Reachability Analysis of Nonlinear Hybrid Systems with Mixed Contacting

Ren Shengbing, Liu Yuan, Huang Fei, Jia Mengyu

***Abstract*—**How to calculate the intersection set of flows and guards quickly and accurately is at the core of nonlinear hybrid systems reachability analysis. The continuous flow condition of dynamic system of nonlinear hybrid systems are usually expressed by the implicit differential equations. Interval Taylor method with QR decomposition method is adopted in this paper that can explicitly express the boundaries of flows. Then the flows and guards can be expressed as Constraint Satisfaction Problem (CSP for short). At this moment, using the mixed contracting algorithms to solve the CSP that can calculate the intersection set of flows and guards. In the mixed contracting algorithm, when contracting the domain, single contractor is no longer used, but according to the use cycle to select the corresponding contractor. In the contraction domain, the strong but time-cost contractor and the weak but time-save contractor will be used interchangeably, that can improve the computing accuracy and reduce the time-consuming. The experimental results show that this method can quickly and accurately solve the intersection set of flows and guards.

***Keywords*—**nonlinear systems, hybrid systems, reachability analysis, interval analysis.

I. INTRODUCTION

Reachability analysis is an important method in the analysis and verification of nonlinear hybrid systems. Reachability analysis plays an important role in many safety verification problems of nonlinear hybrid systems, such as self-driving vehicles, Intelligent Expressway, Industrial Automation, etc. Through the analysis of the reachability of the nonlinear hybrid system, the security of the controller in the system can be verified, and the security of the system is verified [1].

Reachability analysis refers to whether a system in a given initial state can reach a certain target state after a period of time and condition. When calculating the reachability set of nonlinear hybrid system, the most important step is to calculate the reachability set of continuous system in known discrete mode, then calculate the intersection set of reachability set and guard condition. Literature [2-4] use convex polyhedra, ellipsoid, level sets and other mathematical forms to represent the state domain of the system respectively, then the intersection between the flow condition and the guard condition is calculated by the corresponding calculation method. But in this way, the expression of the state domain is not precise enough, and because of the high computational complexity, so that can't be applied to high dimensional hybrid systems. In literature [5] the idea of bounded model detection (BMC) is introduced , and the main method is to encode the reachable set of Hybrid Automaton (HA) in k - step by using satisfiability modulo theories (SMT) .This method lacks a certain degree of integrity, and the memory requirements increase with the size of the problem, so this method limits the size of the problem. In literature [6] the support function and other mathematical models are used to express and calculate the reachable domain. Although this method can deal with the computation of continuous state effectively, it needs very complicated geometric calculation when calculating intersection set. In literature [7], the guard set is represented by a half space (Halfspace), and the intersection set are over approximated by nonlinear mapping, so the geometric operation of the intersection set can be avoided. On this basis, literature [8] explicitly expresses the boundary of the continuous conditions by using the interval Taylor method (Interval Taylor Method, ITM). In this process, the QR decomposition method proposed by Lohner is used to control the wrapping effect. At this moment, the guard condition and the flow condition can naturally be expressed as a constraint satisfaction problem （CSP）that the 3B algorithm is used to solve the CSP, and the solution of flow constraint and guard constraint is obtained. Although the 3B algorithm has high accuracy in calculating CSP, it takes a long time. So, in the literature [9] HC4 algorithm is used to solve the CSP which consists of flow condition and guard condition. Although the speed has improved, the accuracy of the calculation is not high compared with that of literature [8]. Because of the HC4 algorithm, literature [9] cannot handle the case where the number of constrained variables occurrences is greater than one.

This paper proposes an improved scheme based on the above references. Solving the intersection of flow conditions and guard conditions. Firstly, the interval Taylor method with Lohner’ QR decomposition is used to express the flow condition explicitly. Then the mixed contracting algorithm is used to solve the CSP expressed by flow condition and guard condition. In this paper, the mixed contraction method combines LRC and 3B contraction operators. The contract power of LRC is strong but time consuming, the contract power of 3B is weak but short time consuming[10].when contract the

Ren Shengbin is with the Department of Software, Central South University, Changsha 410205, Hunan, China.

Liu Yuan is with the Department of Software, Central South University, Changsha 410205, Hunan, China (corresponding author; e-mail: 570467464@qq.com).

Huang Fei is with the Department of Software, Central South University, Changsha 410205, Hunan, China.

Jia Mengyu is with the Department of Software, Central South University, Changsha 410205, Hunan, China.

contracting domain, single contractor is no longer used, but according to the periodicity to select the corresponding contractor. By adjusting the periodicity of contractors to improving the accuracy and reducing the calculation time of the mixed contracting algorithm. Experiments show that the proposed algorithm is better than the literature [7] and [8], the algorithm is more accurate and takes less time.

II. Procedure for Paper Submission

This section will cover the concepts including nonlinear hybrid automata, interval analysis and constraint satisfaction problems.

A. Nonlinear Hybrid Automata

In the problem of reachability analysis of nonlinear hybrid systems, nonlinear hybrid automata are usually used for mathematical modeling [11]. A hybrid automaton is usually represented by a tuple $<Q,X,E,F,I,G,R,Init>$ where:

$Q=\{q_1,q_2...q_m\}$ $(m\in N)$ represents a set of location, i.e., discrete state or modes.

