A Rigorous Implementation of Taylor Model-Based Integrator in Chebyshev Basis Representation [1]

Tomáš Dzetkulič

Institute of Computer Science
Czech Academy of Sciences

December 15, 2011

[1] This work was supported by Czech Science Foundation grant 201/09/H057, MŠMT project number OC10048 and institutional research plan AV0Z100300504.

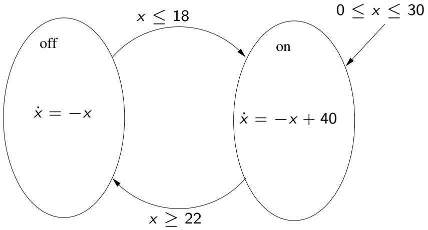
Motivation

Hybrid System: A dynamic system with both discrete and continuous behavior

Motivation

Hybrid System: A dynamic system with both discrete and continuous behavior

Example: Thermostat



Safety Verification of Hybrid Systems

Continuous variables evolve according to differential equations

Safety Verification of Hybrid Systems

Continuous variables evolve according to differential equations

Computing of rigorous enclosures of continuous behavior required

Safety Verification of Hybrid Systems

Continuous variables evolve according to differential equations

Computing of rigorous enclosures of continuous behavior required

Initial condition in a form of Taylor model

Safety Verification of Hybrid Systems

Continuous variables evolve according to differential equations

Computing of rigorous enclosures of continuous behavior required

Initial condition in a form of Taylor model

Implementation in tool HSOLVER [Ratschan and She, 2007]
`http://hsolver.sourceforge.net`
Uses degree one Taylor expansions of continuous evolution

Evolution Enclusure in Hybrid Systems Verification

We want to use better than degree one Taylor expansions

Evolution Enclusure in Hybrid Systems Verification

We want to use better than degree one Taylor expansions

$\Rightarrow$ Use of rigorous Taylor model-based integrator

Evolution Enclusure in Hybrid Systems Verification

We want to use better than degree one Taylor expansions

$\Rightarrow$ Use of rigorous Taylor model-based integrator

We want to handle highly non-linear input models

Evolution Enclusure in Hybrid Systems Verification

We want to use better than degree one Taylor expansions

⇒ Use of rigorous Taylor model-based integrator

We want to handle highly non-linear input models

⇒ We propose new method for long-term wrapping effect suppresion

Evolution Enclusure in Hybrid Systems Verification

We want to use better than degree one Taylor expansions

⇒ Use of rigorous Taylor model-based integrator

We want to handle highly non-linear input models

⇒ We propose new method for long-term wrapping effect suppresion

We want to handle high dimensional examples with small models

Evolution Enclusure in Hybrid Systems Verification

We want to use better than degree one Taylor expansions

$\Rightarrow$ Use of rigorous Taylor model-based integrator

We want to handle highly non-linear input models

$\Rightarrow$ We propose new method for long-term wrapping effect suppresion

We want to handle high dimensional examples with small models

$\Rightarrow$ Use of close to optimal Chebyshev basis approximations

Verified Taylor Model-Based Integrator

Given:

- ODE: $\dot{x} = f(x) + I_1$
- Initial condition $g(a) + I_2$
- Time bound t_{max}

Verified Taylor Model-Based Integrator

Given:

- ODE: $\dot{x} = f(x) + I_1$
- Initial condition $g(a) + I_2$
- Time bound t_{max}

Compute evolution $e(t, a) + I_3$ such that:

- $e(t, a) + I_3$ is the solution to the given ODE and
- $g(a) + I_2 \in e(0, a) + I_3$

Verified Taylor Model-Based Integrator

Given:

- ODE: $\dot{x} = f(x) + I_1$
- Initial condition $g(a) + I_2$
- Time bound t_{max}

Compute evolution $e(t, a) + I_3$ such that:

- $e(t, a) + I_3$ is the solution to the given ODE and
- $g(a) + I_2 \in e(0, a) + I_3$

Problem: In case I_2 is non-empty:

