www.unisi.it

1st HYCON PhD School on Hybrid Systems

www.ist-hycon.org

Reachability Analysis of Stochastic Hybrid Systems

Maria Prandini

Politecnico di Milano, Italy

prandini@elet.polimi.it

HYSCOM

IEEE CSS Technical Committee on Hybrid Systems

Information Society Technologies

Siena, July 19-22, 2005 - Rectorate of the University of Siena

Reachability Analysis for Stochastic Hybrid Systems: a Markov chain approximation method

Maria Prandini

Politecnico di Milano, Italy

E-mail: prandini@elet.polimi.it

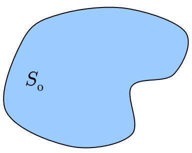
In collaboration with Jianghai Hu, Purdue University, and Shankar Sastry, University of California at Berkeley

Outline

- Reachability
  - Reachability & safety verification
  - Probabilistic safety
- Reachability computations for safety verification
- A Markov chain approximation method for probabilistic safety verification
- Application to aircraft conflict detection

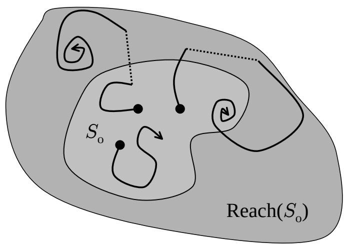
Reachability

Given a system and a set of initial conditions S_o
determine the set of states that can be reached by the system starting from S_o



Reachability

Given a system and a set of initial conditions S_o
determine the set of states that can be reached by the system starting from S_o

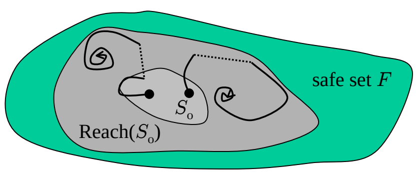


Safety verification

- In some systems, a region of the state space is "unsafe".
- One has to verify that the system operates in safe conditions, i.e., it keeps staying inside the safe set.
  If that is not the case the system has to be modified so as to guarantee safety.

Reachability & safety verification

Reachability analysis can be used for safety verification

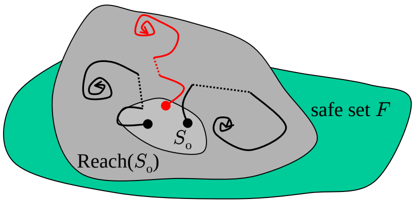


Reach(S_o) $\subset$ safe set F

the system is operating in safe conditions

Reachability & safety verification

Reachability analysis can be used for safety verification



Reach(S_o) $\not\subset$ safe set F

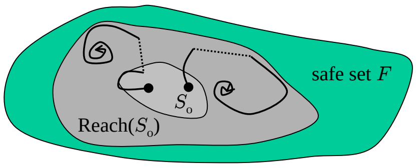
the system is operating in unsafe conditions

Safety for stochastic systems

In stochastic systems, trajectories are realizations of a stochastic process, and different realizations have different likelihood.

- if every realization keeps staying inside the safe set, then the system is 100% safe

100% safe $\leftrightarrow$
Reach(S_o) $\subset$ safe set F

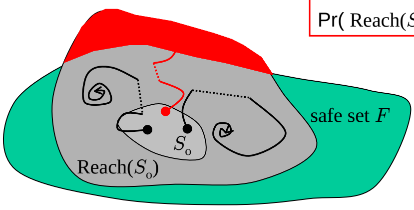


Safety for stochastic systems

In stochastic systems, trajectories are realizations of a stochastic process, and different realizations have different likelihood.

- if the set of realizations exiting the safe set has probability smaller than ε, then the system is 100(1-ε)% safe

100(1-ε)% safe $\leftrightarrow$
Pr(Reach(S_o)\ safe set F) < ε



Safety for stochastic systems

Two safety notions:

- every realization has to keep staying inside the safe set
  → worst-case safety
  - trajectories are considered all equally admissible as if the system were deterministic
  - conservative

- some realizations may exit the safe set, but this event has small probability
  → probabilistic safety
  - trajectories are weighted according to their likelihood
  - no 100% guarantees

Model checking

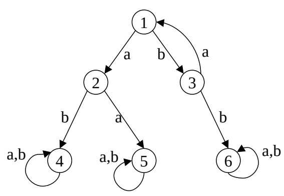
automatic methods for safety verification through reachability computations

- require to be able to "compute" with sets and probabilities (represent and propagate)
- mainly developed for deterministic systems (worst-case safety)

Deterministic finite automata

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\}$ $\equiv$ finite set of states
$\Sigma = \{a, b, c, \ldots\}$ $\equiv$ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S}$ $\equiv$ transition relation

$\mathcal{S} = \{1,2,3,4,5,6\}$ $\Sigma = \{a, b\}$
$T = \{(1,a,2),(1,b,3),(2,a,5),(2,b,4),(3,a,1),(3,b,6),(4,a/b,4),(5,a/b,5),(6,a/b,6)\}$

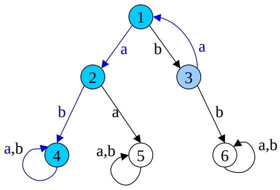


Graph representation

Deterministic finite automata: execution

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\} \equiv$ finite set of states
$\Sigma = \{a, b, c, \ldots\} \equiv$ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S} \equiv$ transition relation

execution $\equiv$ sequence of states $\{s_0, s_1, s_2, \ldots\}$ such that there exists a sequence of events $\{e_0, e_1, e_2, \ldots\}$ for which $(s_i, e_i, s_{i+1}) \in T$, $\forall i$

