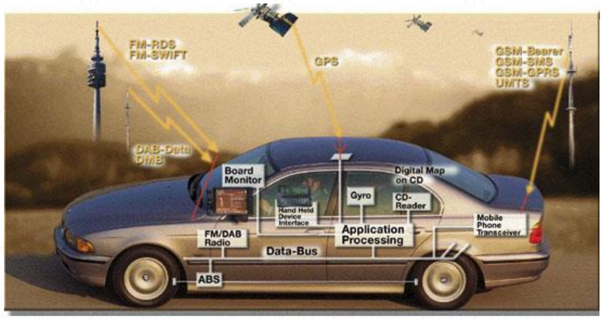
Verification of Annotated Models from Executions

Parasara Sridhar Duggirala,
Sayan Mitra,
Mahesh Viswanathan

ILLINOIS

Motivation

- Embedded Systems – interact with physical environment, controlled by computer
- Deployed in many safety critical applications



- Continuous dynamics involves nonlinear ODEs and several modes of operation
- Requires that the system is always **safe**

Motivation

- **Testing**: Most common technique for *checking* functional properties of embedded systems.
- Problem: Testing can only take us so far!
- Can we obtain *formal guarantees* from sample executions?

Motivation

- **Testing**: Most common technique for *checking* functional properties of embedded systems.
- Problem: Testing can only take us so far!
- Can we obtain *formal guarantees* from sample executions?

- Dealing with continuous executions?
- Can we use additional information from the system designer?

- Annotations for embedded systems – spirit of code contracts and loop invariants

Contributions

- ✓ Propose a notion of annotations called as *discrepancy function*
- ✓ Show how discrepancy function subsumes other proof theoretic notions used in control theory
- ✓ Given a model of switching system and annotations, give a **sound** and **relatively complete** algorithm for safety verification.

Contributions

- ✓ Propose a notion of annotations called as *discrepancy function*
- ✓ Show how discrepancy function subsumes other proof theoretic notions used in control theory
- ✓ Given a model of switching system and annotations, give a **sound** and **relatively complete** algorithm for safety verification.

Annotated Models + Sample Executions = Scalable, Sound and Relative Complete Verification Technique

Outline

- ✓ Motivation & Contributions
- Discrepancy function as annotation and its relation to other notions
- ε – error bound execution
- Sound and relative complete verification algorithms
- Experimental results
- Conclusions and future work

Related work

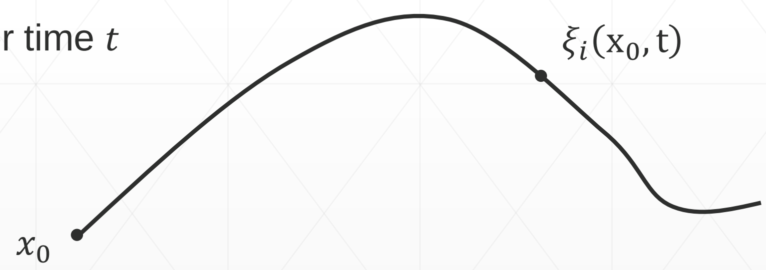
- Verification using Simulations [Girard et. al. 06]
- Sensitivity Analysis and Systematic Simulations – Breach [Donze et.al. 06,09]
- Symbolic analysis of Simulink/Stateflow models [Kanade et.al. 09]
- Monte-Carlo falsification techniques [Nghiem et.al. 10]
- Statistical Model Checking [Clarke et.al. 11]
- Bounded Reach Sets [Huang et.al. 11]

Annotations

- *Annotations in software*
- Annotations for continuous variables
- Continuous behavior $\dot{x} = f_i(x, t)$, $x \in \mathbb{R}^n, t \in \mathbb{R}^{\geq 0}, I, \{f_i\}_{i \in I}$, $\Theta \subseteq \mathbb{R}^n$

Annotations

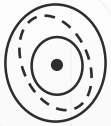
- *Annotations in software*
- Annotations for continuous variables
- Continuous behavior $\dot{x} = f_i(x, t)$, $x \in \mathbb{R}^n, t \in \mathbb{R}^{\geq 0}, I, \{ f_i\}_{i \in I}$, $\Theta \subseteq \mathbb{R}^n$
- Solution or trajectory for each mode i
  - ξ_i: $\mathbb{R}^n \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^n$
  - $\xi_i(\mathrm{x}_0, \mathrm{t})$: state of the system from $x_0 \in \Theta$ after time t



- Annotation would involve states and trajectories

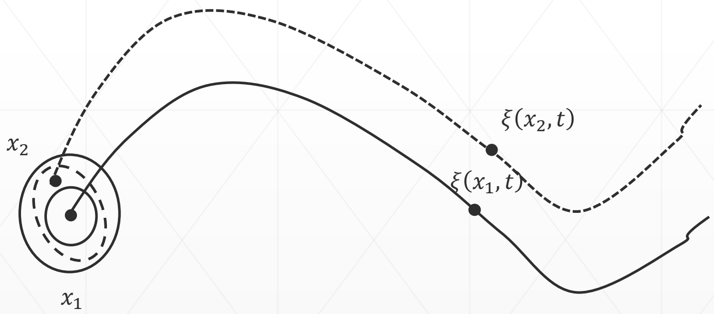
Annotations: Discrepancy function

- Definition. A smooth function $V : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{\geq 0}$ is a *discrepancy function* for $\dot{x} = f(x,t)$ if for any x_1 and $x_2 \in \mathbb{R}^n$
    1. (static bound) $\exists\, \alpha_1, \alpha_2$: $\alpha_1(|x_1 - x_2|) \leq V(x_1, x_2) \leq \alpha_2(|x_1 - x_2|)$
    2. (dynamic bound) $V\big(\xi(x_1,t), \xi(x_2,t)\big) \leq \beta(x_1, x_2, t)$ where $\beta: \mathbb{R}^{2n} \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ and $\beta \rightarrow 0$ as $x_1 \rightarrow x_2$



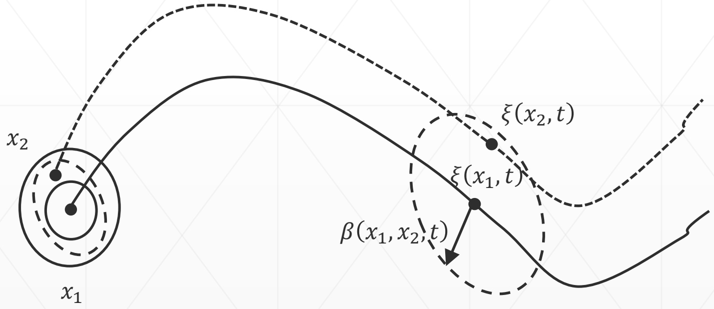
Annotations: Discrepancy function

- Definition. A smooth function $V : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{\geq 0}$ is a *discrepancy function* for $\dot{x} = f(x,t)$ if for any x_1 and $x_2 \in \mathbb{R}^n$
  1. (static bound) $\exists\, \alpha_1, \alpha_2\colon\ \alpha_1(|x_1 - x_2|) \leq V(x_1, x_2) \leq \alpha_2(|x_1 - x_2|)$
  2. (dynamic bound) $V\big(\xi(x_1,t), \xi(x_2,t)\big) \leq \beta(x_1, x_2, t)$ where $\beta\colon \mathbb{R}^{2n} \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ and $\beta \rightarrow 0$ as $x_1 \rightarrow x_2$



Annotations: Discrepancy function

- Definition. A smooth function $V : \mathbb{R}^{2n} \to \mathbb{R}^{\geq 0}$ is a *discrepancy function* for $\dot{x} = f(x,t)$ if for any x_1 and $x_2 \in \mathbb{R}^n$
  1. (static bound) $\exists\ \alpha_1, \alpha_2$: $\alpha_1(|x_1 - x_2|) \leq V(x_1, x_2) \leq \alpha_2(|x_1 - x_2|)$
  2. (dynamic bound) $V(\xi(x_1,t), \xi(x_2,t)) \leq \beta(x_1, x_2, t)$ where $\beta: \mathbb{R}^{2n} \times \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ and $\beta \to 0$ as $x_1 \to x_2$