- I_3 is always wider than I_2
- wrapping effect applies over multiple integration steps

Verified Taylor Model-Based Integrator

Given:

- ODE: $\dot{x} = f(x) + I_1$
- Initial condition $g(a) + I_2$
- Time bound t_{max}

Compute evolution $e(t, a) + I_3$ such that:

- $e(t, a) + I_3$ is the solution to the given ODE and
- $g(a) + I_2 \in e(0, a) + I_3$

Problem: In case I_2 is non-empty:

- I_3 is always wider than I_2
- wrapping effect applies over multiple integration steps

Solution: Make interval I_2 empty or very small

Long-Term Stabilization of Integrator

Known methods:

- Shrink wrapping - Include error interval into the polynomial part of Taylor model
- Preconditioning - Use composition of two Taylor models where the outer Taylor model is error free

Long-Term Stabilization of Integrator

Known methods:

- Shrink wrapping - Include error interval into the polynomial part of Taylor model
- Preconditioning - Use composition of two Taylor models where the outer Taylor model is error free

Methods use linear part of initial condition $g(a)$ to:

- Either absorb the error
- Or construct the composition of models

Long-Term Stabilization of Integrator

Known methods:

- Shrink wrapping - Include error interval into the polynomial part of Taylor model
- Preconditioning - Use composition of two Taylor models where the outer Taylor model is error free

Methods use linear part of initial condition $g(a)$ to:

- Either absorb the error
- Or construct the composition of models

Problem: Non-linear initial condition

New Long-Term Stabilization Method

We propose a new method based on affine arithmetic

In each integration step:

- Use affine combination of new variables b to create Taylor model $g'(b)$ of interval I_2
- Use $g(a) + g'(b)$ as a new initial condition
- Compute evolution enclosure $e(t, a, b) + I_3$

New Long-Term Stabilization Method

We propose a new method based on affine arithmetic

In each integration step:

- Use affine combination of new variables b to create Taylor model $g'(b)$ of interval I_2
- Use $g(a) + g'(b)$ as a new initial condition
- Compute evolution enclosure $e(t, a, b) + I_3$

Observation:

- Taylor model $g(a) + g'(b)$ is error free

New Long-Term Stabilization Method

We propose a new method based on affine arithmetic

In each integration step:

- Use affine combination of new variables b to create Taylor model $g'(b)$ of interval I_2
- Use $g(a) + g'(b)$ as a new initial condition
- Compute evolution enclosure $e(t, a, b) + I_3$

Observation:

- Taylor model $g(a) + g'(b)$ is error free
- Coefficients in terms containing b are small in magnitude

New Long-Term Stabilization Method

We propose a new method based on affine arithmetic

In each integration step:

- Use affine combination of new variables b to create Taylor model $g'(b)$ of interval I_2
- Use $g(a) + g'(b)$ as a new initial condition
- Compute evolution enclosure $e(t, a, b) + I_3$

Observation:

- Taylor model $g(a) + g'(b)$ is error free
- Coefficients in terms containing b are small in magnitude
- Method independent on the structure of $g(a)$

New Long-Term Stabilization Method

We propose a new method based on affine arithmetic

In each integration step:

- Use affine combination of new variables b to create Taylor model $g'(b)$ of interval I_2
- Use $g(a) + g'(b)$ as a new initial condition
- Compute evolution enclosure $e(t, a, b) + I_3$

Observation:

- Taylor model $g(a) + g'(b)$ is error free
- Coefficients in terms containing b are small in magnitude
- Method independent on the structure of $g(a)$

Problem: In each step we add new set of additional variables b

Multi-Step Integration

In $i - th$ integration step:

- Split initial condition $g_i(a, b_{i-1}) + I_2$ into $g_i^*(a)$ and $g_i^+(a, b_{i-1}) + I_2$

Multi-Step Integration

In $i - th$ integration step:

- Split initial condition $g_i(a, b_{i-1}) + I_2$ into $g_i^*(a)$ and $g_i^+(a, b_{i-1}) + I_2$
- Use new variables b_i to create a new Taylor model $g_i'(b_i)$ such that $g_i^+(a, b_{i-1}) + I_2 \in g_i'(b_i)$