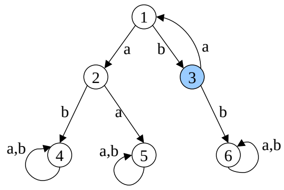


$\{3,1,2,4,4, \ldots\}$ is an execution

Deterministic finite automata: reach set

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\} \equiv$ finite set of states
$\Sigma = \{a, b, c, \ldots\} \equiv$ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S} \equiv$ transition relation

given a set of initial states $S_o \subset \mathcal{S}$:
$\text{Reach}(S_o) \equiv$ set of states $s \in \mathcal{S}$ for which there is a finite execution that starts in S_o and ends at s

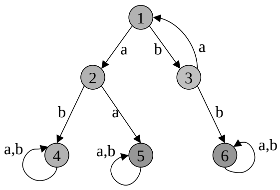


$S_o=\{3\}$

Deterministic finite automata: reach set

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\} \equiv$ finite set of states
$\Sigma = \{a, b, c, \ldots\} \equiv$ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S} \equiv$ transition relation

given a set of initial states $S_o \subset \mathcal{S}$:
$\text{Reach}(S_o) \equiv$ set of states $s \in \mathcal{S}$ for which there is a finite execution that starts in S_o and ends at s



$S_o=\{3\}$

reach set computation by listing all finite executions

$\{3,1,2,4\}$
$\{3,1,2,5\}$
$\{3,6\}$
…
finite executions starting from $s = 3$

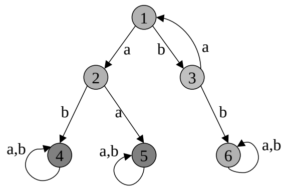
$\text{Reach}(S_o) = \mathcal{S}$

Deterministic finite automata: reach set

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\} \equiv$ finite set of states
$\Sigma = \{a, b, c, \ldots\} \equiv$ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S} \equiv$ transition relation

one-step successor operator:
$\text{Post}: 2^{\mathcal{S}} \rightarrow 2^{\mathcal{S}}$
$\text{Post}(A)=\{s' \in \mathcal{S}: \exists\, s \in A,\, e \in \Sigma,\, (s,e,s') \in T\}$

one-step successors of the set of states A



$S_o = \{3\}$

$\text{Reach}_0 = \{3\}$
$\text{Reach}_1 = \text{Reach}_0 \cup \text{Post}(\text{Reach}_0) = \{1,3,6\}$
$\text{Reach}_2 = \text{Reach}_1 \cup \text{Post}(\text{Reach}_1) = \{1,2,3,6\}$
$\text{Reach}_3 = \text{Reach}_2 \cup \text{Post}(\text{Reach}_2) = \mathcal{S}$
$\text{Reach}_4 = \text{Reach}_3$

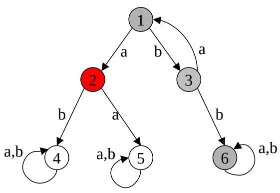
$\text{Reach}(S_o) = \mathcal{S}$

Deterministic finite automata: reach set

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\} \equiv$ finite set of states
$\Sigma = \{a, b, c, \ldots\} \equiv$ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S} \equiv$ transition relation

one-step successor operator:
$\text{Post}: 2^{\mathcal{S}} \rightarrow 2^{\mathcal{S}}$
$\text{Post}(A)=\{s' \in \mathcal{S}: \exists\, s \in A,\, e \in \Sigma,\, (s,e,s') \in T\}$

one-step successors of the set of states A



$S_o = \{3\}$

$\text{Reach}_0 = \{3\}$
$\text{Reach}_1 = \{1,3,6\}$
$\text{Reach}_2 = \{1,2,3,6\} \not\subset F$ → not safe

Safe set:
$F = \{1,3,4,5,6\}$

Safety verification algorithm

initialization: $\text{Reach}_{-1} = \emptyset$
$\text{Reach}_0 = S_o$
$i = 0$

algorithm can terminate immediately if one of the Reach_i is not included in F

loop: while $\text{Reach}_i \neq \text{Reach}_{i-1}$ and $\text{Reach}_i \subseteq$ safe set F do
$\text{Reach}_{i+1} = \text{Reach}_i \cup \text{Post}(\text{Reach}_i)$
$i = i + 1$

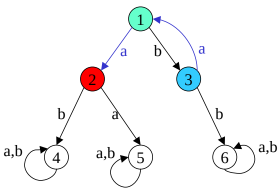
output: if $\text{Reach}_i = \text{Reach}_{i-1}$ then the system is safe else the system is not safe

Theorem: Since $\mathcal{S}$ is finite then
the algorithm can be implemented and always terminates.

Backward-reachability

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\}$ ≡ finite set of states
$\Sigma = \{a, b, c, \ldots\}$ ≡ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S}$ ≡ transition relation

given a set of final states $\mathcal{S}_f \subset \mathcal{S}$:
$\mathrm{BackReach}(\mathcal{S}_f)$ ≡ set of states $s \in \mathcal{S}$ for which there is a finite execution that starts in s and ends at $\mathcal{S}_f$



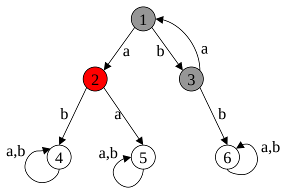
Unsafe set: $\mathcal{S}_f=\{2\}$ $\mathcal{S}_o = \{3\}$

$\mathrm{BackReach}(\{2\}) = \{1,2,3\} \cap \mathcal{S}_o \neq \emptyset$ → not safe