$X=\{x_1,x_2...x_n\}$ represents continuous variables.

$E\subseteq Q\times Q$ represents a finite set of discrete state transitions.

$F=\{flow_q\}_{q\in Q}$ represents a set of flow constraints.

$I=\{Inv_q\}_{q\in Q}$ represents a set of invariants

$G=\{grd_e\}_{e\in E}$ represents a set of guard constraints.

$R=\{rst_e\}_{e\in E}$ represents a set of reset constraints.

$Init=(q_0,x_0)$, $q_0\in Q$, $x_0\in X$ represents the initial states of hybrid automata.

The representation of flow condition, invariant condition and guard condition in nonlinear hybrid system is shown in formula (1) to formula (3):

$$flow_q: \quad x'(t)=f_q(x,p,t) \qquad (1)$$

$$Inv_q: \quad v_q(x(t),p,t)<0 \qquad (2)$$

$$grd_e: \quad r_e(x(t),p,t)=0 \qquad (3)$$

B. Interval Analysis

$U=[\underline{u},\overline{u}]\in IR$ is a real interval when it is a subset of real numbers set[12]. $\underline{u},\overline{u}$ represents the upper and lower limits of the interval U .A vector whose component is an interval is called an interval vector (also known as a box)and the value is the Cartesian product of components, $[U]=[u_1]\times[u_2]...[u_n]\in IR^n$ .A matrix whose elements are interval called as interval matrix, $[x]\in IR^{m\times n}$.

The width, midpoint and absolute value of the interval are defined as $w(u)=\overline{u}-\underline{u}$, $m(u)=\frac{1}{2}(\underline{u}+\overline{u})$, $|U|=\max\{|\underline{u}|,|\overline{u}|\}$ [13].For any interval $[u]=[\underline{u},\overline{u}],[v]=[\underline{v},\overline{v}]$,the expression of its four principle operation is $[u]\circ[v]=\{x\circ y\,|\,x\in[u],y\in[v]\}$, $\circ\in\{+,-,*,/\}$.

Interval function is a function in which both independent variables and dependent variables are interval [14]. Assume $f:R^n\to R^m$ is a continuous real value function. The range of f on the interval vector $U\in IR^n$ is $f(U)$,write as $f(U)=\{f(u)|u\in U\}$,If there is a function F satisfy the $f(U)\subseteq F(U)$,then can see F is Interval extended functions for function f .Interval functions have many forms of expansion, such as natural extension, median expansion and so on. The natural extension forms are the most commonly used expansion form of interval function. If the function f is composed of four principle operations of the basic function. Then the variables in the function are replaced by the interval variables, base functions and operators are replaced by respective interval basic functions and interval operators, then the natural extension function of the function can be obtained.

C. Constraint Satisfaction Problem

The general form of a constraint satisfaction problem can be expressed as: given a set of objects and the constraints on their assignment, determine whether there is at least one set of assignments that satisfy all constraints[15].Solving interval CSP problem mainly includes the following two operations：

1)Bisection Bisect an n-dimensional box is to partition the box $[U]\in IR^n$ along one of its components x_j into two other n-dimensional boxes $L[U]$ and $R[U]$.

$$L[U]=[\underline{u}_1,\overline{u}_1]\times\cdots\times[\underline{u}_j,\frac{\underline{u}_j+\overline{u}_j}{2}]\times\cdots\times[\underline{u}_n,\overline{u}_n]$$

$$R[U]=[\underline{u}_1,\overline{u}_1]\times\cdots\times[\frac{\underline{u}_j+\overline{u}_j}{2},\overline{u}_j]\times\cdots\times[\underline{u}_n,\overline{u}_n]$$

2)Contraction Assume $[U]$ is a prior field of constraint satisfaction problem H which the solution is S .Contract H is to replace $[U]$ with a smaller field $[U']$ and keep the solution set unchanged. The algorithm used to contract H is referred to as the contraction algorithm, which is represented by C_s in this paper, $S\cap[U]\subseteq C_s([U])\subseteq[U]$.

III. Explicit Representation of Flow Conditions

In nonlinear hybrid systems, an ordinary differential equation with initial value (IVP-ODEE) is usually used to implicitly represent the flow conditions of a continuous system [16], as shown by formula (1). In this paper, CSP is used to solve the intersection of flow condition and guard condition, and the constraint expression of CSP is expressed explicitly. Therefore, it is necessary to transform the implicit expression of continuous system into explicit expression. In this paper, the interval Taylor method with Loher' QR decomposition is used for explicit transformation.

A. Interval Integration with Interval Taylor Method

$X(t;t_0,X_0)$ represents the solutions of the system at t originating from each initial state X_0 at t_0 .Define a time grid $t_0<t_1<t_2<\cdots<t_{nT}$,which is taken equally spaced, and h_j represents the step from t_j to t_{j+1} .By integration via interval Taylor methods, we can calculate the interval vector $[x_j], j=1\cdots n_t$, which contains all the solutions $X(t_j;t_0,X_0)$ of formula (1) at t_j .The calculation steps for this method are as follows:

1)Verify the existence and uniqueness of the solution of interval Taylor method.

2)Compute the prior field $[\tilde{x}_j]$. The prior field contains all the solutions $X(t;t_j,X_j)$ that satisfy (1) in $t\in[t_j,t_{j+1}]$.

3)Using the priori enclosure A to compute a tighter enclosure for the set of solutions at t_{j+1}.