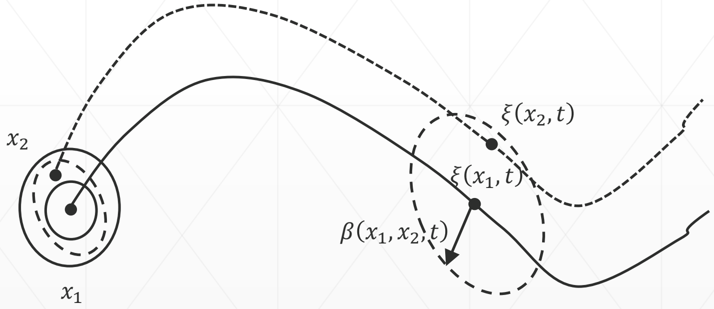


Annotations: Discrepancy function

- Definition. A smooth function $V : \mathbb{R}^{2n} \to \mathbb{R}^{\geq 0}$ is a *discrepancy function* for $\dot{x} = f(x,t)$ if for any x_1 and $x_2 \in \mathbb{R}^n$
  1. (static bound) $\exists\ \alpha_1, \alpha_2$: $\alpha_1(|x_1 - x_2|) \leq V(x_1, x_2) \leq \alpha_2(|x_1 - x_2|)$
  2. (dynamic bound) $V(\xi(x_1,t), \xi(x_2,t)) \leq \beta(x_1, x_2, t)$ where $\beta: \mathbb{R}^{2n} \times \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ and $\beta \to 0$ as $x_1 \to x_2$

- $(\alpha_1, \alpha_2, \beta)$ is a witness for V

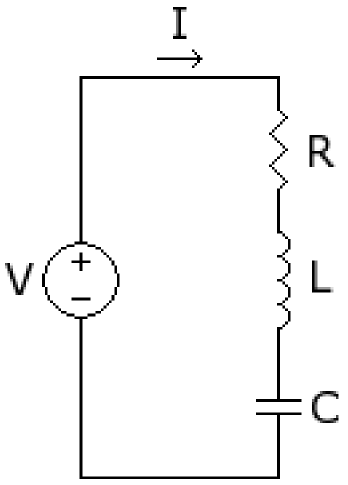
- Stability not required



RLC Circuit as example

- RLC Circuit: $\frac{d^2 i}{dt^2} + \frac{R}{L}\frac{di}{dt} + \frac{i}{LC} = 0$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, say\ A \coloneqq \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix}$$



Example RLC Circuit

- Initially : $3 \le i \le 5$ & $\frac{di}{dt} = 0$, which is $3 \le u \le 5$ & $v = 0$
- Property : after 1 time unit, current should be less or equal to 3 units
  $t > 1 \Rightarrow i \le 3$, unsafe set is $\mathrm{U} \triangleq t > 1$ & $i > 3$
- Look at different possible annotations for this system and then verify the property

Lipschitz dynamics

- Definition. System $\dot{x} = f(x, t)$ is said to be Lipschitz continuous if

$$\exists\, L \in \mathbb{R}^{\geq 0}, \forall x_1, x_2 \in \mathbb{R}^n, f(x_1, t) - f(x_2, t) \leq L\,|x_1 - x_2|$$

- Proposition. If L is the Lipschitz constant for the function $f(x, t)$ then $V(x_1, x_2) = |x_1 - x_2|$ is a discrepancy function with $\beta \coloneqq e^{Lt}|x_1 - x_2|$.

- Worst case estimate : Exponential divergence.

- For the Example RLC Circuit, Lipchitz constant $L = |A| \approx 3$, $|x_1 - x_2|$ is a discrepancy function with $\beta = e^{Lt}|x_1 - x_2|$

Incremental Stability

- Definition. The system is incrementally stable if there is a KL function γ such that for any two initial states $x_1$and x_2 $|\xi(x_1,t) - \xi(x_2,t)| \leq \gamma(|x_1 - x_2|,t)$.

- Theorem. [Angeli 2000]. If the system is incrementally stable then there exists a smooth function (incremental Lyapunov function) $\mathrm{V}: \mathbb{R}^{2n} \to \mathbb{R}^{\geq 0}$ and $\alpha: \mathbb{R} \to \mathbb{R}^{\geq 0}$ s.t.

$$\mathrm{V}\big(\xi(x_1,t), \xi(x_2,t)\big) - V(x_1,x_2) \leq \int_0^t -\alpha(|\xi(x_1,\tau) - \xi(x_2,\tau)|)d\tau.$$

- Proposition. Incremental Lyapunov function is a discrepancy function with $\beta(x_1,x_2,t) = V(x_1,x_2) + \int_0^t -\alpha(|\xi(x_1,\tau) - \xi(x_2,\tau)|)d\tau$.

- For the Example RLC circuit, with $\mathrm{P} = \begin{bmatrix} 2.5 & .5 \\ .5 & .75 \end{bmatrix}, V = (x_1 - x_2)^T P(x_1 - x_2)$ is a discrepancy function with $\beta(x_1,x_2,t) = V(x_1,x_2) + \int_0^t -\alpha(|\xi(x_1,\tau) - \xi(x_2,\tau)|)d\tau$ where $\alpha = (x_1 - x_2)^T(x_1 - x_2)$

About Annotations

- Comparing different annotations:
  - Lipschitz Constant : Exponential divergence
  - Contraction Metric : Exponential Convergence
  - Incremental Stability : Convergence
  - Extension of Incremental Stability called Incremental Forward Completeness

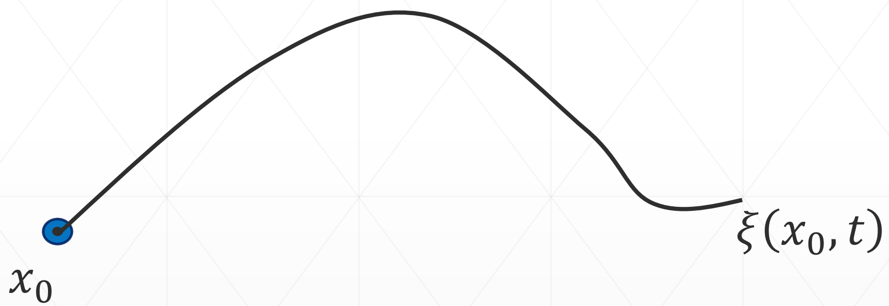
- Discrepancy function does not require convergence

About Annotations

- How are annotations useful : computing sound over approximations

$$\forall\, x \in B_{\delta}(x_0), \xi(x,T) \in B_{\varepsilon}^{V}\big(\xi(x_0,T)\big)\ where\ \varepsilon = \sup_{x\, \in B_{\delta}(x_0), 0\le t\le T} \{\beta(x,x_0,t)\}$$

$$B_{\varepsilon}^{V}(x) = \{\, x' \,|\, V(x,x') \le \varepsilon\}$$

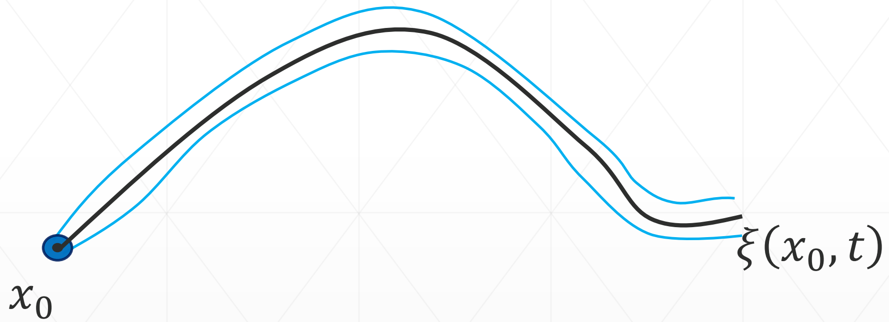


About Annotations

- How are annotations useful : computing sound over approximations

$\forall\, x \in B_\delta(x_0), \xi(x,T) \in\; B_\varepsilon^V\big(\xi(x_0,T)\big)\; where\ \varepsilon = \sup_{x\, \in B_\delta(x_0), 0\le t\le T} \{\beta(x, x_0, t)\}$