Backward-reachability

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\}$ ≡ finite set of states
$\Sigma = \{a, b, c, \ldots\}$ ≡ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S}$ ≡ transition relation

one-step predecessor operator:
$\mathrm{Pre}: 2^{\mathcal{S}} \to 2^{\mathcal{S}}$
$\mathrm{Pre}(A)=\{s \in \mathcal{S}: \exists\, s' \in A, e \in \Sigma, (s,e,s') \in T\}$



Unsafe set: $\mathcal{S}_f=\{2\}$ $\mathcal{S}_o = \{3\}$

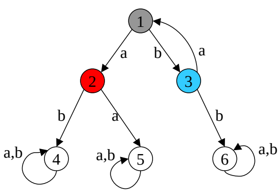
$\mathrm{BReach}_0 = \{2\}$
$\mathrm{BReach}_1 = \mathrm{BReach}_0 \cup \mathrm{Pre}(\mathrm{BReach}_0) = \{2,1\}$
$\mathrm{BReach}_2 = \mathrm{BReach}_1 \cup \mathrm{Pre}(\mathrm{BReach}_1) = \{2,1,3\}$
$\mathrm{BReach}_3 = \mathrm{BReach}_2 \cup \mathrm{Pre}(\mathrm{BReach}_2) = \{2,1,3\}$

$\mathrm{BackReach}(\{2\}) = \{1,2,3\} \cap \mathcal{S}_o \neq \emptyset$ → not safe

Backward-reachability

deterministic finite automaton
$\mathcal{S} = \{q_1, q_2, \ldots\}$ ≡ finite set of states
$\Sigma = \{a, b, c, \ldots\}$ ≡ finite set of input symbols (events)
$T \subset \mathcal{S} \times \Sigma \times \mathcal{S}$ ≡ transition relation

one-step predecessor operator:
$\mathrm{Pre}: 2^{\mathcal{S}} \to 2^{\mathcal{S}}$
$\mathrm{Pre}(A)=\{s \in \mathcal{S}: \exists\, s' \in A, e \in \Sigma, (s,e,s') \in T\}$



Unsafe set: $\mathcal{S}_f=\{2\}$ $\mathcal{S}_o = \{3\}$

$\mathrm{BReach}_0 = \{2\}$
$\mathrm{BReach}_1 = \{2,1\}$
$\mathrm{BReach}_2 = \{2,1,3\} \cap \mathcal{S}_o \neq \emptyset$ → not safe

Safety verification algorithm (backward procedure)

initialization: $\mathrm{BReach}_{-1} = \emptyset$
$\mathrm{BReach}_0 = \mathcal{S}_f$
$i = 0$

algorithm can terminate immediately if BReach_i intersects $\mathcal{S}_o$

loop: while $\mathrm{BReach}_i \neq \mathrm{BReach}_{i-1}$ and $\mathrm{BReach}_i \cap \mathcal{S}_o = \emptyset$ do
$\mathrm{BReach}_{i+1} = \mathrm{BReach}_i \cup \mathrm{Pre}(\mathrm{BReach}_i)$
$i = i + 1$

output: if $\mathrm{BReach}_i = \mathrm{BReach}_{i-1}$ then the system is safe else it is not safe

Theorem: Since $\mathcal{S}$ is finite then the algorithm can be implemented and always terminates.

Safety verification

Deterministic finite automata:

- sets & transitions can be represented by enumeration
- termination of the algorithm is guaranteed

Safety verification is *decidable*:
there exists a computational procedure that decides in a finite number of steps whether the system is safe or not.

- large-scale systems → state space explosion
- technical challenge: devise algorithms and data structure to handle large state spaces
  - binary decision diagrams to obtain a more compact, symbolic representation
  - semantic minimization to reduce the state space
  - …

Deterministic hybrid automata

hybrid automaton
$\mathcal{Q}$ ≡ set of discrete states
$\mathbb{R}^n$ ≡ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field
$\Phi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \times \mathbb{R}^n$ ≡ discrete transition (& reset)

Deterministic hybrid automata: execution

hybrid automaton

- $\mathcal{Q}$ ≡ set of discrete states
- $\mathbb{R}^n$ ≡ continuous state-space
- $f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field
- $\Phi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \times \mathbb{R}^n$ ≡ discrete transition (& reset)

execution ≡ pair of right-continuous signals q:[0,∞) → $\mathcal{Q}$, x:[0,∞) → $\mathbb{R}^n$ such that

1. q is piecewise constant and x is piecewise differentiable
2. on any interval (t_1,t_2) where q is constant and x is differentiable

$$x(t) = x(t_1) + \int_{t_1}^{t} f(q(t_1), x(\tau))d\tau, \quad \forall t \in [t_1, t_2)$$

3. $(q(t), x(t)) = \Phi(q^-(t), x^-(t)), \quad \forall t \geq 0$

Transition systems

hybrid automaton

- $\mathcal{Q}$ ≡ set of discrete states
- $\mathbb{R}^n$ ≡ continuous state-space
- $f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field
- $\Phi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \times \mathbb{R}^n$ ≡ discrete transition (& reset)

$\Downarrow$ same set of reachable states

transition system

- $\mathcal{S} = \mathcal{Q} \times \mathbb{R}^n$ ≡ set of states (infinite)
- $\Sigma = \{\tau, (q_i,q_j): q_i,q_j \in \mathcal{Q}\}$ ≡ alphabet of events:
  - τ is the continuous evolution event
  - (q_i,q_j) is a jump event
- $T \subset \mathcal{S} \times \Sigma \times \mathcal{S}$ ≡ transition relation

