 2, \ldots$$

(7)

where $z$ is the solution of (5a) at $t = k + 1$ with $x(k) = y$ and the control action $u(t) = u_k$ for $k \tau \leq t < (k + 1)\tau$. Without introducing confusion, we simply write Eq. (7) as follows,

$$J^* = \min(U_k + \gamma J^*)$$

(8)

Define the Bellman operator $B$ relative to function $h(z)$ as follows,

$$Bh(z) = \min(U_k + \gamma h(z))$$

(9)

Then, the optimality condition (8) can be simplified into the following with the Bellman operator,

$$J^* = BJ^*$$

(10)

Note that the function $U_k$ is implicitly included in the Bellman operator. Eq. (10) constitutes the optimality condition for problem (5a). It is difficult to solve the explicit form of $J^*$ analytically from (7). However, it is possible to get the solution by iterations. We use the following iterations to solve $J^*$,

$$J(n + 1) = BJ(n)$$

subject to : (5b), (5c)

(11)

The control action keeps constant in the duration between the $k$th and the $k + 1$th step, i.e., $u^*(t) = u_k$ for $k \tau \leq t < (k + 1)\tau$. $u_k$ can be obtained from (7) based on (11),

$$u_k = \arg\min_{u(k) \in \Omega} (U_k + \gamma J^*)$$

(12)

Remark 2. The proposed control strategy uses a different way from the model predictive control method to stabilize the Lorenz chaotic system. We summarize the differences between the proposed method and the model predictive control method as follows,

- The model predictive control method often uses a finite time horizon approximation of the cost function to obtain a numerical solution of the problem. The proposed method uses an iterative method to solve the Bellman equation.
- The proposed method does not have an explicit approximate model for the system model and cannot give estimations of the input in the far future while the desired input in the far future can be estimated based on the estimated system model for the model predictive control method.
- The model predictive control method often uses a long time horizon input–output series in each step to estimate the system model while the proposed method only need the information in the last step for the update.
- The model predictive control method regards the cost function in finite time horizon as an objective function and find solutions by solve this optimization problem under some additional constraints. The decision variables is the input signal to be optimized. This procedure needs to run all the time. Differently, the iterative procedure (11) aims to find the optimal cost function. The optimal cost function only depends on the initial state and is invariant with time. The iterative procedure (11) can progressively approaches the desired optimal cost function.

4.2. Convergence of the iteration

In this part, we investigate the convergence of the iterative procedure given by (11).

Before presenting the convergence result, the following lemma about the contraction property the Bellman operator is stated below,

Lemma 1 [39]. The Bellman operator is a $\gamma$ contraction mapping in the infinity norm, i.e.,

$$\|Bh_1 - Bh_2\|_{\infty} \leq \gamma \|h_1 - h_2\|_{\infty}$$

(13)

The following lemma, known as the contraction mapping principle, is useful for the convergence analysis of the iteration (11).

Lemma 2 [40]. Let $(S, d)$ be a complete metric space and $T : S \rightarrow S$ be a mapping such that

$$d(T(x), T(y)) \leq cd(x, y)$$

for some constant $c < 1$. Then, $T$ has a unique fixed point in $S$.
for some \( 0 < c < 1 \) and all \( x \) and \( y \) in \( S \). Then \( T \) has a unique fixed point in \( S \) such that \( x' = T(x') \). Moreover, for any \( x_0 \in S \) the sequence of iterates \( x_0, T(x_0), T^2(x_0), \ldots \) converges to the fixed point \( x' \).

Now, we are ready to state the convergence result on (11).

**Theorem 1.** For any \( J(0) \), the iteration (11) converges to \( J' \) such that \( J' = BJ' \) for \( 0 < \gamma < 1 \).

**Proof.** The result directly follows Lemma 1 and Lemma 2. According to Lemma 1, the Bellman operator \( B \) is a \( \gamma \) contraction mapping in the infinity norm, which yields,

\[
d(B(x), B(y)) \leq \gamma d(x, y)
\]

with \( d(x, y) = \|x - y\|_{\infty} \) and \( 0 < \gamma < 1 \). With Lemma 2, we conclude that the sequence \( J(0), B(J(0)), B(B(J(0))), \ldots \) converges to the fixed point \( J' \), which is the solution of \( J' = BJ' \). Clearly, the sequence can be generated by the iteration (11) initialized with any \( J(0) \). This completes the proof. \( \square \)

4.3 Control Performance

Stability is often the first consideration in most control designs. It requires all trajectories generated by the dynamic system travel to the origin eventually. In real applications, this requirement is often relaxed to ultimately boundedness due to the presence of additive noises, perturbations, modeling errors, etc. In this part, we consider ultimately boundedness of the control strategy (12) with the utility function (3), which means all trajectories converges to a small vicinity of the origin eventually.