B. Verify the Existence and Uniqueness of the Solution

The expansion of interval Taylor series is

$$U(t,[\tilde{x}_j^0])=[x_j]+\sum_{i=1}^{k-1}[x_j]_i(t-t_j)^i+f^{[k]}([\tilde{x}_j^0])(t-t_j)^k \qquad (4)$$

The interval $[\tilde{x}_j^0]$ can be calculated by the formula (5)

$$[\tilde{x}_j^0]=[x_j]+f([x_j])[0,h] \qquad (5)$$

If we define

$$\begin{aligned}[\tilde{x}_j^1]&:=U([t_j,t_{j+1}],[\tilde{x}_j^0])\\ &=[x_j]+\sum_{i=1}^{k-1}[x_j]_i[0,h_j]^i+f^{[k]}([\tilde{x}_j^0])[0,h_j^k]\subseteq[\tilde{x}_j^0]\end{aligned} \qquad (6)$$

then there exists a unique solution $x(t;t_j,x_j)$ that satisfies $x(t;t_j,y_j)\in U(t,[\tilde{x}_j^1])$ at $t\in[t_j,t_{j+1}]$ of(1). $U([t_j,t_{j+1}],[\tilde{x}_j^1])\subseteq U([t_j,t_{j+1}],[\tilde{x}_j^0])=[\tilde{x}_j^1]$ can be obtained from formula (6),so $[\tilde{x}_j]=U(t,[\tilde{x}_j^1])$ can be used as a priori field of $x(t;t_j,x_j)$ at $t\in[t_j,t_{j+1}]$.When the step h_j is small enough , the formula (6) can be established, h_j and ε can be obtained from formula (7) and formula (8)

$$h_j=\begin{cases}\left(\dfrac{\varepsilon\|x_j\|}{(k+1)\|f^{(k+1)}(x_j)\|}\right)^{\frac{1}{k+1}}, & \|x_j\|\ge h_j\|f(x_j)\|\\[2ex] \left(\dfrac{\varepsilon\|x_j\|}{(k+1)\|f^{(k+1)}(x_j)\|}\right)^{\frac{1}{k}}, & \text{other conditions}\end{cases} \qquad (7)$$

$$\varepsilon\approx\frac{\|w(U(t,[\tilde{x}_j^0]))\|}{\|U(t,[\tilde{x}_j^0])\|} \qquad (8)$$

C. Computing a Prior Enclosure

According above, we can see that to calculate a priori enclosur $[\tilde{x}_j]$ must calculate $[\tilde{x}_j^0]$ and $[\tilde{x}_j^1]$ first. The step size h_j and ε can be calculated by formula (7) and formula (8). The algorithm description is shown in figure 1:

```
Algorithm 1: computing a prior enclosure
input: $[x_j]$
out: $[\tilde{x}_j]$
1  compute $h_j$, $\varepsilon$, $[\tilde{x}_j^0]$
2  $[\tilde{x}_j^1]:=U([t_j,t_{j+1}],[\tilde{x}_j^0])$
3  If $[\tilde{x}_j^1]\subseteq[\tilde{x}_j^0]$ then
4        $[\tilde{x}_j^1]:=U([t_j,t_{j+1}],[\tilde{x}_j^0])$
5  Else
6        $[\tilde{x}_j^0]:=[\tilde{x}_j^1]$
7        $[\tilde{x}_j^1]:=U([t_j,t_{j+1}],[\tilde{x}_j^1])$
8        If $[\tilde{x}_j^1]\subseteq[\tilde{x}_j^0]$ then
9              $[\tilde{x}_j^1]:=U([t_j,t_{j+1}],[\tilde{x}_j^1])$
10       Else
11            If $h>\varepsilon$ then
12              $h=h/2$ ; goto 2
13            End-If
14       End-If
15 End-If
16 $[\tilde{x}_j]:=U(t,[\tilde{x}_j^1])$
```

Fig.1 Priori enclosure algorithm

In figure 1, the first line shows h_j, ε, $[\tilde{x}_j^0]$ can calculated according to formula (5), formula (7) and formula (8).the second line shows $[\tilde{x}_j^0]$, $[\tilde{x}_j^1]$ can calculated according to formula (6).Line 3-line 15 indicates whether or not $[\tilde{x}_j^1]$ belongs to $[\tilde{x}_j^0]$,if $[\tilde{x}_j^1]$ belongs to $[\tilde{x}_j^0]$, Then use the formula (5) to calculate $[\tilde{x}_j^1]$, else let $[\tilde{x}_j^0]$ be equal to $[\tilde{x}_j^1]$, $[\tilde{x}_j^1]$ be equal to $U([t_j,t_{j+1}],[\tilde{x}_j^1])$.At this moment, if $[\tilde{x}_j^1]$ belongs to $[\tilde{x}_j^0]$, then we can use formula (5) to calculate $[\tilde{x}_j^1]$ .Otherwise , judging whether the step size ε is greater than the required precision or not,if greater than the required precision,then bisect the step size and goto the second line, continue to loop until $[\tilde{x}_j^1]$ belongs to $[\tilde{x}_j^0]$.Line 16 indicates that a prior enclosure $[\tilde{x}_j]$ is calculated by formula (4) useing $[\tilde{x}_j^1]$.