$((q_0,x_0), (q_0,q_f) , (q_f,x_f)) \in T$ if $(x_f, q_f) = \Phi(x_0, q_0)$

$((q_0,x_0), \tau, (q_0,x_f)) \in T$ if $\exists\, t_f > 0$ s.t. $\dot{x} = f(q_0, x),\ x(0) = x_0,\ x(t_f) = x_f$

$(x(t), q_0) = \Phi(x^-(t), q_0),\ \forall t \in (0, t_f)$

same (q_0,x_0) and τ appear in many distinct elements of T

Deterministic hybrid automata: reach set

Same algorithms as for the deterministic finite automata, but:

- the set of states $\mathcal{S} = \mathcal{Q} \times \mathbb{R}^n$ is not finite
- computation and representation of the successor/ predecessor of set A when the event is a continuous evolution:

  $\mathrm{Post}_c(A) = \{s' \in \mathcal{S} : \exists\, s \in A,\ e = \tau \in \Sigma,\ (s,e,s') \in T\}$

  $\mathrm{Pre}_c(A) = \{s \in \mathcal{S} : \exists\, s' \in A,\ e = \tau \in \Sigma,\ (s,e,s') \in T\}$

  is not simple (in general)

Safety verification

Deterministic hybrid automata:

- termination is not guaranteed in general
- set representation and propagation by continuous flow is difficult
  - exact methods for classes of systems with simple dynamics
  - approximation methods for more general classes of systems:
    - Over-approximation methods
    - Asymptotic approximation methods

Decidability results have been proven by using discrete abstraction for certain classes of hybrid automata: building a finite quotient transition system (deterministic finite automaton) that is “equivalent” to the original hybrid automaton for the purpose of safety verification

Asymptotic approximation methods

Aim:

obtaining an approximation of the reachable sets that converges to the true reachable sets as some accuracy parameter tends to zero

Characteristics

- can be applied to general classes of systems and they do not require a specific shape for the reachable sets
- reachability computations become more intensive as the dimension of the continuous state space grows

Stochastic finite automata

Markov chain

- $\mathcal{S} = \{q_1, q_2, \ldots\}$ ≡ finite set of states
- Φ: $\mathcal{S} \times \mathcal{S} \to [0,1]$ ≡ transition probability function

$\Phi(s, s')$ ≡ probability of transitioning to state s' when in state s

$$\sum_{s' \in S} \Phi(s, s') = 1, \quad \forall s \in S$$

$s \in \mathcal{S}$	$s' \in \mathcal{S}$	$\Phi(s,s')$
1	1	1
1	2	0
1	3	0
2	1	0.95
2	2	0
2	3	0.05
3	1	0.5
3	2	0
3	3	0.5

$\mathcal{S} = \{1,2,3\}$

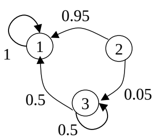
Stochastic finite automata

Markov chain
$$\begin{cases} \mathcal{S} = \{q_1, q_2, \ldots\} & \equiv \text{finite set of states} \\ \Phi:\ \mathcal{S}\times\mathcal{S}\to[0,1] & \equiv \text{transition probability function} \end{cases}$$

$\Phi(s, s')$ ≡ probability of transitioning to state s' when in state s

$$\sum_{s'\in S}\Phi(s,s') = 1, \quad \forall s\in S$$

$\mathcal{S} = \{1,2,3\}$



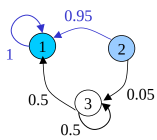
Stochastic finite automata: execution

Markov chain
$$\begin{cases} \mathcal{S} = \{q_1, q_2, \ldots\} & \equiv \text{finite set of states} \\ \Phi:\ \mathcal{S}\times\mathcal{S}\to[0,1] & \equiv \text{transition probability function} \end{cases}$$

execution ≡ sequence of states $\{s_0, s_1, s_2, \ldots\}$ such that $\Phi(s_i, s_{i+1})>0, \forall i$

$$P(s(0)=s_0,\ldots,s(k_f)=s_{k_f}) = \Pi_{i=1}^{t_f}\Phi(s_{i-1},s_i)\boxed{P_0(s_0)}$$

initial state probability distribution



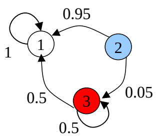
$\{2,1,1\}$ is a finite execution starting from 2

$$P_0(s) = \begin{cases} 1, & \text{if } s=2 \\ 0, & \text{otherwise} \end{cases}$$

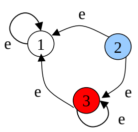
$$P(s(0)=2, s(1)=1, s(2)=1) = 0.95$$

Stochastic finite automata: worst-case safety

- One has to guarantee that every realization of the Markov chain process keeps staying inside the safe set



stochastic finite automaton



deterministic finite automaton

$$P_0(s) = \begin{cases} 1, & \text{if } s=2 \\ 0, & \text{otherwise} \end{cases} \qquad S_f = \{3\}$$

$S_o = \{2\}$ $S_f = \{3\}$

$S_o = \{2\}$ $S_f = \{3\}$

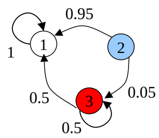
$\text{BReach}_0 = \{3\}$

$\text{BReach}_1 = \{3, 2\}$ $(\cap S_o \neq \emptyset)$

not 100% safe

Stochastic finite automata: probabilistic safety

- One can allow that some realizations of the Markov chain process exit the safe set, if this event has low probability



stochastic finite automaton

$$P_0(s) = \begin{cases} 1, & \text{if } s=2 \\ 0, & \text{otherwise} \end{cases} \qquad S_f = \{3\}$$

The realizations starting from state 2 that eventually reach the unsafe state 3 have probability 0.05.