About the boundedness of the eventually states, we have the following theorem,

**Theorem 2.** If there exists a control law \( u = u(x_1, x_2, x_3) \), such that the chaotic system (1a) stabilizes to the origin asymptotically, the system (1a) under the control input (12) with the utility function (3) ultimately converges to the set \( S = \{x = (x_1, x_2, x_3) \in \mathbb{R}^3, |x_1| < \delta_1, |x_2| < \delta_2, |x_3| < \delta_3 \} \) (\( \delta_1 > 0, \delta_2 > 0, \delta_3 > 0 \)).

**Proof.** Since there exists a control law \( u = u(x_1, x_2, x_3) \) to stabilize the chaotic system (1a) to the origin asymptotically, we conclude that there exists a class \( K \) function \( \beta(\cdot, \cdot) \), such that \( \|x(t)\| \leq \beta(||x(0)||, t) \) \( \forall t > 0 \) [41]. According to the property of the class \( K \) function that \( \lim_{t \to \infty} \beta(||x(0)||, t) = 0 \). Therefore, we can always find a time \( t_1 > 0 \) for any constant \( c > 0 \), such that \( \beta(||x(0)||, t) < c \) \( \forall t > t_1 \). Choosing \( c = \min\{\delta_1, \delta_2, \delta_3\} \) yields \( \|x(t)\| \leq \beta(||x(0)||, t) \leq \min(\delta_1, \delta_2, \delta_3) \) \( \forall t > t_1 \). Accordingly, \( \|x_1(t)\| \leq \|x(t)\| \leq \delta_1, \|x_2(t)\| \leq \delta_2, \|x_3(t)\| \leq \delta_3 \) for \( t > t_1 \). Define \( k_1 \) as the least integer greater than \( \frac{t_1}{T} \). Then, for any \( n > k_1 \), we have \( U_n = 0 \) for the utility function with the definition (3). Therefore, the cost function for this control law is \( J_1 = \sum_{i=0}^{k_1} g_{i}/U_i = \text{constant}. \)

Now we consider the cost function \( J_2 \) under the control input (12). As \( J_2 \) is the minimum one under the constraints, \( J_2 \leq J_1 \). Also note that \( J_1 \) is upper bounded and \( J_2 \) is the summation of non-negative values, we conclude that \( \lim_{n \to \infty} U_n = 0 \) for the control law (12) (otherwise if \( \lim_{n \to \infty} U_n \neq 0 \), \( J_2 \) will keep increasing and be ultimately unbounded). As there are only two possible values for \( U_n \), we must exist an integer \( k_2 \), such that \( U_n = 0 \) for all \( n > k_2 \). Recalling the definition of the utility function (3), we conclude that \( |x_1| < \delta_1, |x_2| < \delta_2, |x_3| < \delta_3 \) for \( n > k_2 \) (i.e., \( t > k_2T \)). This completes the proof. \( \square \)

**Remark 3.** As reported in many papers, the chaotic system can be stabilized by many different methods, such as OGY method [42], passivity based method [43,44], backstepping [45,46], differential geometry method [47], etc., which means there indeed exists a control law \( u = u(x_1, x_2, x_3) \) to stabilize the system (1a). Therefore, the conclusion drawn in Theorem 2 always holds, i.e., the system (1a) under the control law (12) does reach an ultimately bounded convergence.

5. Implementation considerations

In the last section, the iteration (11) is derived to calculate \( J' \) and the optimization (12) is obtained to calculate the control law. The iteration to approach \( J' \) and the optimization to derive \( u' \) have to be run in every time step in order to obtain the most up-to-date values. Inspired by the learning strategies widely studied in artificial intelligence [33,48,49], a learning based strategy is used in this section to facilitate the processing. After a enough long time, the system is able to memorize the mapping of \( J' \) and the mapping of \( J(0) \). After this learning period, there will be no need to repeat any iterations or optimal searching, which will make the strategy more practical.

Note that the optimal cost \( J' \) is a function of the initial state. Counting the cost from the current time step, \( J' \) can also be regarded as a function of both the current state and the optimal action at current time step according to (8). Therefore, \( J(n) \), the approximation of \( J' \), can also be regarded as a function relative to the current state and the current optimal input. As to the optimal control action \( u' \), it is a function of both the current state and the current input. Our goal in this section is to obtain the mapping from the current state and the current input to \( J(n) \) and the mapping from the current state to the optimal control action \( u' \) using
parameterized models, denoted as the critic model and the action model, respectively. Therefore, we can write the critic model and the action model as \( f_c(x_n, W_c) \) and \( u_n(x_n, W_a) \) respectively, where \( W_c \) is the parameters of the critic model and \( W_a \) is the parameters of the action model.