D. Computing a Tighter Enclosure

The wrapping effect will to produced when we use the interval Taylor method to solve the IVP-ODE. This is because the set needs to be wrapped in an interval vector during the calculation, but in most cases, these sets cannot be wrapped precisely in an interval vector that produces the wrapping effect. Because of the wrapping effect may exists during each iteration, the width of the resulting interval may increase until it is not acceptable. In this paper, Lohner' QR decomposition method is used to eliminate the wrapping effect in the calculation process [17]. The main idea is to make the axis parallel to the longest edge of the interval vector by rotating the axis.

Considering 3.1 a priori enclosure $[\tilde{x}_j]$ can be obtained in the step j+1, a tighter enclosure $[x_{j+1}]$ which $[x_{j+1}]\subseteq[x_j]$ to be calculated now. According to the expansion of Taylor series we can acquire the formula (9) as following:

$$x_{j+1}=\hat{x}_j+\sum_{i=1}^{k-1}f^{[i]}(\hat{x}_i)h_j^i+f^{[k]}(x;t_j,t_{j+1})h_j^k+\left\{I+\sum_{i=1}^{k-1}J(f^{[i]};x_j,\hat{x}_j)h_j^i\right\}(x_j-\hat{x}_j) \qquad (9)$$

Where $J(f^{[i]};x_j,\hat{x}_j)$ is the Jacobi matrix of $f^{[i]}(x_j)$ at point $x_j+\theta_{il}(x_j-\hat{x}_j)$, $\theta_{il}\in[0,1](l=1,2,...,m)$ (l represents the l line of a matrix).

Define:

$$z_{j+1}=f^{[k]}(x;t_j,t_{j+1})h_j^k\in f^{[k]}([\tilde{x}_j])h_j^k=[z_{j+1}] \qquad (10)$$

$$S_{j+1}=mid([z_{j+1}]) \qquad (11)$$

$$\hat{x}_{j+1}=\hat{x}_j+\sum_{i=1}^{k-1}f^{[i]}(\hat{x}_j)h_j^i+s_{j+1} \qquad (12)$$

$$S_j=I+\sum_{i=1}^{k-1}J(f^{[i]};x_j,\hat{x}_j)h_j^i\in I+\sum_{i=1}^{k-1}J(f^{[i]};[\tilde{x}_j])h_j^i=[S_j] \qquad (13)$$

$$A_0=I,r_0=x_0-\hat{x}_0\in[r_0]=[x_0]-\hat{x}_0,\ \ \hat{x}_0=mid([x_0]) \qquad (14)$$

Bring the formula from (10) to (14)into formula (9)can get the formula(15) :

$$\begin{aligned}x_{j+1}&=\hat{x}_{j+1}+S_j([x_j]-\hat{x}_j)+z_{j+1}-S_{j+1}\\ &=\hat{x}_{j+1}+A_{j+1}\{(A_{j+1}^{-1}S_jA_j)r_j+A_{j+1}^{-1}(z_{j+1}-S_{j+1})\}\in\{\hat{x}_{j+1}+A_{j+1}r_{j+1}\,|\,r_{j+1}\in[r_{j+1}]\}\end{aligned} \qquad (15)$$

Define $[r_{j+1}]=(A_{j+1}^{-1}[S_j]A_j)[r_j]+A_{j+1}^{-1}([z_{j+1}]-S_{j+1})$ can get formula (16):

$$[x_{j+1}]=[\hat{x}_{j+1}]+A_{j+1}[r_{j+1}] \qquad (16)$$

The algorithm for calculating a tighter enclosure at $t \in [t_j, t_{j+1}]$ is shown in figure 2：

```
Algorithm 2: computing tighter enclosure
input: $[x_0], \hat{x}_0, t_0, [r_0], A_0, [\tilde{x}_j], t$
output: $[x_{j+1}]$
1 $[z_{j+1}] = f^{[k]}([\tilde{x}_j])h_j^k$
2 $S_{j+1} = mid([z_{j+1}])$;
3 $[\hat{x}_{j+1}] = [\hat{x}_j] + \sum_{i=1}^{k-1} f^{[i]}([\hat{x}_j])h_j^i + S_{j+1}$;
4 $[S_j] = I + \sum_{i=1}^{k-1} J(f^{[i]};[\tilde{x}_j])h_j^k$
5 use algorithm 3 to calculate $A_{j+1}$
6 $[r_{j+1}] = (A_{j+1}^{-1}[S_j]A_j)[r_j] + A_{j+1}^{-1}([z_{j+1}] - S_{j+1})$;
7 $[x_{j+1}] = [\hat{x}_{j+1}] + A_{j+1}[r_{j+1}]$
```

Fig.2 The algorithm of compute tighter enclosure

In algorithm 2, lines 1-4 shows $[z_{j+1}]$、S_{j+1}、$[\hat{x}_{j+1}]$、$[S_j]$ can calculated according to formula (10) to formula (13).Line 6 shows how to calculate $[r_{j+1}]$. In line 7, according to formula (15), the tighter enclosure $[x_{j+1}]$ of the flow condition at the time interval $[t_j, t_{j+1}]$ is obtained. In line 5 there needs to select the appropriate A_{j+1} that will be described in algorithm 3.

```
Algorithm 3: computating $A(t)$
input: $A_j, S_j, r_j$
output: $A(t), l_i$
1 $A(t) := Mid((S_j)A_j)$
2 $l_i = \|A_{j+1}^i\|_2 \cdot w([r_j]^i)$
3 $sort(l_i)$ and $A_{perm}(t) := Permute(A(t))$
4 compute $Q(t) \leftarrow A_{perm}(t) = Q(t)R(t)$
5 $A(t) = Q^{-1}(t)A_{perm}(t)r_j$
```