95% safe

Probabilistic safety analysis

Markov chain
$$\begin{cases} \mathcal{S} = \{q_1, q_2, \ldots\} & \equiv \text{finite set of states} \\ \Phi:\ \mathcal{S}\times\mathcal{S}\to[0,1] & \equiv \text{transition probability function} \end{cases}$$

P_o ≡ initial state probability distribution over S_o

$$\boxed{P(s(k)\in S_f,\ \text{for some } k\in[0,k_f]) < \epsilon\,?}$$

modified Markov chain
$$\begin{cases} \mathcal{Q} = \{q_1, q_2, \ldots\} & \equiv \text{finite set of states} \\ \text{p}:\ \mathcal{Q}\times\mathcal{Q}\to[0,1] & \equiv \text{transition probability function} \end{cases}$$

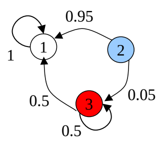
$$p(q,q') = \begin{cases} 1, & \text{if } q\in S_f \ \&\ q=q' \\ 0, & \text{if } q\in S_f \ \&\ q\neq q' \\ \Phi(q,q'), & \text{otherwise} \end{cases}$$

every state in the unsafe set becomes absorbing

P_o ≡ initial state probability distribution over S_o

$$\boxed{P(q(k_f)\in S_f) < \epsilon\,?}$$

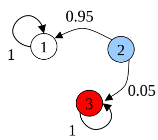
Probabilistic safety analysis



$$P_0(s) = \begin{cases} 1, & \text{if } s=2 \\ 0, & \text{otherwise} \end{cases}$$

$S_f = \{3\}$

same safety properties



$$P_0(s) = \begin{cases} 1, & \text{if } s=2 \\ 0, & \text{otherwise} \end{cases}$$

$S_f = \{3\}$

P-Safety verification: backward procedure

Markov chain
- $\mathcal{Q} = \{q_1, q_2, \dots\}$ ≡ finite set of states
- p: $\mathcal{Q} \times \mathcal{Q} \to [0,1]$ ≡ transition probability function

P_o ≡ initial state probability distribution over S_o

$$\boxed{P(q(k_f) \in S_f) < \epsilon\,?}$$

$$P(q(k_f) \in S_f) = \sum_{q \in \mathcal{Q}} \boxed{P(q(k_f) \in S_f | q(0) = q)} P_0(q)$$

Backward procedure for computing this conditional probability map

$$P(q(k_f) \in S_f | q(k+1) = q), q \in \mathcal{Q} \;\rightarrow\; P(q(k_f) \in S_f | q(k) = q), q \in \mathcal{Q}$$

P-Safety verification: backward procedure

Markov chain
- $\mathcal{Q} = \{q_1, q_2, \dots\}$ ≡ finite set of states
- p: $\mathcal{Q} \times \mathcal{Q} \to [0,1]$ ≡ transition probability function

P_o ≡ initial state probability distribution over S_o

$$\boxed{P(q(k_f) \in S_f | q(k+1) = q), q \in \mathcal{Q} \;\rightarrow\; P(q(k_f) \in S_f | q(k) = q), q \in \mathcal{Q}}$$

$$P(q(k_f) \in S_f | q(k) = q) = \sum_{q' \in \mathcal{Q}} \boxed{p(q,q')} \boxed{P(q(k_f) \in S_f | q(k+1) = q')}$$

probability of reaching q' from q in one step

probability of reaching the unsafe set starting from q' at time k+1

P-Safety verification: backward procedure

Markov chain
- $\mathcal{Q} = \{q_1, q_2, \dots\}$ ≡ finite set of states
- p: $\mathcal{Q} \times \mathcal{Q} \to [0,1]$ ≡ transition probability function

P_o ≡ initial state probability distribution over S_o

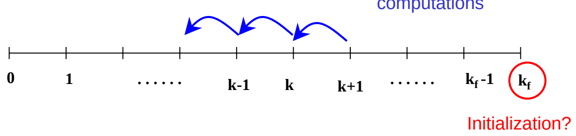
Define

$$P_c^{(k)}(q) := P(q(k_f) \in S_f | q(k) = q), q \in \mathcal{Q}$$

then

$$P_c^{(k)}(q) = \sum_{q' \in \mathcal{Q}} p(q,q') P_c^{(k+1)}(q')$$

backward reach computations



P-Safety verification: backward procedure

Markov chain
- $\mathcal{Q} = \{q_1, q_2, \dots\}$ ≡ finite set of states
- p: $\mathcal{Q} \times \mathcal{Q} \to [0,1]$ ≡ transition probability function

P_o ≡ initial state probability distribution over S_o

Define

$$P_c^{(k)}(q) := P(q(k_f) \in S_f | q(k) = q), q \in \mathcal{Q}$$

then

$$P_c^{(k)}(q) = \sum_{q' \in \mathcal{Q}} p(q,q') P_c^{(k+1)}(q')$$

Initialization

$$P_c^{(k_f)}(q) = \begin{cases} 1, & \text{if } q \in S_f \\ 0, & \text{otherwise} \end{cases}$$

<u>P-Safety verification algorithm</u>

initialization: $k = k_f - 1$

$$P_c^{(k_f)}(q) = \begin{cases} 1, & \text{if } q \in S_f \\ 0, & \text{otherwise} \end{cases}$$

loop: while $k \geq 0$ do

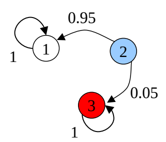
$$P_c^{(k)}(q) = \sum_{q' \in \mathcal{Q}} p(q,q') P_c^{(k+1)}(q')$$