In order to train the critic model with the desired input–output correspondence, we define the following error at time step \( n + 1 \) to evaluate the learning performance,

\[
e_n(n+1) = E(n) - \hat{E}(n+1) = \frac{1}{2} \varepsilon_n^2(n+1)
\]

Note that \( \hat{E}(n) \) is the desired value of \( E(n+1) \) according to (11). Using the back-propagation rule, we get the following rule for updating the weight \( W_c \) of the critic model,

\[
W_c(n+1) = W_c(n) + \Delta W_c(n) = W_c(n) - l_c(n) \frac{\partial E(n)}{\partial W_c(n)} = W_c(n) - l_c(n) \frac{\partial E(n)}{\partial \hat{J}(n)} \frac{\partial \hat{J}(n)}{\partial W_c(n)}
\]

where \( l_c(n) \) is the step size for the critic model at the time step \( n \).

As to the action model, the optimal control \( u' \) in (12) is the one that minimizes the cost function. Note that the possible minimum cost is zero, which corresponds to the scenario with the state staying inside the desired bounded area. In this regard, we define the action error as follows,

\[
e_a(n) = \hat{J}_n
\]

\[
E_a(n) = \frac{1}{2} \varepsilon_a^2(n)
\]

Then, similar to the update rule of \( W_c \) for the critic model, we get the following update rule of \( W_a \) for the action model,

\[
W_a(n+1) = W_a(n) - l_a(n) \frac{\partial E_a(n)}{\partial W_a(n)} \frac{\partial \hat{J}(n)}{\partial u(n)} \frac{\partial u(n)}{\partial W_a(n)}
\]

where \( l_a(n) \) is the step size for the action model at the time step \( n \).

Eq. (15) and Eq. (17) update the critic model and the action model progressively. After \( W_c \) and \( W_a \) have learnt the model information by learning for a long enough time, their values can be fixed at the one obtained at the final step and no further learning is required any longer, which is in contrast to Eq. (12) requiring to solve an optimization problem even after a long enough time.

6. Simulations

In this section, we consider the simulation implementation of the proposed control strategy. For the Lorenz system (1a), we choose \( \sigma = 10 \), \( \rho = 97 \) and \( \beta = 8/3 \). With this set of parameters, the uncontrolled Lorenz system with \( u = 0 \) in (1a) is chaotic and a typical trajectory and the corresponding time profiles are shown in Figs. 1 and 2. From the figures, it can be observed that the values of \( x_1, x_2 \) and \( x_3 \) vary in \((-50, 50), (-100, 100) \) and \((0, 100) \) respectively. The control objective is to regulate the absolute value of \( x_1 \), the absolute value of \( x_2 \) and that of \( x_3 \) to be less than 0.5. Accordingly, we choose \( \phi_1 = \phi_2 = \phi_3 = 0.5 \) for the utility function (3). The feasible control action set \( \Omega \) in (5a) is defined as \( \Omega = \{ u \in \mathbb{R}, u = \pm 50 \} \). This

![Fig. 1. A typical trajectory of the uncontrolled Lorenz system.](image-url)
definition corresponds to the widely used bang-bang control in industry. To make the output of the action model within the feasible set, the output of the action network is clamped to 50 if it is greater than or equal to zero and clamped to $-50$ if less than zero. The sampling period $\tau$ is set to 0.005 seconds. The discount factor $\gamma$ is chosen as 0.5. Both the critic model and the action model are linearly parameterized. The step size of the critic model, which is $\delta_c(n)$ and that of the action model, which is $\delta_a(n)$ are both set to 0.03. Both the update of the critic model weight $W_c$ in (15) and the update of the action model weight $W_a$ in (17) last for 30 s.

To validate the effectiveness of the proposed method, the output of the action model is used as the control input for the Lorenz system (1a). A typical trajectory with the proposed method is shown in Fig. 3, from which we can observe that the
Lorenz chaotic system is successfully regulated to the region with $|x_1| < 0.5$, $|x_2| < 0.5$ and $|x_3| < 0.5$ as expected (at time $t = 10\ s$, the state value is $x(10) = [-0.0043, 0.0325, 0.0001]$). Fig. 4 shows trajectories of the Lorenz system with different initializations under the control law obtained by the proposed method. Clearly, all of them convergence to the desired region. It is worth noting that in Fig. 4, some trajectories takes a relatively long path (especially for the trajectories in subfigures (a) and (d)) to the origin, instead of going there directly, to take advantage of the favorable flows of the inherent dynamics of the Lorenz chaotic system.

7. Conclusions

In this paper, the model-free control of Lorenz chaos is considered. The control problem is formulated from the optimal control perspective and solved via iterative methods. In contrast to existing models, this method does not need pre-knowl-
edge on the mathematic model of the chaotic system. The convergence of the iteration and the stability of the obtained control law are both proven in theory. The critic model and the action model are introduced to make the method more practical. Simulations show that the control law obtained by the proposed method indeed achieves the control objective and also demonstrates tendency to take advantage of favorable flows.

Acknowledgements

The authors would like to acknowledge the support by the National Science Foundation of China under Grant NSFC: 61105090.

References