Figure 3 The algorithm of find the proper $A(t)$

The first line indicates that $A(t) = Mid((S_j)A_j) \in [S_j]A_j$ .The second line indicates that how to calculate the length of each side of the matrix $A[r]$. $\|A_{j+1}^i\|_2$ denotes the Euclidean norm of the i column of matrix A .Line 3 sort the length of $A[r]$ and use permutation matrix to permute it .In the fourth line, we use Lohner'QR decomposition method to decompose the matrix which permuted that can obtain $Q(t)$ .Line 5 shows how to calculate $A(t)$.

IV. Computing CSP Composed of Flow Condition and Guard Condition

A. Use CSP to Express the Flow Condition and Guard Condition

At t_e , if the flow condition intersects with the guard condition, that is, the continuous state $x(t_e)$ satisfies the guard condition, then the discrete transition $e = q \to q'$ will occur. Assuming that the flow condition is continuation at time $[t_j, t_{j+1}]$, it is possible to intersect with the guard condition at any time during that time period. Our goal is to find a set of $[\underline{t}^*, \bar{t}^*] \times [x_j^*(t)]$ .As shown in figure 4[1],where $[x_j^*(t)]$ represents the set of initial flow conditions that satisfy the guard condition in time period $[\underline{t}^*, \bar{t}^*]$ .By the previous calculation we can obtain a tight enclosure $[x_{j+1}]$ for the flow condition at time period $[t_j, t_{j+1}]$,so the set of solution flow conditions and guard conditions can be obtained by solving the following CSP shows as formula (17).

$$([t_j, t_{j+1}] \times [x_j], '\gamma_e(x_j(t)) = 0') \qquad (17)$$

If $[\underline{t}^*, \bar{t}^*] \times [x_j^*(t)]$ is not empty, then it can be considered that there is a discrete transition $e = q \to q'$ at $t_e = \underline{t}^*$ .In this paper, we first find priori enclosure of the flow condition at the time period $[t_i, t_{i+1}]$ .Determine whether 0 belongs to $r_e([\tilde{x}_j])$ to determine whether there is an intersection between the flow condition and the guard condition in the time period $[t_j, t_{j+1}]$ .If $0 \in r_e[\tilde{x}_j]$ that represents there have intersection,that means, by solving the CSP, we can find the set of flow conditions and guard conditions in this time period.Otherwise, continue to look for the intersections of flow conditions and guard conditions in the next time period.

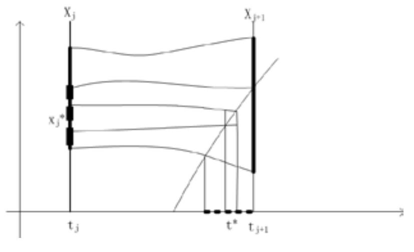


Fig.4 The intersection of flow with guard

B. Use Mixed Contracting Algorithms to Solve CSP

The CSP system can be represented as $S = (c, x, [x])$ [18],where c represents constraint, x represents variable, box $[x]$ denotes the value interval of a variable satisfying the constraint condition. Solving CSP is to find the set of boxes B_ε that the accuracy is less than the required, and the solution of the problem exists in these boxes. The algorithm for solving CSP which based on mixed contractor is shown in figure 4:

```
Algorithm 4: mixed contracting algorithm to solve CSP
input: $c, [x], C_s, T[C_s]$
output: $B_\varepsilon$
1 $L \leftarrow \{[x]\}$ , $I = 1$
2 While $L \neq \phi$ do
3   $[x] \leftarrow pop(L)$;
4   $([x^r], [x^l]) \leftarrow bisect([x])$;
5   For all $[x] \in \{[x^r], [x^l]\}$ do
6     For all $C_s$ do
7       If $I \% T[C_s] = 0$ then
8         $[x] \leftarrow C_s([x], c)$
9         If $[x] \neq \phi$ then
10          If $([x])$ is atomic box then
11            $B_\varepsilon \leftarrow B_\varepsilon \cup \{[x]\}$
12          Else
```

```
13                  $push(L,[x])$
14              End
15            End
16          End
17        End
18      End
19    $I = I + 1$
20  End
```

Fig.4 Mixed contracting algorithm to solve CSP

In algorithm 4, we need to input the initial interval $[x]$, the constraint conditions c, the contractors C_s and the hash table $T[C_s]$. $T[C_s]$ stores the operating cycles of each contractor.

For example, $T[LRC]=2$ means every two contracting nodes use LRC once. Line 1 of algorithm 4 indicates that the initial interval $[x]$ of each variable is stored in stack L ,Set the initial value of variable I to 1.Line 2-line 4 indicates that during each iteration, if the stack is not empty, an interval node is taken from the top of the stack and bisect it. Line 5-line 8 denotes contracting the bisected interval. Line 9-line 14 indicates whether the contracted interval has reached the required accuracy. If the accuracy is achieved, the interval is stored in the B_ε ,or pressed back into the stack and repeated the above steps. In line 7, for each contractor, determine whether $I\%T[C_s]$ is zero, and if it is 0, then this node can use the contractor. In this paper, LRC and 3B operators are chosen, because the LRC contraction operator is strong in contracting but takes a lot of time and 3B is weak in contracting but takes short time. By setting the use cycles of LRC and 3B, the accuracy of the contracting algorithm can be improved and the calculation time can be reduced that will be demonstrated experimentally later in this paper.