Multi-Step Integration

In $i - th$ integration step:

- Split initial condition $g_i(a, b_{i-1}) + I_2$ into $g_i^*(a)$ and $g_i^+(a, b_{i-1}) + I_2$
- Use new variables b_i to create a new Taylor model $g_i'(b_i)$ such that $g_i^+(a, b_{i-1}) + I_2 \in g_i'(b_i)$
- Use initial condition $g_i^*(a) + g_i'(b_i)$ to compute evolution enclosure $e_i(t, a, b_i) + I_3$

Multi-Step Integration

In $i - th$ integration step:

- Split initial condition $g_i(a, b_{i-1}) + I_2$ into $g_i^*(a)$ and $g_i^+(a, b_{i-1}) + I_2$
- Use new variables b_i to create a new Taylor model $g_i'(b_i)$ such that $g_i^+(a, b_{i-1}) + I_2 \in g_i'(b_i)$
- Use initial condition $g_i^*(a) + g_i'(b_i)$ to compute evolution enclosure $e_i(t, a, b_i) + I_3$
- Use $e_i(t_{max}, a, b_i) + I_3$ as the initial condition in $(i+1)$-th step

Multi-Step Integration

In $i - th$ integration step:

- Split initial condition $g_i(a, b_{i-1}) + I_2$ into $g_i^*(a)$ and $g_i^+(a, b_{i-1}) + I_2$
- Use new variables b_i to create a new Taylor model $g_i'(b_i)$ such that $g_i^+(a, b_{i-1}) + I_2 \in g_i'(b_i)$
- Use initial condition $g_i^*(a) + g_i'(b_i)$ to compute evolution enclosure $e_i(t, a, b_i) + I_3$
- Use $e_i(t_{max}, a, b_i) + I_3$ as the initial condition in $(i + 1)$-th step

In every step variables b_{i-1} are replaced with variables b_i

Only one set of additional variables is present in any step

Multi-Step Integration

In $i - th$ integration step:

- Split initial condition $g_i(a, b_{i-1}) + I_2$ into $g_i^*(a)$ and $g_i^+(a, b_{i-1}) + I_2$
- Use new variables b_i to create a new Taylor model $g_i'(b_i)$ such that $g_i^+(a, b_{i-1}) + I_2 \in g_i'(b_i)$
- Use initial condition $g_i^*(a) + g_i'(b_i)$ to compute evolution enclosure $e_i(t, a, b_i) + I_3$
- Use $e_i(t_{max}, a, b_i) + I_3$ as the initial condition in $(i + 1)$-th step

In every step variables b_{i-1} are replaced with variables b_i

Only one set of additional variables is present in any step

We can estimate the error through back-substitution, since the dependency between b_{i-1} and b_i is known

New Method Properties

Method independent of the structure of initial condition

New Method Properties

Method independent of the structure of initial condition

Method able to suppress the error wrapping effect

New Method Properties

Method independent of the structure of initial condition

Method able to suppress the error wrapping effect

Additional variables that represent the unknown error required

Chebyshev Polynomials

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

Chebyshev Polynomials

$T_0(x) = 1$
$T_1(x) = x$
$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$

From approximation theory:
Expanding function in Chebyshev polynomials:

$$f(x) = \sum_{i=0}^{\infty} a_i T_i(x)$$

Cutting off the series after the T_N term is close to optimal approximation of $f(x)$

Chebyshev Polynomials

$T_0(x) = 1$
$T_1(x) = x$
$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$

From approximation theory:
Expanding function in Chebyshev polynomials:

$$f(x) = \sum_{i=0}^{\infty} a_i T_i(x)$$

Cutting off the series after the T_N term is close to optimal approximation of $f(x)$

Approximation is orders of accuracy more accurate than expansions in Taylor series [Kaucher and Miranker, 1988]

