Improved flowpipe/guard intersections for hybrid reachability using Taylor models

Mentors: -Marcelo Forets, David P. Sanders, Louis Benet.

Deshmukh Aadesh Suhas
Indian Institute of Information and Technology Trichy
Tamil Nadu
UTC + 05:30

Contact Information
Email -aadeshdeshmukh25@gamil.com
Github-https://github.com/Aadesha

Overview

When we have a dynamical system, given an initial condition one can find out the motion of the system by solving the ordinary differential equation (ODE).
The goal of reachability analysis is to build an enclosure of the ODE given a set of initial conditions and uncertain inputs. All trajectories are guaranteed to be contained in the "flowpipe" obtained with set-based methods or optimization.
Taylor method is one approach to construct a guaranteed solution of the ODE, adding

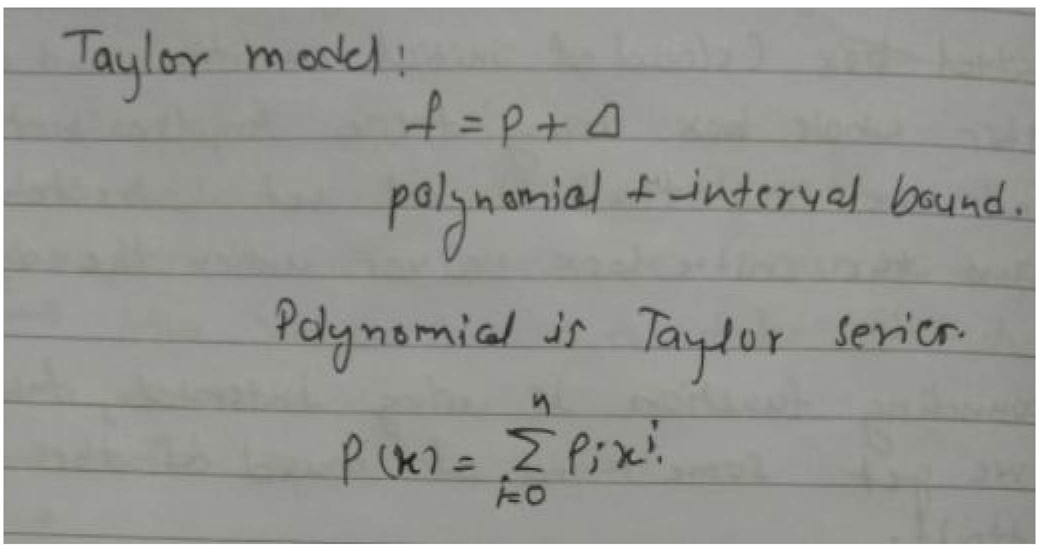
interval bounds that are proven to enclose the true solution.

Interval (closed) arithmetics allows defining mathematical operations so the result is a new interval and that new interval contains the true solution.

Now if we have the composition of functions the result will be still true.

Taylor Models(TM):-

To get a more accurate approximation of function one can use the polynomials here, by adding the interval to the polynomial we can bound the functions. The polynomial here is the Taylor series of degree n.



To find the bound over the polynomial in the box. so evaluation of polynomial is done by using interval arithmetics and elementary function will be expanded using Taylor series with Lagrange remainder.

bound f using Taylor series with Lagrange.

$$f(x) = \sum_{i=0}^{n} \overbrace{\frac{f^{(i)}(0)}{i!}}^{p(x)} x^{i} + \frac{\overbrace{f^{(n+1)}}^{\Delta}(\xi)}{(n+1)!} x^{n+1} \longrightarrow \text{lagrange remainder.}$$

$f^{(i)} = i^{th}$ derivative of f

$\xi \in [0, x]$

Lagrange form of the remainder is the infinite tail of the Taylor series can be written as the n+1 derivative evaluated at some point in an interval which also bound with use of interval arithmetic.

Taylor Method:-

ODE can be solved using the Picard iteration.

$$\dot{x} = f(x)$$

$$u_{n+1} = x_0 + \int_{t_0}^{t} f(u_n(s))\,ds.$$

Finding the solution which guaranteed to contain the exact solution of initial value problem, Taylor Model solution contain the exact solution of the equation and solution exists for some time interval only. To validate the solution we satisfy the delta test and enclosure relation.

Reachable State Computation for Hybrid Systems:-

When we have continuous variables and discrete transitions (sudden changes in ODEs) such a system is called a hybrid system and can be modeled by hybrid automata.

A state then represents a discrete mode and valuation of the continuous variables. given that the system is in a mode 'i', the variable values are changed continuously according to the ODE associated with the state 'i' or discretely by a jump starting from 'i'.