V. Experimental Results and Analysis

All experiments in this paper are implemented using IBEX tools, which is downloaded from http://www.ibex-lib.org. Hardware environment: Core i52.30GHz dual-core, Samsung DDR3L 1600MHz 8GB memory.

A. *Compare the Time Interval of The Intersection*

In order to compare the calculation of time interval which is calculated by each contracting algorithm. In this experiment, the ball bounce experiment was carried out in reference 1. In this experiment, the small ball can be regarded as a particle. The bounce surface is represented by a sinusoidal curve $y=\sin(2x)$. $p_x(t), p_y(t), v_x(t), v_y(t)$ represent the position and velocity of the particle in the direction of X,Y ,respectively. The time of the first three contact with the eject surface. Find out the time interval of the first three times contact with bounce surface. This nonlinear hybrid system is represented as formula（18）:

$$\begin{cases} \textit{flow:}\ p'_y(t)=v_y(t)\ ,\quad v'_x(t)=0\ ,\quad v'_y(t)=-g+kv_y^2(t) \\ \textit{guard:}\ \ \sin(p_x(t))-p_y(t)=0 \\ \textit{reset:}\ \ p_x=p_x,\ p_y=p_y\ ,\quad v_x=\dfrac{(1-e\cdot 4\cdot\cos(2\cdot p_x)^2)\cdot v_x+(1+e)\cdot 2\cdot\cos(2\cdot p_x)\cdot v_y}{1+\cos(p_x)^2} \\ \qquad\qquad\qquad\qquad v_y=\dfrac{(1+e)\cdot 2\cos(2p_x)\cdot v_x+(-e+4\cos(2\cdot p_x)^2)\cdot v_y}{1+4\cos(2\cdot p_x)^2} \\ \textit{inv:}\ \sin(p_x)-p_y\ge 0 \end{cases} \tag{18}$$

Among them, assume that $g=9.8$, $k=0.3$, $e=0.8$, the initial domain of $p_x(t), p_y(t), v_x(t), v_y(t)$ is $[2.0,2.1]\times[5.0,5.1]\times[0.0,0.1]\times[-5.1,-5.0]$, $h=0.05$, the threshold of the bisected time interval is $\varepsilon_t=10^{-11}$ $\varepsilon_z=0.2$ .Table 1 shows the time interval of the ball in contact with the bounce surface at the first, second, and third times that calculated by HC4 and 3B and mixed contracting algorithms.

Table 1 the time interval of each contracting algorithm

	t_1	t_2	t_3	S
Algorithm in literature[9]	0.56636310[267,534]	1.51931342[197,465]	2.68833630[669,917]	976
Algorithm in literature[9]	0.56636310[289,431]	1.51931342[221,396]	2.68833630[791,905]	4021
the use cycle of contractor is (1,4)	0.56636310[311,368]	1.51931342[256,316]	2.68833630[813,863]	793
the use cycle of contractor is (1,2)	0.56636310[302,397]	1.51931342[238,327]	2.68833630[806,889]	810

The first column shows we use the algorithm in the literature [9], literature [8] and mixed contracting algorithm that the use cycles of LRC and 3B are (1,4) and (1,2), respectively.
(1,4) means that each node uses LRC once, and every four nodes use 3B algorithm once. Since there are many permutations and combinations of LRC and 3B algorithms, only the two cases with the highest accuracy and the shortest computing time are listed in this table.

The 2-4 column represent the time interval that the ball first times to third times come into contact with the surface. Denote $w(t)=\bar{t}-\underline{t}$,the smaller the $w(t)$,the greater the accuracy. The last column, represents the total time of three calculations in milliseconds. It can be seen from the table that when the use cycle of the mixed contracting algorithms LRC and 3B is set to (1,4) and (1,2), the contracting force is larger than that of HC4 and 3B, and the time spent is less.

As you can see from the comparison of the last two lines. Although (1,4) is more accurate than (1, 2), it takes more time. So, it is necessary to choose between the calculation precision and the calculation time by using the mixed contracting algorithm in the calculation process.

B. *Compare the Intersection Interval of Flow Conditions States*

In this experiment, the experiment carried out from literature [8]. There is a hybrid system has two modes q_1 and q_2 ,a discrete transition $e=q_1\to q_2$ .Its expression is (19)

$$\begin{cases} \textit{flow}(1): & f_1(x_1,x_2)=(x_2,-px_2-g\sin(x_1)) \\ inv(1): & v_1(x_1,x_2)=\cos(x_1)-x_2/10-0.7 \\ \textit{flow}(2): & f_2(x_1,x_2)=(x_2,-3px_2-g\sin(x_1)) \\ inv(2): & v_2(x_1,x_2)=-v_1(x_1,x_2) \\ guard: & \gamma_1(x_1,x_2)=v_1(x_1,x_2) \\ reset: & \rho_1(x_1,x_2)=(\alpha_1x_1,\alpha_2x_2) \end{cases} \tag{19}$$

Where $\alpha_1=-1,\alpha_2\in[-2.05,-2],g=10,p\in[6,6.3],x_0\in[-0.9,-0.8]\times[3,3.5]$, $h=0.05$, $\varepsilon_T=0.005s$, $\varepsilon_z=0.2$ .Calculate the state of the flow condition when the flow condition intersects with the guard condition within 0-0.45 s, the results are shown in Table 2.