$B_\varepsilon^V(x) = \{\, x' \,|\, V(x,x') \le\; \varepsilon\}$

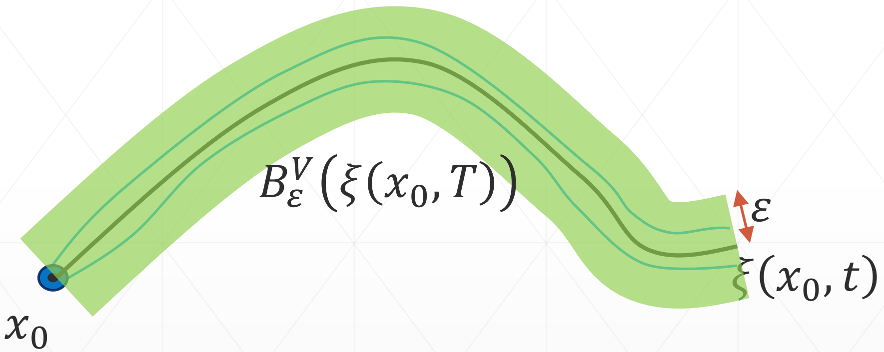


About Annotations

- How are annotations useful : computing sound over approximations

$$\forall\, x \in B_\delta(x_0), \xi(x,T) \in\ B_\varepsilon^V\big(\xi(x_0,T)\big)\ where\ \varepsilon = \sup_{x\, \in B_\delta(x_0), 0\le t\le T} \{\beta(x,x_0,t)\}$$

$$B_\varepsilon^V(x) = \{\, x' |\, V(x,x') \le\ \varepsilon\}$$

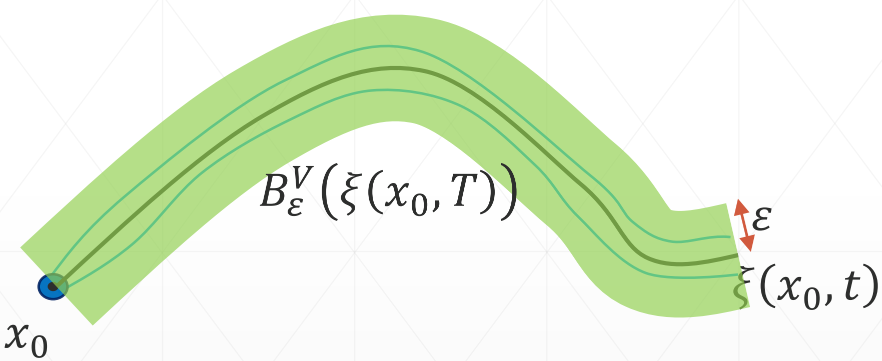


About Annotations

- How are annotations useful : computing sound over approximations

$$\forall\, x \in B_\delta(x_0), \xi(x,T) \in B^V_\varepsilon\big(\xi(x_0,T)\big)\ where\ \varepsilon = \sup_{x\, \in B_\delta(x_0), 0 \le t \le T} \{\beta(x, x_0, t)\}$$

$$B^V_\varepsilon(x) = \{\, x' \,|\, V(x, x') \le\ \varepsilon\}$$



- How to store the trajectory?

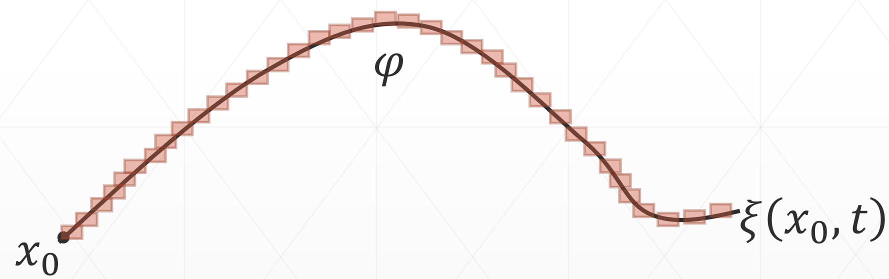
Execution Trace

- Analytical solution for ODE, $\xi(x_0, t)$ need not exist, rely on numerical methods
- Validated ODE solver (VNODE-LP)

Execution Trace

- Analytical solution for ODE, $\xi(x_0, t)$ need not exist, rely on numerical methods
- Validated ODE solver (VNODE-LP)

- Definition: $(x_0, T, \varepsilon, \tau) - simulation$ is a sequence $\varphi = (R_0, t_0), (R_1, t_1), \cdots, (R_k, t_k)$ s.t.
  1. $t_i - t_{i+1} \leq \tau$
  2. $\forall t \in [t_i \, , t_{i+1}], \xi(x_0, t) \ \in R_i$
  3. $diameter(R_i) \leq \varepsilon$



- Validated ODE solvers can indeed produce such enclosures using implicit and explicit methods for numerical integration