$k = k-1$

output: if $P(q(k_f) \in S_f) = \sum_{q \in \mathcal{Q}} P_c^{(0)}(q) P_0(q) < \epsilon$

then the system is P-safe else the system is not P-safe

P-safety verification



$$P_0(s) = \begin{cases} 1, & \text{if } s = 2 \\ 0, & \text{otherwise} \end{cases}$$

$S_f = \{3\}$

$$P_c^{(k_f)}(q) = \begin{cases} 1, & \text{if } q = 3 \\ 0, & \text{otherwise} \end{cases}$$

$$P_c^{(k_f-1)}(q) = \sum_{q' \in \mathcal{Q}} p(q,q') P_c^{(k_f)}(q') = \begin{cases} 1, & \text{if } q = 3 \\ 0.05 \cdot 1 + 0.95 \cdot 0 = 0.05, & \text{if } q = 2 \\ 1 \cdot 0 = 0, & \text{if } q = 1 \end{cases}$$

$$P_c^{(k_f-2)}(q) = P_c^{(k_f-1)}(q) \quad \rightarrow P_c^{(0)}(q) = P_c^{(1)}(q) = \cdots = P_c^{(k_f-1)}(q) = \begin{cases} 1, & \text{if } q = 3 \\ 0.05, & \text{if } q = 2 \\ 0, & \text{if } q = 1 \end{cases}$$

$$P(q(k_f) \in S_f) = \sum_{q \in \mathcal{Q}} P_c^{(0)}(q) P_0(q) = 0.05$$

P-Safety verification algorithm

initialization: $k = k_f - 1$

$$P_c^{(k_f)}(q) = \begin{cases} 1, & \text{if } q \in S_f \\ 0, & \text{otherwise} \end{cases}$$

loop: while $k \geq 0$ do

$$P_c^{(k)}(q) = \sum_{q' \in \mathcal{Q}} p(q,q') P_c^{(k+1)}(q')$$

$k = k-1$

output: if $P(q(k_f) \in S_f) = \sum_{q \in \mathcal{Q}} P_c^{(0)}(q) P_0(q) < \epsilon$

then the system is P-safe else the system is not P-safe

If $k_f < \infty$ (finite time horizon) → the algorithm terminates

If $k_f = \infty$ (infinite time horizon) → convergence issue….

P-Safety verification algo: convergence

Define the column vector of unknowns for all safe states

$$\pi_c^{(k_f-k)} := [P_c^{(k)}(q)]_{q \in \mathcal{Q} \setminus S_f}$$

$$(P_c^{(k)}(q) = 1, \forall q \in S_f, \forall k)$$

then

$$\pi_c^{(k+1)} = \boxed{A}\, \pi_c^{(k)} + \boxed{b}, \quad \pi_c^{(0)} = 0$$

matrix of the transition probabilities between safe states $[p(q,q')]_{q,q' \in \mathcal{Q} \setminus S_f}$

column vector of the probabilities of reaching the unsafe set in one step $\sum_{q' \in S_f} p(q,q')$

P-Safety verification algo: convergence

Define the column vector of unknowns for all safe states

$$\pi_c^{(k_f-k)} := [P_c^{(k)}(q)]_{q \in \mathcal{Q} \setminus S_f}$$

$$(P_c^{(k)}(q) = 1, \forall q \in S_f, \forall k)$$

then

$$\pi_c^{(k+1)} = \boxed{A}\, \pi_c^{(k)} + \boxed{b}$$

discrete time system with constant input and state π_c

A has on each row positive elements whose sum is smaller or equal to 1

→ asymptotically stable → convergence of π_c to some (unique) equilibrium

Continuous stochastic systems: execution

continuous stochastic system
$$\begin{cases} \mathbb{R}^n & \equiv \text{continuous state-space} \\ b: \mathbb{R}^n \to \mathbb{R}^n & \equiv \text{drift} \\ \sigma: \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n & \equiv \text{diffusion} \end{cases}$$

$P_o \equiv$ initial state probability distribution over S_o

execution $\equiv$ solution to the stochastic differential equation (SDE)

$$dX = b(X)dt + \sigma(X)dW, X(0) \sim P_0$$

standard n-dimensional Brownian motion

Probabilistic safety analysis

continuous stochastic system
$$\begin{cases} \mathbb{R}^n & \equiv \text{continuous state-space} \\ b: \mathbb{R}^n \to \mathbb{R}^n & \equiv \text{drift} \\ \sigma: \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n & \equiv \text{diffusion} \end{cases}$$

$P_o \equiv$ initial state probability distribution over S_o

$$\boxed{P(X(t) \in S_f, \text{ for some } t \in [0, t_f]) < \epsilon \,?}$$

Problem to be Solved

Given the stochastic differential equation (SDE)

$$dX = b(X)dt + \sigma(X)dW$$

and a look-ahead time horizon $[0, t_f]$,

compute the probability

$$P_c = P(X(t) \in S_f \text{ for some } t \in [0, t_f]),$$

with initial condition $X(0) \sim P_0$.

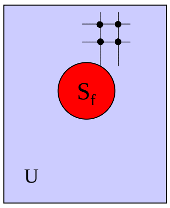
Impossible to solve analytically, in general.