Table 2 the state interval of each contracting algorithm

	Algorithm in literature [9]	Algorithm in literature [8]	the use cycle of contractor is (1,4)	the use cycle of contractor is (1,4)
Volume	0.19504378019	0.12677795432	0.06592534891	0.08240567113
S (ms)	308	1384	280	273

The first row of table 2 shows the names of each contracting algorithm，and the last two are mixed shrinkage algorithms for the periods of LRC and 3B, respectively

The first line of table 2 shows the names of each contracting algorithm, and the last two represent the mixed contracting algorithms which the use cycle of LRC and 3B setting （1,2） and （1,4）, respectively. The second line indicates that the volume of the interval of intersection states is calculated by each contracting algorithm $\text{Vloume} = \prod_{i=1}^{n} w(x_i)$ .The third line represents the computation time of each shrinkage algorithm.

The table 2 show that the accuracy of the mixed contracting algorithm with LRC and 3B use cycle is set to (1, 4) and (1 , 2) is higher than only using HC4 and 3B algorithm, and the computation time is less than that of HC4 and 3B algorithm., when calculating the intersection between the flow condition and the guard condition

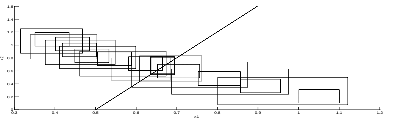
Through experiments 1 and 2, we can see that when the use cycle of LRC and 3B set at (1 , 4) and (1 , 2), the solution set $[\underline{t}^*, \overline{t}^*] \times [x_j^*(t)]$ of (8) is more accurate and takes less time than using HC4 and 3B algorithm alone.

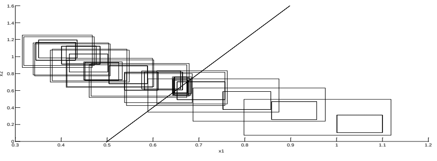
C. Comparison of The Flow Condition States of the Intersection Calculated

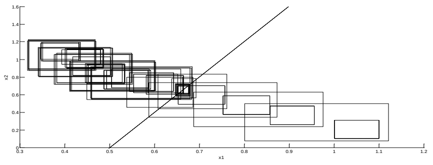
In this experiment, the experiment carried out from literature[9].There is a hybrid system have two modes q_1 and q_2 ,a discrete transition $e = q_1 \to q_2$ .Its expression is (20)

$$\begin{cases} flow(1): & f_1(x_1,x_2) = (1-(b_1+1)x_1 + a_1x_1^2x_2, b_1x_1 - a_1x_1^2x_2)) \\ inv(1): & v_1(x_1,x_2) = -4x_1 + x_2 + 2 \\ flow(2): & f_2(x_1,x_2) = (1-(b_2+1)x_1 + a_2x_1^2x_2, b_2x_1 - a_2x_1^2x_2)) \\ inv(2): & v_2(x_1,x_2) = -v_1(x_1,x_2) \\ guard: & \gamma_1(x_1,x_2) = v_1(x_1,x_2) \\ reset: & \rho_1(x_1,x_2) = (\alpha_1x_1, \alpha_2x_2) \end{cases} \quad (20)$$

Where $\alpha_1 = -1, \alpha_2 \in [-2.05,-2], g = 10, p \in [6,6.3], x_0 \in [-0.9,-0.8] \times [3,3.5]$, $h = 0.05$, $\varepsilon_T = 0.005s$, $\varepsilon_z = 0.2$ .Figure 3 (a)-(c)shows a comparison of the state $x_1 \times x_2$ in 0-0. 45 s.


（a）algorithm in literature [9]


（b）algorithm in literature [9]


（c）the mixed contracting algorithm with the use cycle set(1,4)

In the figure 3, the lateral axis and vertical axis of the horizontal axis represents x_1 and x_2 , respectively. An oblique line indicates the guard condition. The solid line box represents all possible ranges of values from 0-0.45s. The dotted line box represents the range of invariants. The solid line box which guard condition represents the intersection of the flow condition and the guard condition. In figure 3 (c) represents a mixed contracting algorithm that the use cycle of for LRC and 3B is setted (1,4). The calculation time used in figure 3(a)-(c) is: 324ms ,1324ms, 267ms, respectively. The volume is: 0.143546, 0.104943,0.647125 respectively. By contrast, we can see that using the mixed contracting algorithm to calculate the $x_1 \times x_2$ states in the intersection interval is more accurate than using the HC4 and 3B only. This is because in this method take the mixed contracting algorithm instead of applying a single contracting algorithm to all the contracting nodes. At the beginning, the weak contracting but less time consuming algorithm is used in the large contracting domain that improves the efficiency of the operation. When the contracting domain is reduced to a certain extent, a contracting algorithm with strong contracting but more time consuming is used in the region to improve the accuracy of the operation. Therefore, this mixed contracting algorithm can improve the efficiency and accuracy of the operation.