Now the goal is to find all reachable sets over bounded time and finitely many jumps, to compute all possible behaviors of the hybrid system.

Domain Contraction:-

Given a Taylor model flowpipe over domain and Guard set we can evaluate a valid Contraction of the domain. So we consider a class of contraction set to be an interval and the smallest valid interval can be evaluated. The lower bounds of it in every dimension as well as the value of time step can also be derived using the optimization of following Problem.

$$G = \{ \vec{x} \in \mathbb{R}^n \mid \wedge_{i=1}^{m} g_i(\vec{x}) \leq 0 \}$$

where $g_1, \ldots g_n$ are analytic function

$$F = (p(\vec{x}_0, t), I) \quad \vec{x}_0 \in X$$

$$F \cap G = \{ \vec{x} \mid \vec{x} = p(\vec{x}_0, t) + \vec{y} \wedge \vec{x}_0 \in X_0 \wedge t \in [0, \delta] \wedge \bigwedge_{i=1}^{m} g_i(\vec{x}) \leq 0 \}$$

-lower bound

$$\inf\{z\} \text{ subject to } \vec{x} = p(\vec{x}_0, t)$$

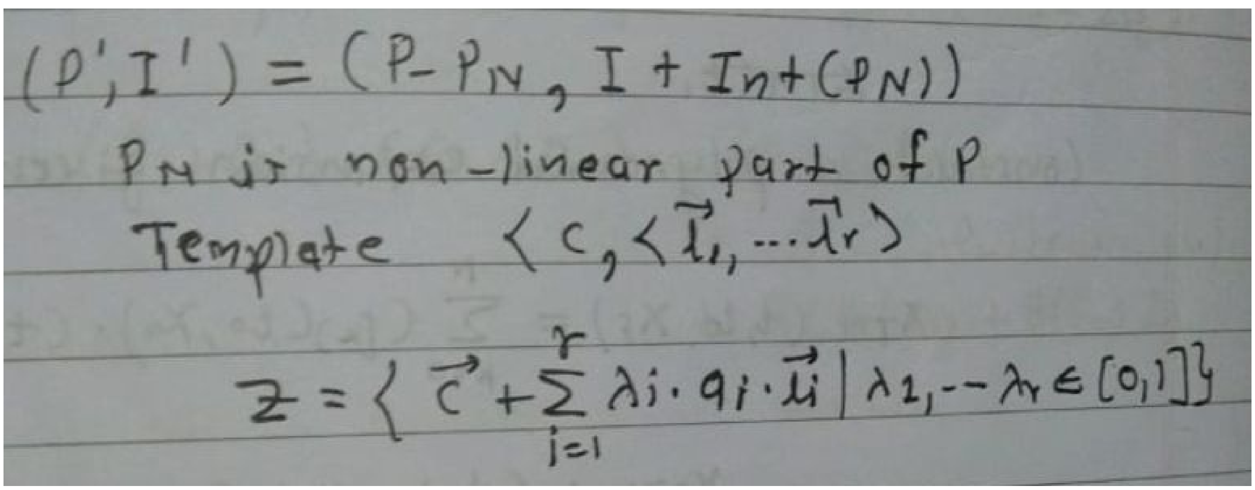
$$X = \{ \vec{x}_0[1], \ldots \vec{x}_0[d] \} \;\&\; z = t$$

Interval Constraint Propagation (ICP):-

The technique of ICP is a combination of interval arithmetic and constraint propagation to conservatively solve a Constraint Satisfiability Problem (CSP). That is to compute a superset of the exact solution set. It often costs much less time than finding the exact set and returns a fairly tight enclosure.

Zonotope Over-Approximation:-

The flowpipe can be enclosed by zonotope, A effective way to it is to perform a conservative linearization on TM and equivalently translating the TM into Zonotope.



$$(P', I') = (P - P_N, I + Int(P_N))$$

P_N is non-linear part of P

Template $\langle c, \langle \vec{l}_1, \dots \vec{l}_r \rangle$

$$Z = \{ \vec{c} + \sum_{i=1}^{r} \lambda_i \cdot q_i \cdot \vec{l}_i \mid \lambda_1, \dots \lambda_r \in [0,1] \}$$

Goals

In TaylorModels.jl and Reachability.jl there is an initial implementation of hybrid reachability, but the intersections with the guards are performed after the over-approximation. The goal of this project is to improve the way that this is handled, to have more accuracy and scalability.