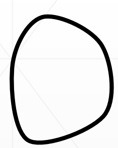
Basic Algorithm

- Partition, Simulate, Bloat, Check

Unsafe set

$$\dot{x} = f_i(x, t)$$

$$\xi_i \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^n$$



Initial Set

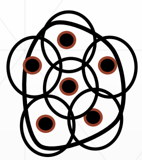
Basic Algorithm

- Partition, Simulate, Bloat, Check

Unsafe set

$$\dot{x} = f_i(x, t)$$

$$\xi_i \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^n$$



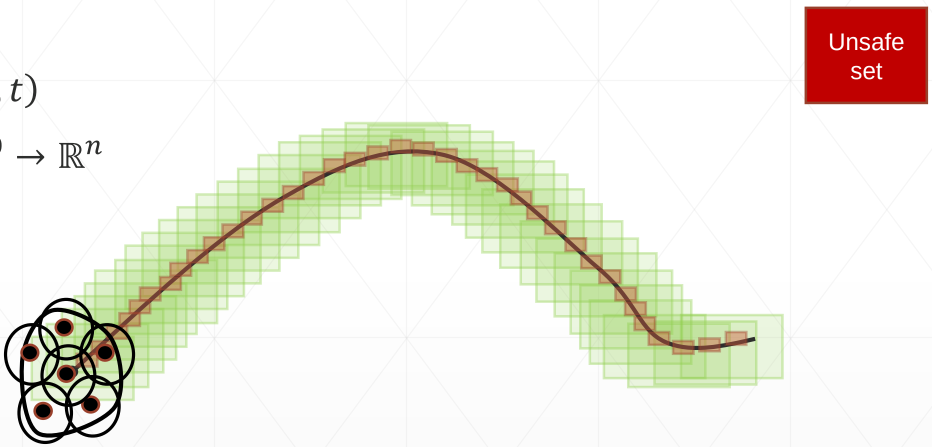
Initial Set

Basic Algorithm

- Partition, Simulate, Bloat, Check, Refine

$$\dot{x} = f_i(x, t)$$

$$\xi_i \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^n$$

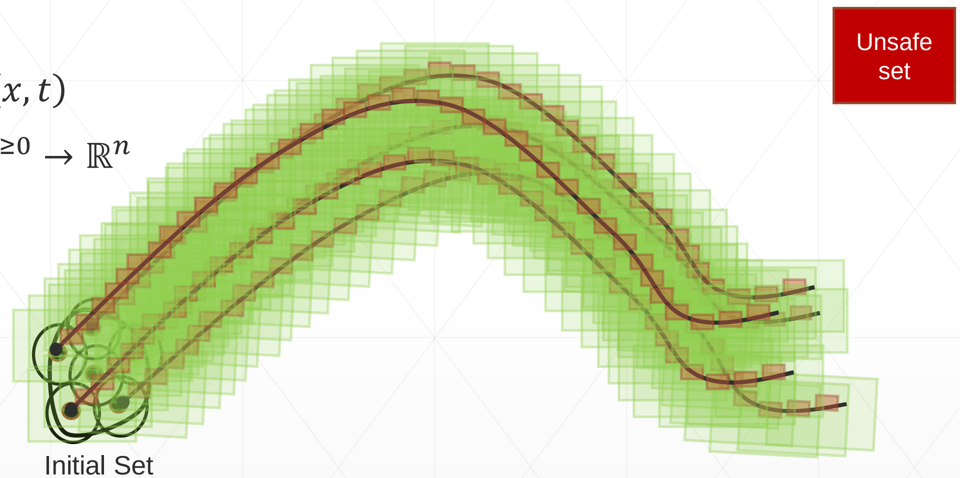


Basic Algorithm

- Partition, Simulate, Bloat, Check, Refine

$$\dot{x} = f_i(x, t)$$

$$\xi_i\colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R}^n$$

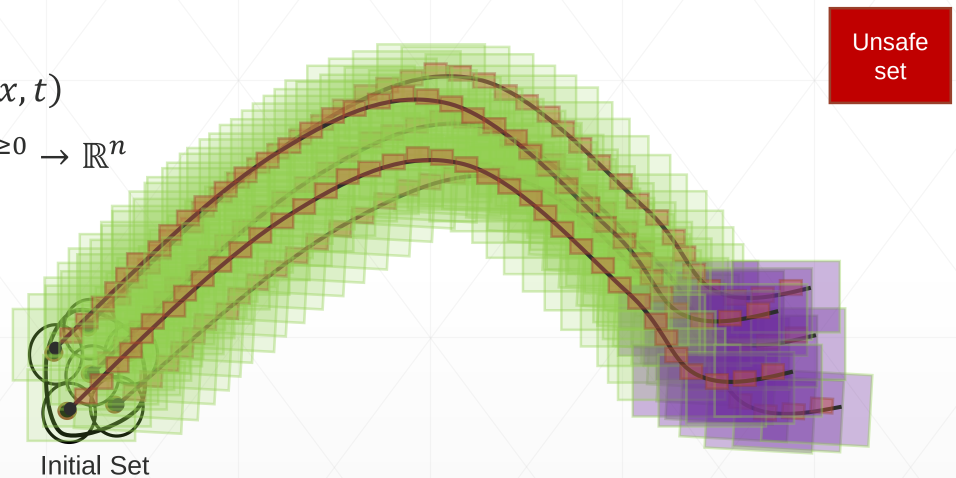


Basic Algorithm

- Partition, Simulate, Bloat, Check, Refine

$$\dot{x} = f_i(x, t)$$

$$\xi_i \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R}^n$$



Switch to new mode

$$\dot{x} = f_{i+1}(x, t)$$

$$\xi_{i+1} \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R}^n$$

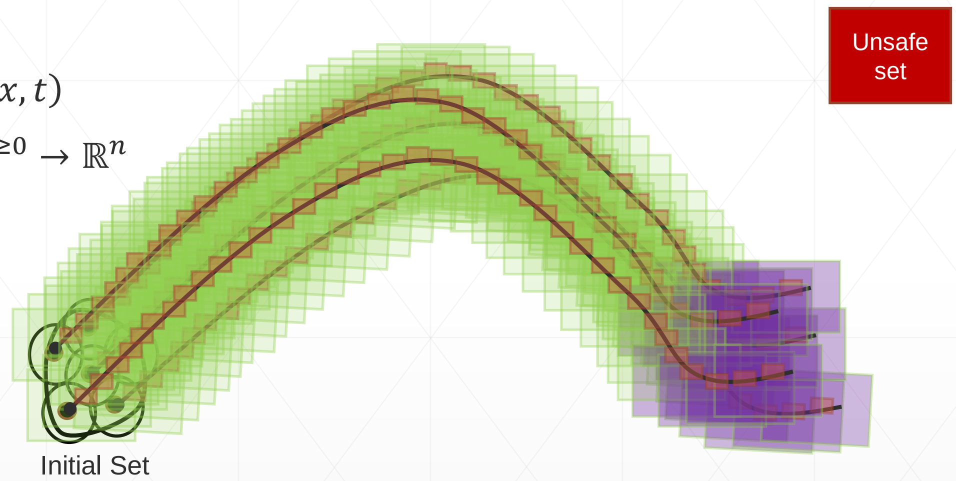
Switching time interval $[t_1, t_2]$

Basic Algorithm

- Partition, Simulate, Bloat, Check, Refine

$$\dot{x} = f_i(x, t)$$

$$\xi_i \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R}^n$$



Switch to new mode

$$\dot{x} = f_{i+1}(x, t)$$

$$\xi_{i+1} \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R}^n$$

Switching time interval $[t_1 , t_2]$

Unknown:
- Exact initial set for mode $i + 1$

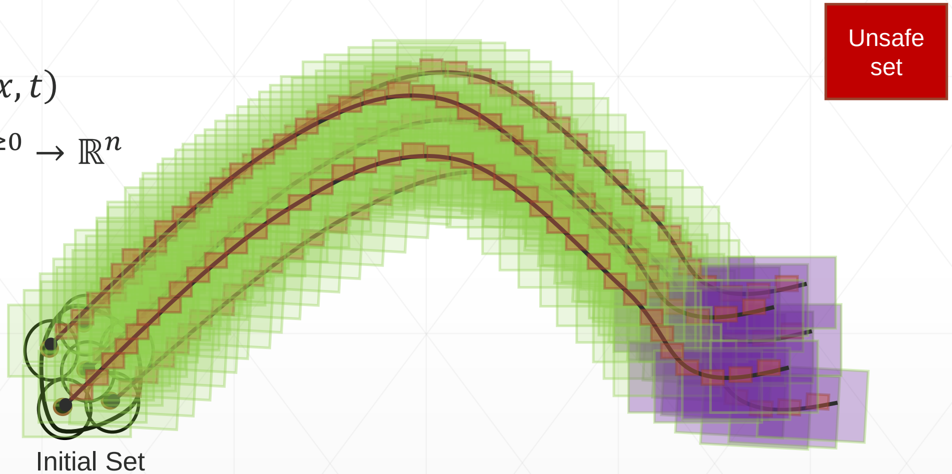
Known:
- Overapproximation of the set
- Upper bound on the order of overapproximation

Basic Algorithm

- Partition, Simulate, Bloat, Check, Refine

$$\dot{x} = f_i(x,t)$$

$$\xi_i \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R}^n$$



Switch to new mode

$$\dot{x} = f_{i+1}(x,t)$$

$$\xi_{i+1} \colon \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R}^n$$

Switching time interval $[t_1, t_2]$

New building block:
Provide guarantees for all executions from a set only from a known overapproximation of it

Unknown:
- Exact initial set for mode $i+1$

Known:
- Overapproximation of the set
- Upper bound on the order of overapproximation

Building block algorithm

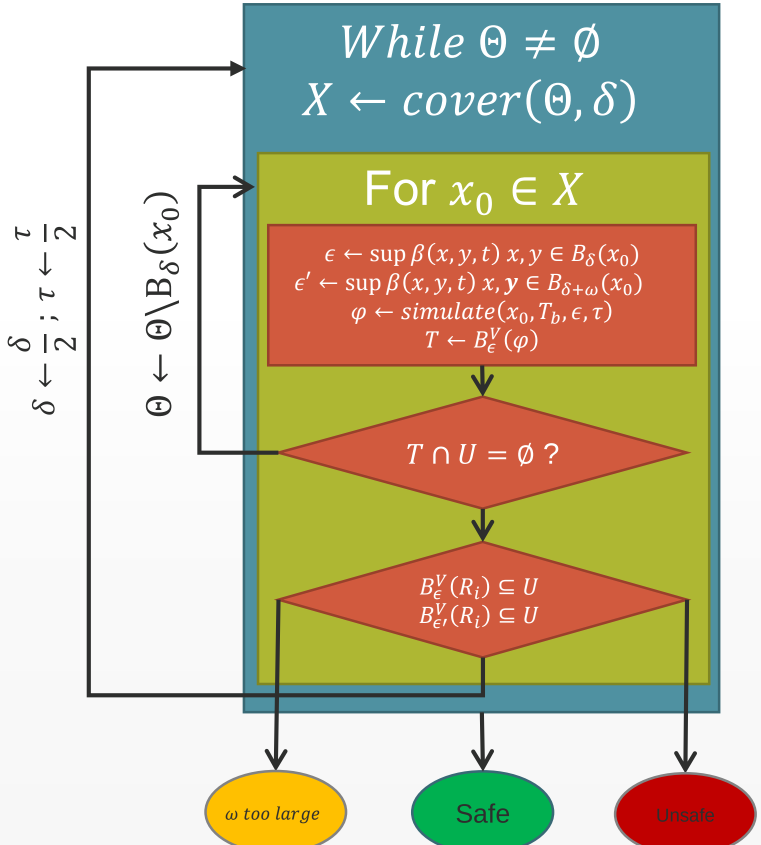
Input to the algorithm

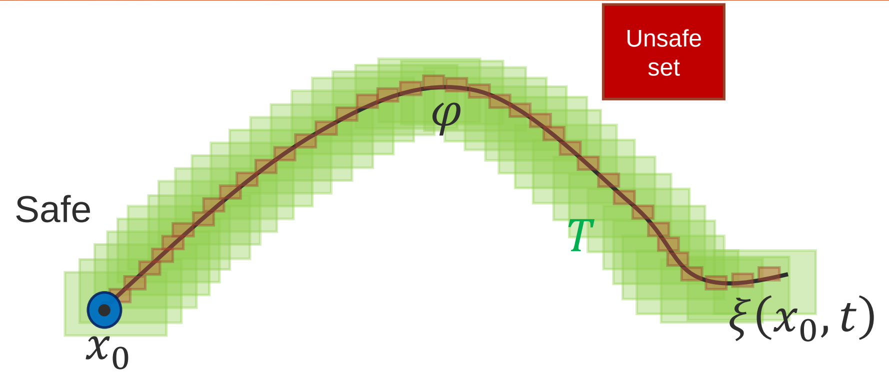
- Dynamics $\dot{x} = f(x,t)$ with annotation V and witness β
- Initial partitioning δ, time step τ, time bound T_b
- Are all executions from set I are safe?
- $I \subseteq \Theta$ and Θ is an ω – Over approximation of I

Building block algorithm

Input to the algorithm

- Dynamics $\dot{x} = f(x,t)$ with annotation V and witness β
- Initial partitioning δ, time step τ, time bound T_b
- Are all executions from set I are safe?
- $I \subseteq \Theta$ and Θ is an ω – Over approximation of I