VI. Conclusions

In the reachability analysis of nonlinear hybrid systems, the most important thing is to determine when and when the continuous system will lead to the transition of discrete state, that is, to solve the intersection of flow conditions and guard conditions. In this paper, the method of interval Taylor is used to explicit expression the flow conditions and use Lohner' QR decomposition method to control the wrapping effect during the process. In this moment, the problem of solving the intersection of flow conditions and guard conditions can be converted into a CSP problem. Then, the mixed contracting algorithm is used to solve the CSP problem, which consists of flow condition and guard condition. The mixed contracting algorithm according to the pre-set use cycle of contractor to contract the contracting domain. In this way, the weak contractility but less time consuming contractor can be used in the large contracting domain that can improve the arithmetic speed. When the contracting domain is reduced to a certain extent, a contractor with strong contractility but much time consuming is used in the region to improve the accuracy of the operation. It can be seen from the experimental comparison that the method is more accurate and faster in calculating the time interval and the flow conditions state interval of the intersection between the flow condition and the guard condition. That means taking this method to determine when and what state the continuous system leads to discrete transformation, the results are more accurate and the calculation time is shorter.

References

[1] L.Bu, DB.Xie, “Formal Verification of Hybrid System”, *Journal of Software*, vol.25, no.2, pp.219-233,2014.

[2] Asarin, O.Bournez, T. Dang, et al., "Approximate reachability analysis of piecewise linear dynamical systems", in *Proc. Hybrid Systems: Computation and Control,London*,2000,pp.9-12.

[3] A.Kurzhanskiy, P.Varaiya, "Ellipsoidal techniques for reachability analysis of discrete-time linear systems", *IEEE Transactions on Automatic Control*, vol.52, no.1, pp.26-38, 2007.

[4] Lan M.Mitchell, Y. Susuki, "Level set methods for computing reachable sets of hybrid systems with differential algebraic equation dynamics", in *Proc. Hybrid Systems: Computation and Control, Berlin*, 2008, pp.630-633.

[5] D.Ishii, K.Ueda , H.Hosobe , et al., " An interval-based SAT modulo ODE solver for model checking nonlinear hybrid systems", *International Journal on Software Tools for Technology Transfer*, vol.13, no.5, pp.449-461,2011.

[6] Frehse G, "Flowpipe-Guard Intersection for Reachability Computations with Support Functions", in *Proc. Analysis and Design of Hybrid Systems, New York*, 2012, pp.94-101.

[7] M.Althoff, B H.Krogh ,"Avoiding geometric intersection operations in reachability analysis of hybrid systems", in *Proc. Hybrid Systems: Computation and Control, Beijing*, 2012, pp,45-54.

[8] N.Ramdani, N S.Nedialkov," Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint-propagation techniques ", *Nonlinear Analysis: Hybrid Systems*, vol.5,no.2,pp. 149-162.

[9] M.Maiga, N. Ramdani, and L. Trave-Massuyes, "A fast method for solving guard set intersection in nonlinear hybrid reachability", in *Proc. Decision and Control IEEE,* 2013, pp.508-513.

[10] I K.Kueviakoe, A.Lambert, P.Tarroux, et al., "Comparison of Interval Constraint Propagation Algorithms for Vehicle Localization", *Journal of Software Engineering and Applications*, vol.05,no.12,pp.157-162,2013.

[11] R.Alur, C.Courcoubetis, N.Halbwachs, et al., "The algorithmic analysis of hybrid systems", *Theoretical Computer Science*, vol.138, no.1, pp.3-34,1995.

[12] Moore R E, "Interval analysis", in *Proc.* Prentice Hall, New Jersey,1966, pp.166-168.

[13] M J.Cloud, B C.Drachman, L P.Lebedev, "A Brief Introduction to Interval Analysis", in *Porc. Springer International Publishing*, Berlin,2014, pp.179-193.

[14] E Hansen, "Methods and Applications of Interval Analysis", Siam Review, vol.23, no.1, pp.121-123 ,2012.

[15] G.Chabert, L.Jaulin, "Contractor Programming", *Artificial Intelligence*, vol.173, no.11,pp.1079-1100,2009.

[16] N S.Nedialkov. K.Jackson, Corliss G F, et al, "Validated solutions of initial value problems for ordinary differential equations", *Applied Mathematics and Computation*, vol.105, no.1, pp. 21-68,1999.

[17] R J.Lohner, "On the Ubiquity of the Wrapping Effect in the Computation of Error Bounds". in *Proc Perspectives on Enclosure Methods*, 2001, pp.201-217.

[18] I.Araya, R.Soto, B.Crawford, et al, "Adaptive filtering strategy for numerical constraint satisfaction problems", *Expert Systems With Applications*, vol.42,no.21,pp.8086-8094,2015.

Ren Shengbing was born on Aug. 5, 1969**.** He received the PhD degree in computer science and technology from Central South University of China. Currently, he is a assistant professor at Central South University of software, China. His major research interests include software engineering embedded system and image processing.

Liu Yuan was born on Apr. 24, 1990**.** She is a graduate student of Central South University of China. Her major research interests include information reachability analysis of nonlinear hybrid systems.

Huang Fei was born on 1993**.** She is a graduate student of Central South University of China. Her major research interests include in cyber physical systems.

Jia Mengyu was born on 199**.** She is a graduate student of Central South University of China. Her major research interests include in use case slice.