1.Flowpipe /guard intersection algorithm (domain contraction)

a> Branch-and-prune algorithm.

b> Efficient constraint satisfiability problem.

c>Lower bound approximation search.

d>Upper bound approximation search.

2.Range over-approximation (computing an enclosure of the TM flowpipe)

a>Zonotope over-approximation.

3.Framework of the flowpipe construction (using 1 and 2)

a>Reachable set computation for hybrid automata

b> Flowpipe construction for hybrid automata

4.Benchmark evaluation

a> Spiking neurons

b>Nonlinear transmission line circuit

c>Space Rendezvous Benchmark

Timeline (UTC +05:30)

Pre GSoC Period

1.April 10 - May 6

Read the papers mentioned in the references section.It will help me learn technicals in depth, which is necessary for implementing methods, proposed in this project. Also, I will get myself familiar with the code base of TaylorModes.jl, Reachability.jl, HybridSystems.jl and IntervalArithmetic.jl

2.May 7 - May 26 (Community Bonding Period)

Discuss the plan of action with the community. I have found the community on Gitter and Slack to be very helpful based on my interactions so far. Adapt the plan based on the feedback.

GSoC Period

3.May 27 - June 2 (Week 1)

Work begins. I shall begin first by porting the TMJets algorithm to reachability to rewrite validated_integ(JuliaReach/Reachability.jl#602) in Reachability. To return the polynomials, then enclose the polynomials using different evaluation methods.

4.June 3 - June 16 (week 2&3)

By this time I shall start moving towards the main goals. I shall implement domain Contraction methods.(JuliaReach/Reachability.jl#609)

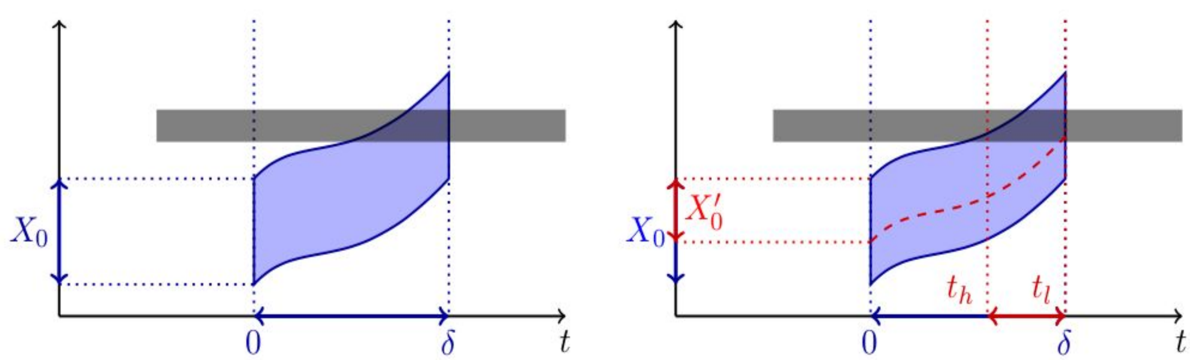
a>Given a Taylor model flowpipe over domain and guard set method should Calculate the valid contraction of Domain such that it will produce the Computable Taylor model after the intersection with a guard.

b>convex optimisation will be implemented by interior point method. If the problem is not convex then convex relaxation will be performed.

c> Interval constraint propagation will be used in constraint satisfiability Problem which is a valuation V :X->R of the variables such that all constraint are Satisfied, Then the domain contraction task becomes computing an interval enclosure of the solution.

1>Branch-and-prune algorithm. (week 2)

2>Efficient constraint satisfiability problem. (week 3)



5.June 24 - June 30 (Week 5)

Catch up with any unfinished work left in the past weeks.

preparation for **Phase 1** Evaluation Submission Deadline which is on June 28.

I shall write a post describing my work and my experience so far.

6.June 17 - June 23 & July 1-July 7(Week 4&6)

I shall implement the contraction in each dimension by searching approximations for the exact lower and upper bounds.

a>split the domain at its midpoint .

b>update the solution domain in the choose dimension.

c>check whether the solution is in the contracted domain if no solution is there then discard the branch and update the interval.