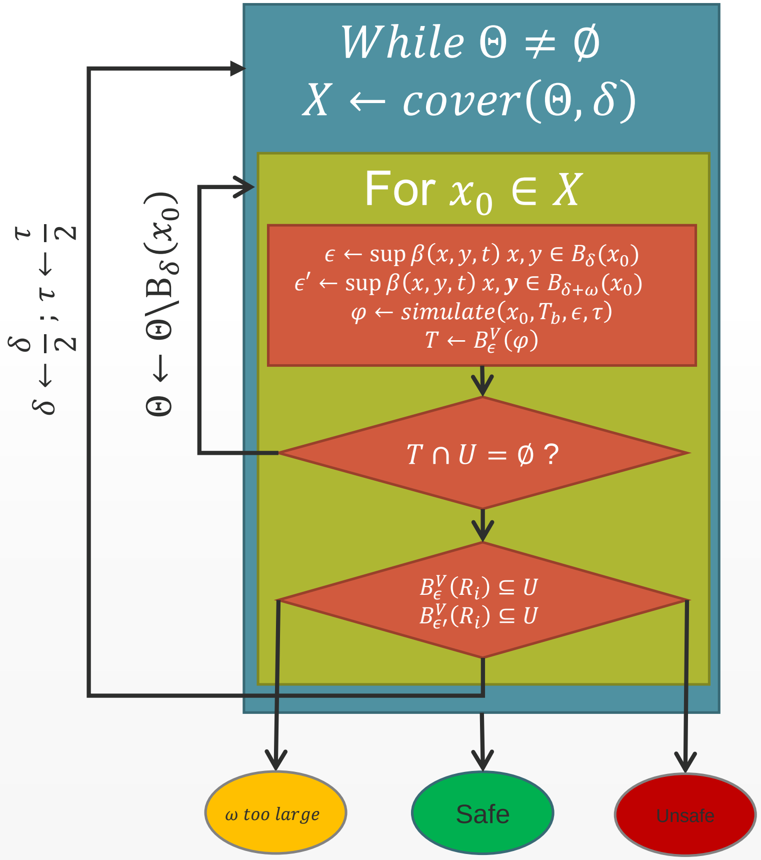


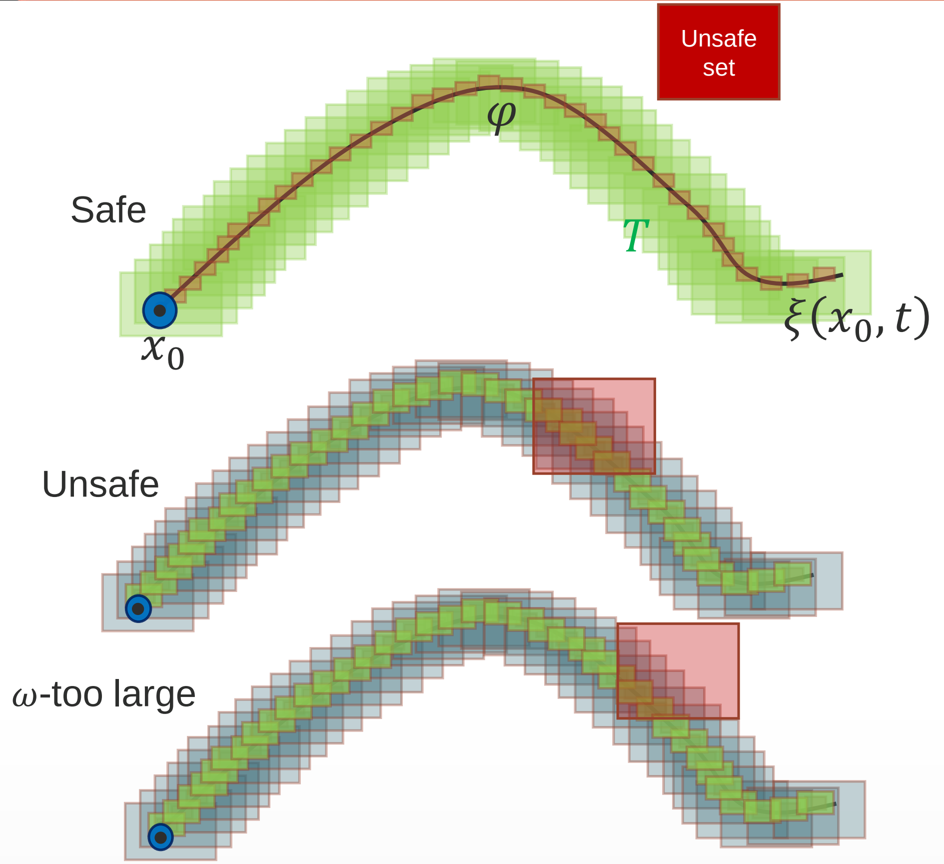


Building block algorithm

Input to the algorithm

- Dynamics $\dot{x} = f(x, t)$ with annotation V and witness β
- Initial partitioning δ, time step τ, time bound T_b
- Are all executions from set I are safe?
- $I \subseteq \Theta$ and Θ is an ω – Over approximation of I

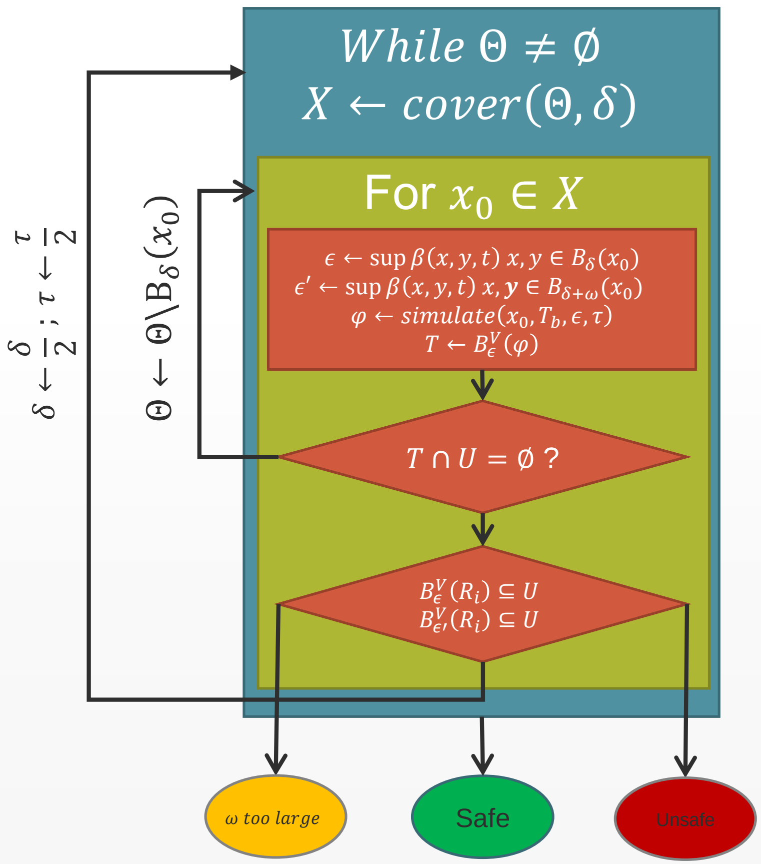


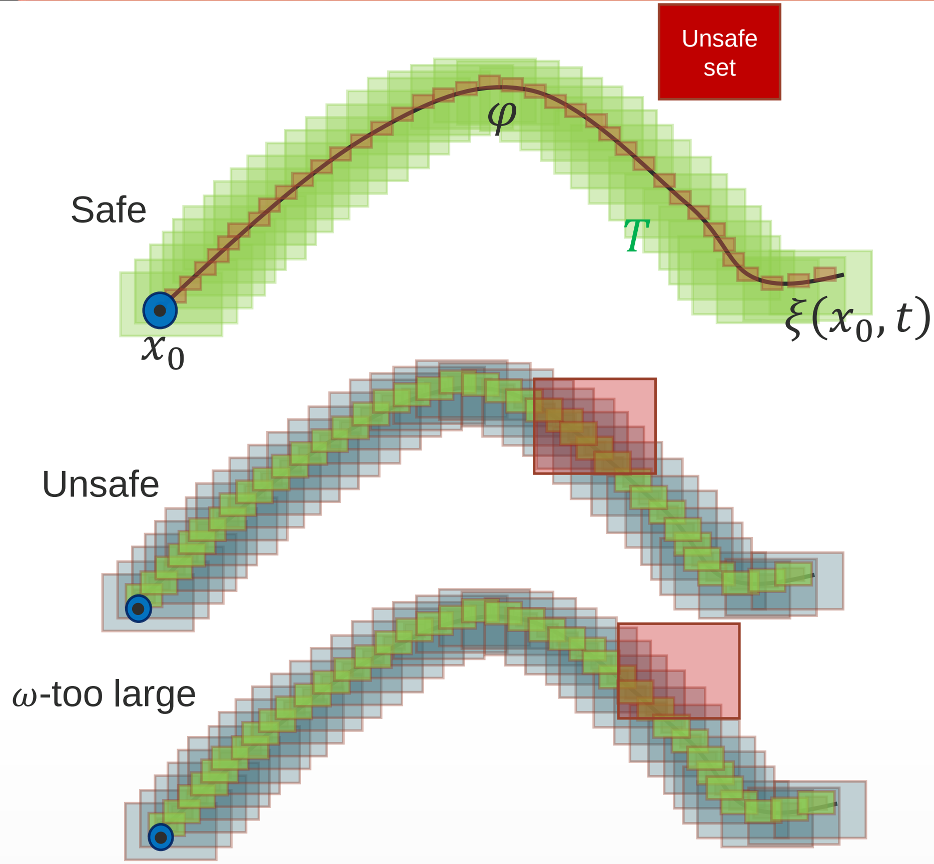


Building block algorithm

Input to the algorithm

- Dynamics $\dot{x} = f(x,t)$ with annotation V and witness β
- Initial partitioning δ, time step τ, time bound T_b
- Are all executions from set I are safe?
- $I \subseteq \Theta$ and Θ is an $\omega -$ Over approximation of I