Stochastic Approximation

- Idea: approximate the solution to the SDE with a Markov chain defined on some grid points

Find an open U containing S_f with compact support

Consider S = all the grid points $\delta\mathbf{Z}^2$ in $U \backslash S_f$



Stochastic Approximation

- Idea: approximate the solution to the SDE with a Markov chain defined on some grid points

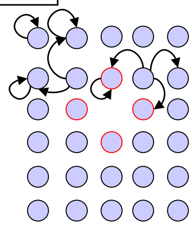
Find an open U containing S_f with compact support

Consider S = all the grid points $\delta\mathbf{Z}^2$ in $U \backslash S_f$

Define a Markov chain Q on S such that $Q \xrightarrow{d} X$ as $\delta \to 0$

For a small δ, compute P(Q reaches S_f first than U^c during $[0,t_f]$)

A good approximate of P_c



Weak Convergence of MC

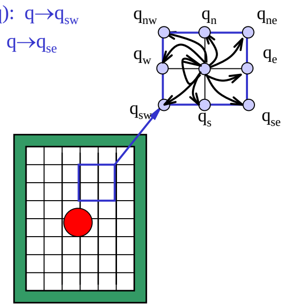
Each interior point q in S has eight neighbors: q_w, q_e, q_n, q_s, q_{nw}, q_{sw}, q_{ne}, q_{se}

$p_o^{(\delta)}(q)$: $q \to q$,

$p_w^{(\delta)}(q)$: $q \to q_w$, $p_{nw}^{(\delta)}(q)$: $q \to q_{nw}$, $p_{sw}^{(\delta)}(q)$: $q \to q_{sw}$

$p_e^{(\delta)}(q)$: $q \to q_e$, $p_{ne}^{(\delta)}(q)$: $q \to q_{ne}$, $p_{se}^{(\delta)}(q)$: $q \to q_{se}$

$p_n^{(\delta)}(q)$: $q \to q_n$, $p_s^{(\delta)}(q)$: $q \to q_s$



Weak Convergence of MC

Each interior point q in S has eight neighbors: q_w, q_e, q_n, q_s, q_{nw}, q_{sw}, q_{ne}, q_{se}

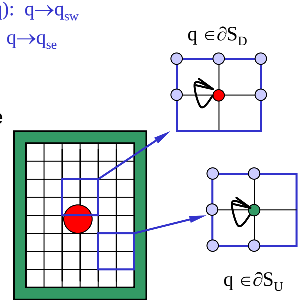
$p_o^{(\delta)}(q)$: $q \to q$,

$p_w^{(\delta)}(q)$: $q \to q_w$, $p_{nw}^{(\delta)}(q)$: $q \to q_{nw}$, $p_{sw}^{(\delta)}(q)$: $q \to q_{sw}$

$p_e^{(\delta)}(q)$: $q \to q_e$, $p_{ne}^{(\delta)}(q)$: $q \to q_{ne}$, $p_{se}^{(\delta)}(q)$: $q \to q_{se}$

$p_n^{(\delta)}(q)$: $q \to q_n$, $p_s^{(\delta)}(q)$: $q \to q_s$

Each point q in ∂S is an absorbing state



Weak Convergence of MC

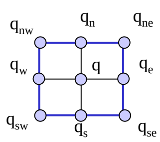
Each interior point q in S has eight neighbors: q_w, q_e, q_n, q_s, q_{nw}, q_{sw}, q_{ne}, q_{se}

$p_o^{(\delta)}(q)$: $q \to q$,

$p_w^{(\delta)}(q)$: $q \to q_w$, $p_{nw}^{(\delta)}(q)$: $q \to q_{nw}$, $p_{sw}^{(\delta)}(q)$: $q \to q_{sw}$

$p_e^{(\delta)}(q)$: $q \to q_e$, $p_{ne}^{(\delta)}(q)$: $q \to q_{ne}$, $p_{se}^{(\delta)}(q)$: $q \to q_{se}$

$p_n^{(\delta)}(q)$: $q \to q_n$, $p_s^{(\delta)}(q)$: $q \to q_s$



Each point q in ∂S is an absorbing state

Time it takes for each jump is $\Delta t^{(\delta)}$ ($\to 0$, as $\delta \to 0$)

Theorem: The Markov chain Q converges weakly to the solution X to the SDE on $U \backslash S_f$ with absorption on the boundary, if as $\delta \to 0$

1. $E_\delta[Q_{n+1}-Q_n | Q_n=q]/\Delta t^{(\delta)} \to b(q)$;
2. $E_\delta[(Q_{n+1}-Q_n)(Q_{n+1}-Q_n)^T | Q_n=q]/\Delta t^{(\delta)} \to \sigma(q)\sigma(q)^T$.

(local consistency conditions)

P-safety verification by MC approximation

continuous stochastic system
$$\begin{cases} \mathbb{R}^n & \equiv \text{continuous state-space} \\ b: \mathbb{R}^n \to \mathbb{R}^n & \equiv \text{drift} \\ \sigma: \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n & \equiv \text{diffusion} \end{cases}$$

$P_o \equiv$ initial state probability distribution over S_o

$\boxed{P(X(t) \in S_f, \text{ for some } t \in [0,t_f]) < \epsilon ?}$

$\Downarrow$ $P^{(\delta)}(q(k_f) \in S_f) \to P(X(t_f) \in S_f)$, as $\delta \to 0$

Markov chain
$$\begin{cases} \mathcal{Q} = \{q_1, q_2, \ldots\} & \equiv \text{finite set of states} \\ p^{(\delta)}: \mathcal{Q} \times \mathcal{Q} \to [0,1] & \equiv \text{transition probability function} \end{cases}$$

$P_o \equiv$ initial state probability distribution over $S_o \cap \delta\mathbf{Z}^2$

$\boxed{P^{(\delta)}(q(k_f) \in S_f) < \epsilon ?}$ $\quad (k_f := t_f/\Delta t)$