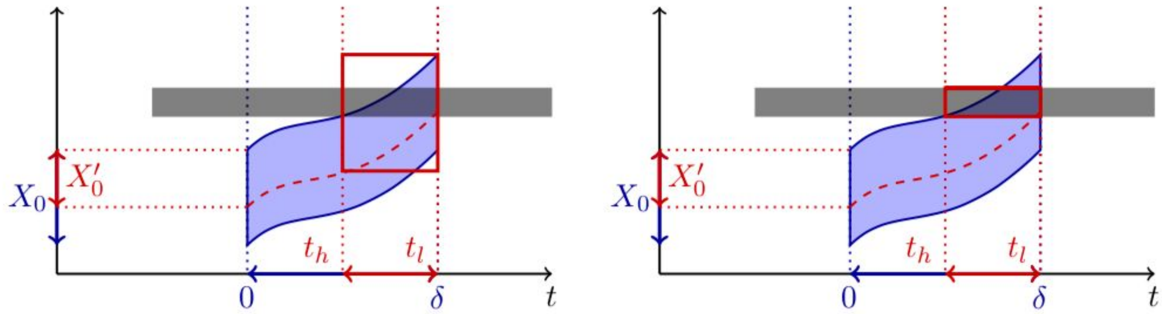
1>Lower bound approximation search.(week 4)

2>Upper bound approximation search.(week 6)

7.July 7 - July 22 (Weeks 7 & 8)

I these weeks, I shall implement the over-approximate the range of TM flowpipe.

1>Zonotopes over approximation.



(a) Domain contraction along with range over-approximation

(b) Intersection of the range over-approximation and the guard

8.July 22 - July 28 (Week 9)

Catch up with any unfinished work left in the past weeks. Preparation for **Phase 2** Evaluation Submission Deadline which is on July 26.

9.July 29 - August 4 (Weeks 10)

In this weeks, I shall write algorithm Reachable set computation for hybrid Automata by better implementation of reach set computation and Intersection of TMs with guard set.

10.August 5 - August 10 (Week 11)

During this time span, I shall construct flowpipe for Hybrid automata which over approximates reach set within bounded time zone and maximum jump depth.

11.August 11 - August 17 (Week 12)

Extensive and rigorous testing by cross-checking the obtained Benchmark results.

a>Space Rendezvous Benchmark

$$\dot{x} = v_x$$

$$\dot{y} = v_y$$

$$\dot{v}_x = n^2 x + 2n v_y + \frac{\mu}{r^2} - \frac{\mu}{r^3}(r+x) + \frac{u_x}{m_c}$$

$$\dot{v}_y = n^2 y - 2n v_x - \frac{\mu}{r_c^3} y + \frac{u_y}{m_c}$$

b> Spiking neurons

c>Nonlinear transmission line circuit

12.August 18 - August 26 (Week 13 and Wrap up)

Preparation for Final Phase Evaluation Submission Deadline which is on August 26. This includes code-formatting, completing documentation and solving unknown bugs. I shall write a Blog post recounting my experience and my work.

References

1>https://www.cs.colorado.edu/~xich8622/papers/thesis.pdf

2>http://www-bcf.usc.edu/~jdeshmuk/teaching/cs699-fm-for-cps/Papers/A2.pdf

3>https://bt.pa.msu.edu/pub/papers/VISNC09/VISNC09.pdf

4>https://easychair.org/publications/open/gjfh

Theoretical Background

(http://www.iiitt.ac.in/downloads/curriculum/CSE_Syllabus_16.pdf)

1> Mathematics I.

2> Mathematics II (ODEs,Taylor approximations).

3>Introduction to Probability Theory.

4>Principles of Operations Research.

5>Algorithms.

Mathematics-||

Course Content

Vector space – Subspaces – Linear dependence and independence – Spanning of a subspace– Basis and Dimension.Inner product – Inner product spaces – Orthogonal and orthonormal basis – Gram- Schmidt orthogonalization process.

Basic review of first order differential equation - Higher order linear differential equations with constant coefficients –Particular integrals for $x^n e^{ax}$, e^{ax} cos (bx), e^{ax} sin (bx) –

Equation reducible to linear equations with constant coefficients using $x\ e^t$ - Simultaneous linear equations with constant coefficients – Method of variation of parameters – Applications – Electric circuit problems.

Gradient, Divergence and Curl – Directional Derivative – Tangent Plane and normal to surfaces – Angle between surfaces –Solenoidal and irrotational fields – Line, surface and volume integrals – Green's Theorem, Stokes' Theorem and Gauss Divergence Theorem (all without proof) – Verification and applications of these theorems.