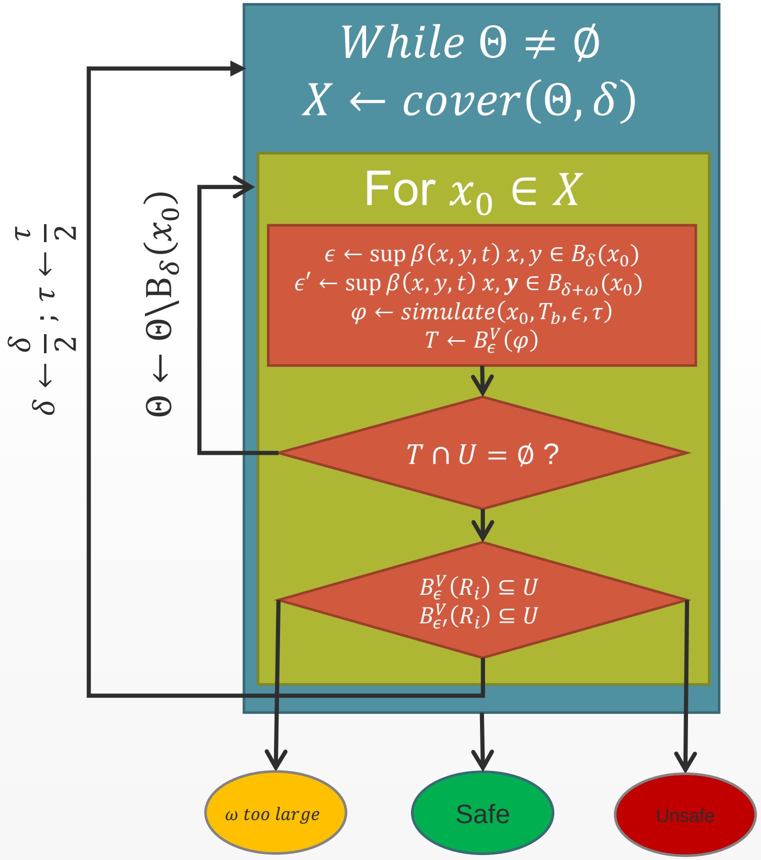
Soundness

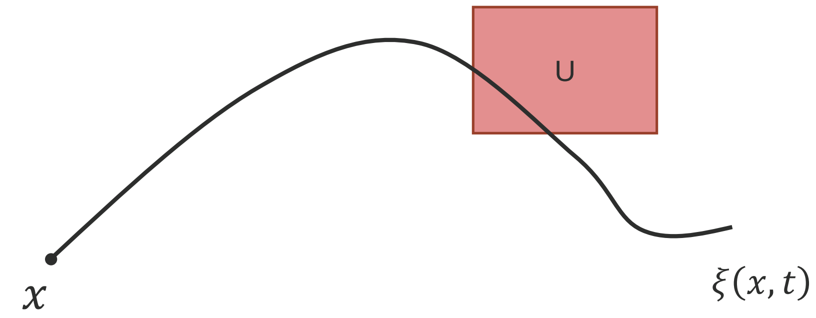
1. $\forall\ x \in B_\delta(x_0), \xi(x,t) \in B_\varepsilon^V(\varphi)$
2. $T \cap U = \emptyset$ implies all executions in $B_\delta(x_0)$ are safe
3. $B_{\delta+\omega}(x_0)$ contains at least one state from initial set
4. $B_{\epsilon'}^V(R_i)$ contains at least one reachable state
5. $B_{\epsilon'}^V(R_i) \nsubseteq U, B_\epsilon^V(R_i) \subseteq U$ then the initial over approximation ω is too large for inferring safe/unsafe.
6. $2\varepsilon + \varepsilon'$ is the upper bound on the over approximation

Building block algorithm

Input to the algorithm

- Dynamics $\dot{x} = f(x,t)$ with annotation V and witness β
- Initial partitioning δ, time step τ, time bound T_b
- Are all executions from set I are safe?
- $I \subseteq \Theta$ and Θ is an ω – Over approximation of I





Relative Completeness (when $\omega = 0$)

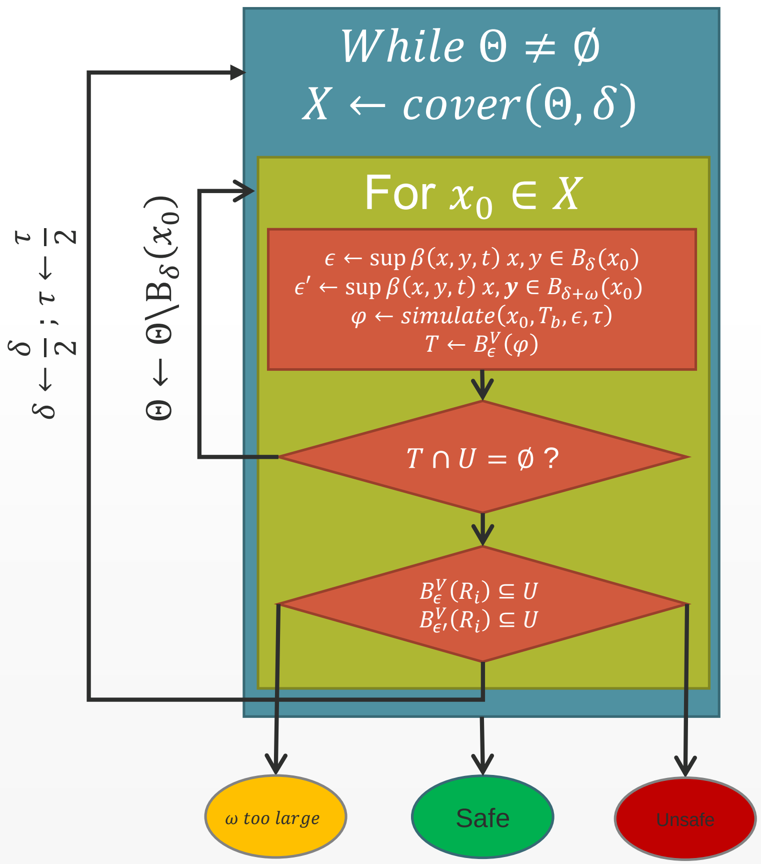
1. $\delta \to 0, \tau \to 0$ implies $\varepsilon \to 0, \varepsilon' \to 0$
2. If system is robustly safe, $\exists \delta, \tau$ such that all tubes are safe
3. Hence algorithm returns safe.
4. If system is unsafe, since U is unsafe,
   $\exists\ x, t, \varepsilon_1, B_{\varepsilon_1}\big(\xi(x,t)\big) \subseteq U$
5. Hence $\exists \delta, \tau$, such that $B_\epsilon^V(R_i) \subseteq U$ and $B_{\epsilon'}^V(R_i) \subseteq U$
6. Hence algorithm returns unsafe.