Chebyshev Polynomial Operations

Multiplication: $T_i(x)T_j(x) = (T_{i+j}(x) + T_{|i-j|}(x))/2$

Chebyshev Polynomial Operations

Multiplication: $T_i(x)\,T_j(x) = (T_{i+j}(x) + T_{|i-j|}(x))/2$

Substitution: Clenshaw algorithm [Clenshaw, 1955]

Chebyshev Polynomial Operations

Multiplication: $T_i(x)T_j(x) = (T_{i+j}(x) + T_{|i-j|}(x))/2$

Substitution: Clenshaw algorithm [Clenshaw, 1955]

Integration:

- $\int_0^y T_0(x)dx = T_1(y)$
- $\int_0^y T_1(x)dx = (T_0(y) + T_2(y))/4$
- for even $i > 1$:
  $\int_0^y T_i(x)dx = (-T_{i-1}(y)/(i-1) + T_{i+1}(y)/(i+1))/2$
- analogous for odd $i > 1$

Chebyshev Polynomial Operations

Multiplication: $T_i(x)T_j(x) = (T_{i+j}(x) + T_{|i-j|}(x))/2$

Substitution: Clenshaw algorithm [Clenshaw, 1955]

Integration:

- $\int_0^y T_0(x)dx = T_1(y)$
- $\int_0^y T_1(x)dx = (T_0(y) + T_2(y))/4$
- for even $i > 1$:
  $\int_0^y T_i(x)dx = (-T_{i-1}(y)/(i-1) + T_{i+1}(y)/(i+1))/2$
- analogous for odd $i > 1$

Possibility to extend all operations with rigorous error estimation

Computational Experiments

We have an implementation of the verified integrator using our wrapping effect suppression method

Computational Experiments

We have an implementation of the verified integrator using our wrapping effect suppression method

Tested example:
$\dot{x} = 2x(1-y) \quad \dot{y} = y(x-1)$
$x_0 \in [0.95, 1.05] \quad y_0 \in [2.95, 3.05]$
$t_{max} = 5.488138468035$

Computational Experiments

We have an implementation of the verified integrator using our wrapping effect suppression method

Tested example:
$\dot{x} = 2x(1-y) \quad \dot{y} = y(x-1)$
$x_0 \in [0.95, 1.05] \quad y_0 \in [2.95, 3.05]$
$t_{max} = 5.488138468035$

COSY computes resulting Taylor model in time t_{max} with error interval of width 3×10^{-9} (degree 12 Taylor model)
[Makino and Berz, 2006]

Computational Experiments

Table of error interval width:

	Our Implementation		COSY
Order	Taylor series	Chebyshev poly.	
4	4.4E-2	7.4E-3	
6	1.1E-3	6.6E-5	
8	3.4E-5	6.2E-7	
10	1.1E-6	5.7E-9	
12	3.4E-8	5.2E-11	3E-9
14	1.1E-9	9.8E-13	

Computation of our tool used fixed time step

Conclusion

New method for long-term stabilization of the Taylor model-based verified integrator

Conclusion

New method for long-term stabilization of the Taylor model-based verified integrator

Based on adding additional variables that hold the error

Conclusion

New method for long-term stabilization of the Taylor model-based verified integrator

Based on adding additional variables that hold the error

Implementation with both:

- Taylor series
- Chebyshev approximations

Conclusion

New method for long-term stabilization of the Taylor model-based verified integrator

Based on adding additional variables that hold the error

Implementation with both:

- Taylor series
- Chebyshev approximations

Computational experiments demonstrate the usefulness of the method

Ongoing, Future Work

- Automatic selection of the time step length

Ongoing, Future Work

- Automatic selection of the time step length
- Automatic selection of the representation degree given the required precision

Ongoing, Future Work

- Automatic selection of the time step length
- Automatic selection of the representation degree given the required precision
- Using the method in the hybrid system safety verification

Ongoing, Future Work

- Automatic selection of the time step length
- Automatic selection of the representation degree given the required precision
- Using the method in the hybrid system safety verification

I want to thank Stefan Ratschan for his comments that helped me to prepare this presentation