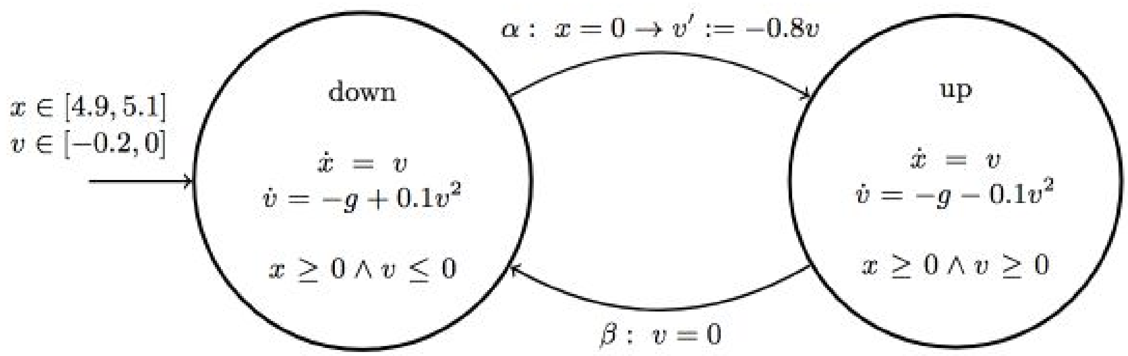
Analytic functions – Cauchy – Riemann equations (Cartesian and polar) –Properties of analytic functions – Construction of analytic functions given real or imaginary part – Conformal mapping of standard elementary functions (z^2,e^z , sin z, cos z, $z+k^2/z$) and bilinear transformation.

Cauchy's integral theorem, Cauchy's integral formula and for derivatives– Taylor's and Laurent's expansions (without proof) – Singularities – Residues – Cauchy's residue theorem – Contour integration involving unit circle.

Why Me?

Basically, I use C and python for my classwork, In November of last year I found out about Julia. “Fast like C and Easy like python”, So started using Julia. Due to this exposure to Julia and coursework on Mathematics II(ODEs, Taylor approximation), I am quite familiar with implementing in Reachability using Taylor Models. This is demonstrated by my pull requests #114, in which I implemented

Example 4.1.2 on page 98 in
https://www.cs.colorado.edu/~xich8622/papers/thesis.pdf.



Availability and Working Hours

I have no other commitments during the summers.I can put 40+ hours a week.

To speed up the project I am ready to do even more.

Algorithm 13 Branch-and-prune algorithm

Input: a CSP $\langle X, D, Cons\rangle$, a threshold $\epsilon > 0$
Output: an over-approximation of the solution set

$S \leftarrow \emptyset$;
$Queue.enqueue(D)$; # queue for the solution set
while $Queue$ is not empty **do**
  $B \leftarrow Queue.dequeue()$;
  if B possibly contains a solution **then**
    if the width of B in the i-th dimension for some i is no smaller than ϵ **then**
      Split B equally in the i-th dimension into B_1, B_2;
      $Queue.enqueue(B_1)$;
      $Queue.enqueue(B_2)$;
    else
      $S \leftarrow S \cup \{B\}$; # $\mathrm{W}(B) < \epsilon$
    end if
  end if
end while
return S;

Algorithm 15 Lower bound approximation search

Input: a CSP $\langle\{\vec{x}_0[1], \ldots, \vec{x}_0[d], t\}, D, Cons\rangle$, a threshold $\epsilon > 0$, a variable $z \in \{\vec{x}_0[1], \ldots, \vec{x}_0[d], t\}$
Output: a conservative approximation for the lower bound of the solution set in the dimension of z

Set α as the lower bound of D in the dimension of z;
Set β as the upper bound of D in the dimension of z;
$D' \leftarrow D$;
while $\beta - \alpha \geq \epsilon$ **do**
  $\gamma \leftarrow \frac{\alpha+\beta}{2}$;
  Update the interval of D' in the dimension of z to $[\alpha, \gamma]$;
  if D' possibly contains a solution **then**
    $\beta \leftarrow \gamma$;
  else
    $\alpha \leftarrow \gamma$;
  end if
end while
return α;

Algorithm 14 Main procedure of the efficient approach

Input: a CSP $\langle\{\vec{x}_0[1], \ldots, \vec{x}_0[d], t\}, D, \mathit{Cons}\rangle$

Output: an interval enclosure of the solution set

$D_c \leftarrow D$;
repeat
    for all $z = \vec{x}_0[1], \ldots, \vec{x}_0[d], t$ **do**
        Compute a value `lo` such that $\mathtt{lo} \leq \nu(z)$ for any solution ν w.r.t. D_c.
        Compute a value `up` such that $\mathtt{up} \geq \nu(z)$ for any solution ν w.r.t. D_c.
        if $\mathtt{lo} \leq \mathtt{up}$ **then**
            Update the interval of D_c in the dimension of z to $[\mathtt{lo}, \mathtt{up}]$;
        else
            Set D_c empty and break; # the constraints are unsatisfiable
        end if
    end for # refine the domain D_c
until no great refinement on D_c
return D_c;