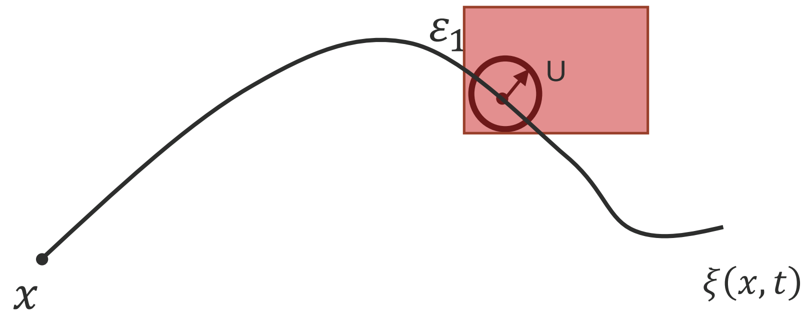
Also holds when $\omega \to 0$

Building block algorithm

Input to the algorithm

- Dynamics $\dot{x} = f(x,t)$ with annotation V and witness β
- Initial partitioning δ, time step τ, time bound T_b
- Are all executions from set I are safe?
- $I \subseteq \Theta$ and Θ is an ω – Over approximation of I





Relative Completeness (when $\omega = 0$)

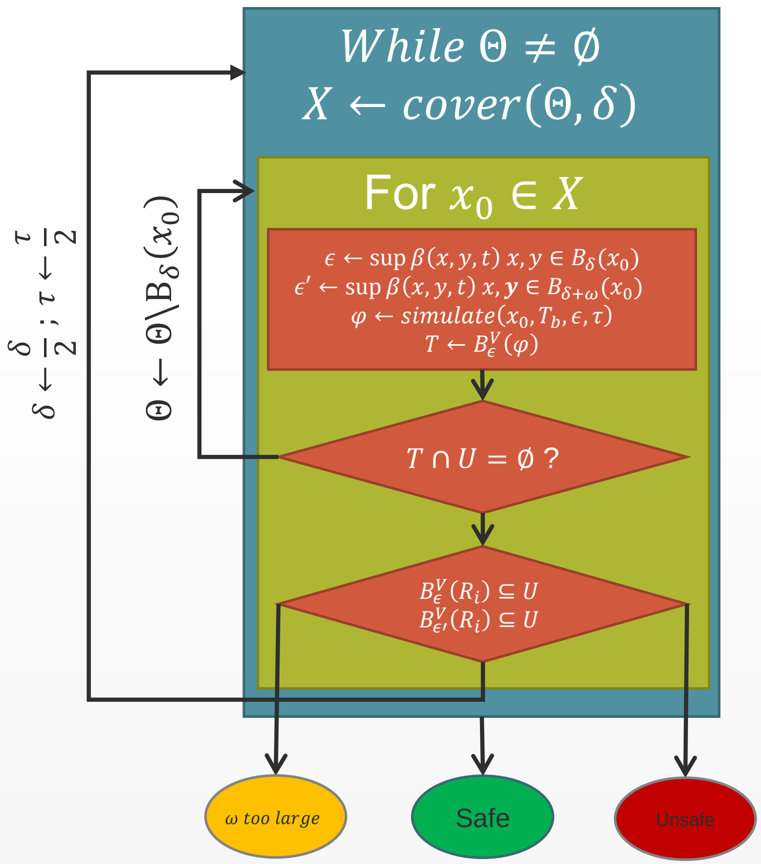
1. $\delta \to 0, \tau \to 0$ implies $\varepsilon \to 0, \varepsilon' \to 0$
2. If system is robustly safe, $\exists \delta, \tau$ such that all tubes are safe
3. Hence algorithm returns safe.
4. If system is unsafe, since U is unsafe,
   $\exists\ x, t, \varepsilon_1, B_{\varepsilon_1}\big(\xi(x,t)\big) \subseteq U$
5. Hence $\exists \delta, \tau$, such that $B_\epsilon^V(R_i) \subseteq U$ and $B_{\epsilon'}^V(R_i) \subseteq U$
6. Hence algorithm returns unsafe.

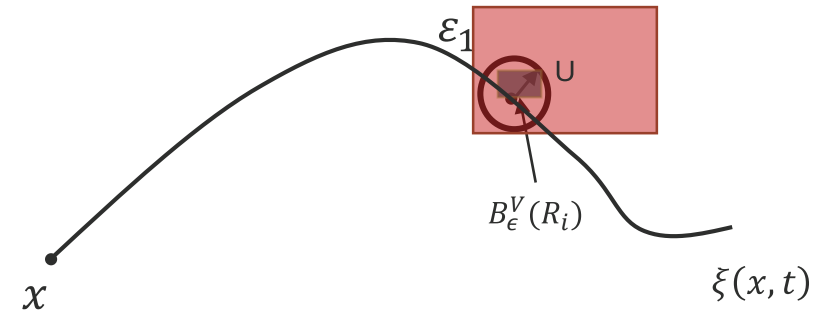
Also holds when $\omega \to 0$

Building block algorithm

Input to the algorithm

- Dynamics $\dot{x} = f(x,t)$ with annotation V and witness β
- Initial partitioning δ, time step τ, time bound T_b
- Are all executions from set I are safe?
- $I \subseteq \Theta$ and Θ is an ω – Over approximation of I





Relative Completeness (when $\omega = 0$)

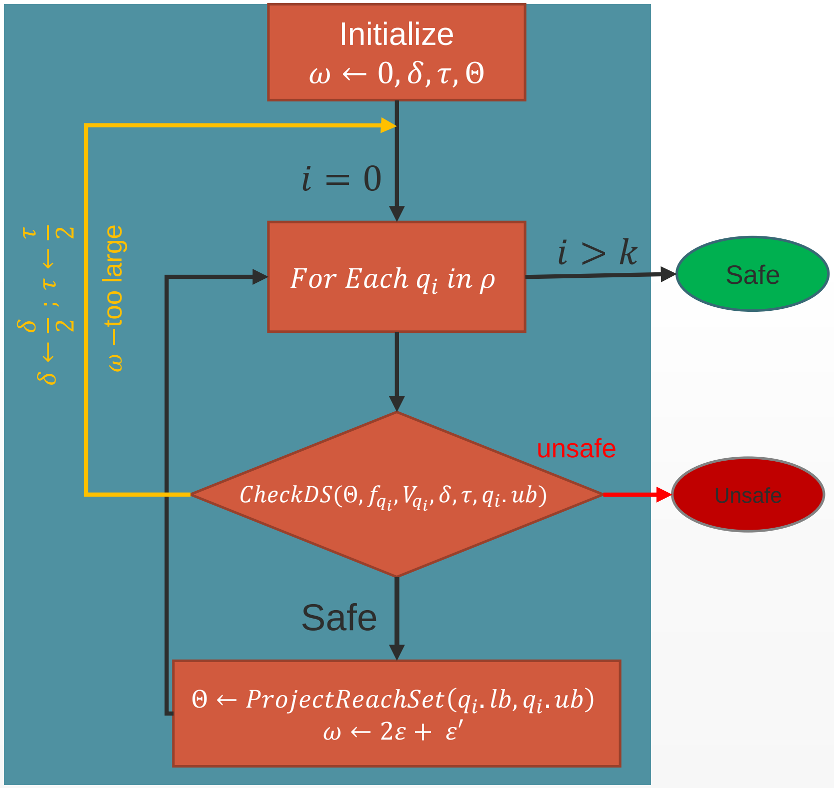
1. $\delta \to 0, \tau \to 0$ implies $\varepsilon \to 0, \varepsilon' \to 0$
2. If system is robustly safe, $\exists \delta, \tau$ such that all tubes are safe
3. Hence algorithm returns safe.
4. If system is unsafe, since U is unsafe,
   $\exists\ x, t, \varepsilon_1, B_{\varepsilon_1}(\xi(x,t)) \subseteq U$
5. Hence $\exists \delta, \tau$, such that $B_\epsilon^V(R_i) \subseteq U$ and $B_{\epsilon'}^V(R_i) \subseteq U$
6. Hence algorithm returns unsafe.

Also holds when $\omega \to 0$

Verification of Switched System

Input to the algorithm

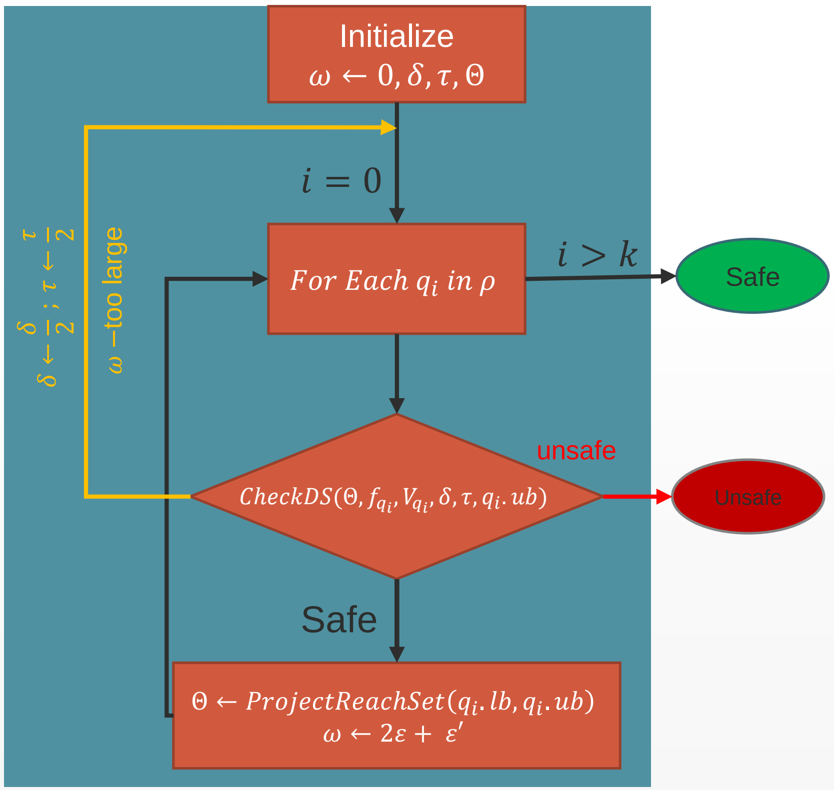
- Initial set Θ
- Dynamics $\{f_i\}_{i \in I}$ and annotations with witness V_i, β_i
- Switching interval sequence $\rho = q_0, q_1, \cdots, q_k$.
- Initial partitioning δ, time step τ

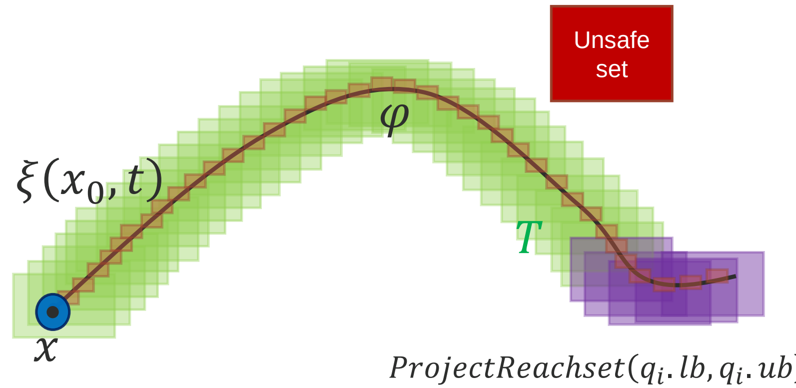


Verification of Switched System

Input to the algorithm

- Initial set Θ
- Dynamics $\{f_i\}_{i \in I}$ and annotations with witness V_i, β_i
- Switching interval sequence $\rho = q_0, q_1, \cdots, q_k$.
- Initial partitioning δ, time step τ

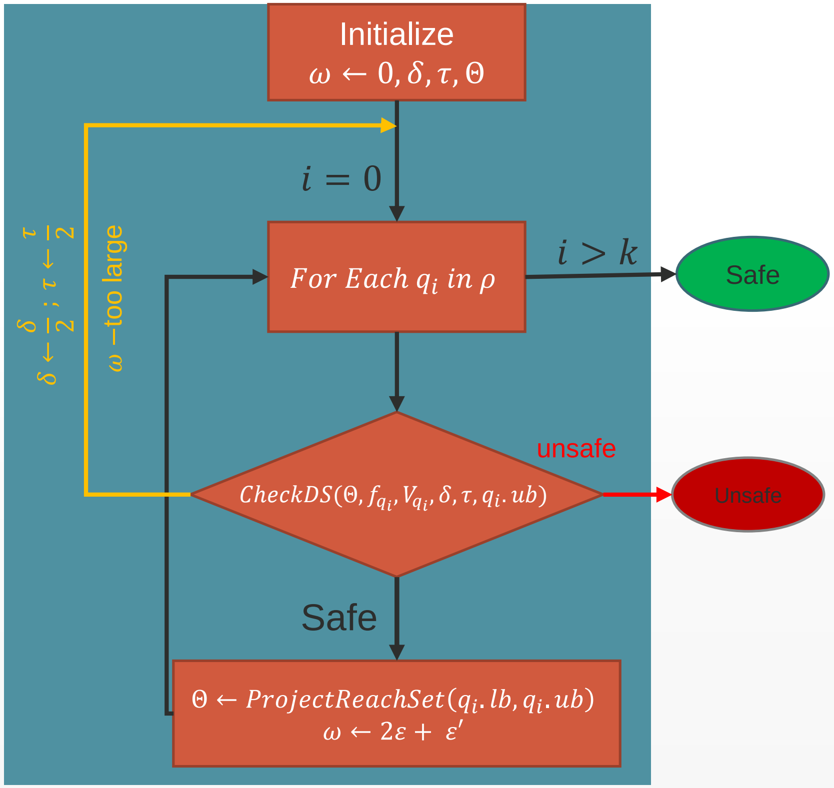


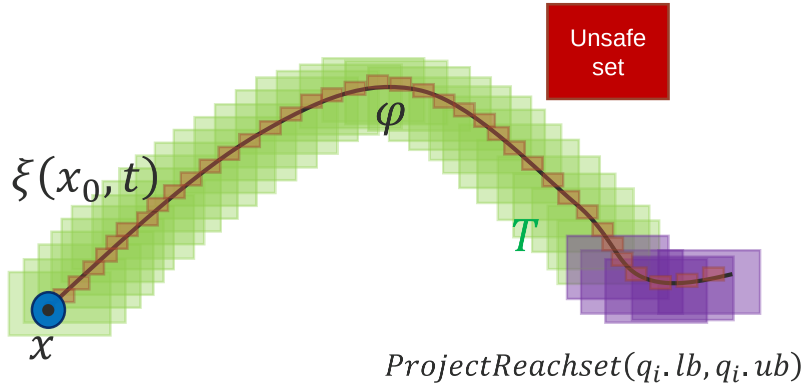


Verification of Switched System

Input to the algorithm

- Initial set Θ
- Dynamics $\{f_i\}_{i \in I}$ and annotations with witness V_i, β_i
- Switching interval sequence $\rho = q_0, q_1, \cdots, q_k$.
- Initial partitioning δ, time step τ





Soundness

1. Each call to $CheckDS(\Theta, f_{q_i}, V_{q_i}, \delta, \tau, q_i.ub)$ is sound
2. If the algorithm returns safe, all modes in ρ are safe
3. If algorithm returns unsafe, there is one mode in ρ that exhibits unsafe behavior

Relative Completeness

1. Order of over approximation for each subroutine call is bounded by $2\varepsilon + \varepsilon'$
2. As $\delta \to 0, \tau \to 0,\ \omega \to 0$
3. If all the modes are safe, $\exists\ \delta', \tau'$ that will prove safety
4. If at least one mode is unsafe, as $\omega \to 0$, the algorithm should return unsafe

Experimental Results

Benchmark	Variables	Time horizon	Refs.	Sims.	C2E2 (sec)	Flow* (sec)	Ariadne (sec)
Moore-G. Jet Engine	2	10	12	36	1.56	10.54	56.57
Brussellator	2	10	33	115	5.26	16.77	72.75
VanDerPol	2	10	5	17	0.75	8.93	98.36
Coupled VanDerPol	4	10	10	62	1.43	90.96	270.61
Sinusoidal Tracking	6	10	12	84	3.68	48.63	763.32
Linear Adaptive	3	6	8	16	0.47	NA	NA
Nonlinear Adaptive	2	10	16	32	1.23	NA	NA
Nonlinear Disturbance	3	10	22	48	1.52	NA	NA

Experimental Results

Benchmark	Variables	Time horizon	Refs.	Sims.	C2E2 (sec)	Flow* (sec)	Ariadne (sec)
Moore-G. Jet Engine	2	10	12	36	1.56	10.54	56.57
Brussellator	2	10	33	115	5.26	16.77	72.75
VanDerPol	2	10	5	17	0.75	8.93	98.36
Coupled VanDerPol	4	10	10	62	1.43	90.96	270.61
Sinusoidal Tracking	6	10	12	84	3.68	48.63	763.32
Linear Adaptive	3	6	8	16	0.47	NA	NA
Nonlinear Adaptive	2	10	16	32	1.23	NA	NA
Nonlinear Disturbance	3	10	22	48	1.52	NA	NA

Benchmark	Sims.	Time (sec)
12 fluid tanks (ft)	16	2.74
18 ft	76	15.28
24 ft	100	22.12
30 ft	124	28.82
3 vehicles 12 vars	32	5.68
16 vars	64	12.23
20 vars	128	25.14
24 vars	256	54.23

Switched-Nonlinear models

Conclusions

- Presented a notion of annotations for embedded systems.
- Sound and relative complete verification technique for nonlinear systems using executions
- Works for models with unknown parameters (adaptive control examples)
- Shows promise in scaling to higher dimensions

Future Work

- Extension to Hybrid Systems
- Automatically obtaining annotations from sample executions, Taylor Models or Lagrangian remainders.
- Approximate bisimulations from annotations.

Acknowledgements:

Funding: NSF

Discussions with Daniel Liberzon