Transition Probabilities

Assume that $\sigma(x)= a(x)\, I$ (diagonal matrix)
One example of transition probabilities that work is

$$p_o^{(\delta)}(q)=\chi_q/C_q^{(\delta)}$$
$$p_w^{(\delta)}(q)=\exp(-\delta\xi_q)\,/C_q^{(\delta)},\quad p_e^{(\delta)}(q)=\exp(\delta\xi_q)\,/C_q^{(\delta)},$$
$$p_s^{(\delta)}(q)=\exp(-\delta\eta_q)\,/C_q^{(\delta)},\quad p_n^{(\delta)}(q)=\exp(\delta\eta_q)\,/C_q^{(\delta)},$$
$$p_{nw}^{(\delta)}(q)= p_{sw}^{(\delta)}(q) = p_{ne}^{(\delta)}(q)= p_{se}^{(\delta)}(q)=0$$

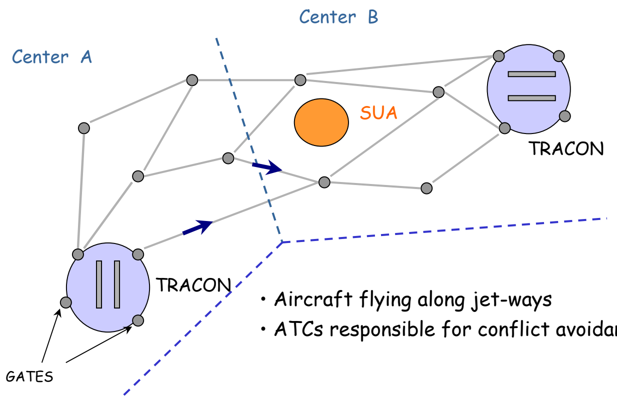
where

$$\xi_q=[b(q)]_x/a(q)^2,\quad \eta_q=[b(q)]_y/a(q)^2$$
$$\chi_q=2/(\lambda a(q)^2)-4,\qquad C_q^{(\delta)}=2\mathrm{csh}(\delta\xi_q)+2\mathrm{csh}(\delta\eta_q)+\chi_q$$
$$\Delta t= \lambda\delta^2,\ \text{for some } 0<\lambda<1/(2 \max a(q)^2)$$

P-safety verification by MC approximation

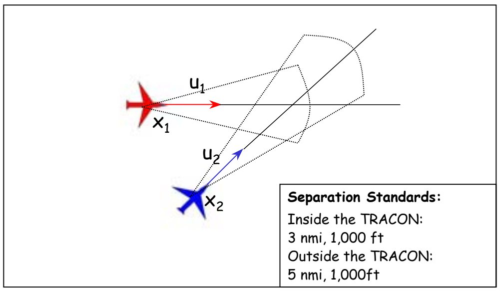
- Same backward procedure as for stochastic finite automata
- Extension to the case of SDE with time-varying drift & diffusion
- MC asymptotic approximation can used within a stochastic hybrid setting:
  - Time-driven switching
  - Jump Markov processes
  - SHS (Hu, Lygeros & Sastry)

Current ATMS architecture



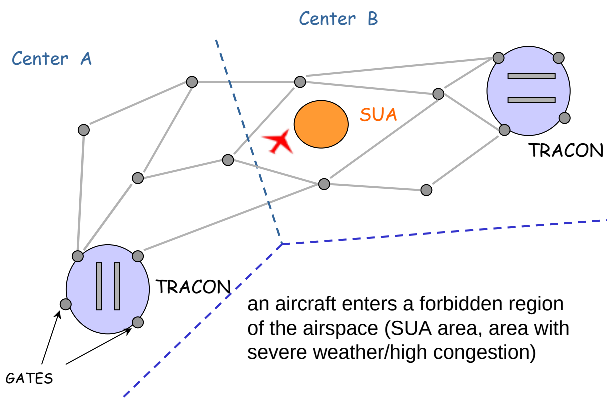
- Aircraft flying along jet-ways
- ATCs responsible for conflict avoidance

Aircraft-to-aircraft conflict



an aircraft comes closer than a minimum prescribed distance to another aircraft

Aircraft-to-airspace conflict



an aircraft enters a forbidden region of the airspace (SUA area, area with severe weather/high congestion)

Current ATMS initiatives

- Goal:
  - increasing the performance of the current network-based ATMS structure without reducing safety
- ATMS automation process:
  - assisting ATCs and pilots in detecting and solving potential situations of conflict

Mid-range conflict detection

- At the ATC level, tens of minutes horizon
- Introduction of a model for predicting the aircraft future position
- Evaluation of the possibility that a conflict would occur within a certain time horizon, based on this model

Aircraft Motion Model

Aircraft dynamics:

$$\underset{\text{aircraft position}}{\underset{\uparrow}{dX(t)/dt}} = \underset{\text{flight plan}}{\underset{\uparrow}{u(t)}} + \underset{\text{wind field}}{\underset{\uparrow}{f(X,t)}} + \sigma \underset{\text{noises}}{\underset{\uparrow}{w(X,t)}}$$

- Flight plan $u(t)$: deterministic, typically piecewise linear
- Wind field $f(x,t)$: deterministic, known from forecast or measurement
- Noises $w(x,t)$: random, modeling air turbulences and forecast/ measurement errors, modulated by σ

Observation: the closer the two aircraft, the more correlated the random perturbations to their velocities.

Random Field Perturbation

$B(x,t)$, the time integral of $w(x,t)$, is a spatially correlated Gaussian random field.

- For each fixed x, $B(x,t)$ is a standard Brownian motion
- $B(x,t)$ is time-increment independent
- For $t_1<t_2$, $\{B(x,t_2)-B(x,t_1), x\in\mathbf{R}^3\}$ is a collection of Gaussian random variables with zero mean and covariance

$$E\{[B(x,t_2)-B(x,t_1)][B(y,t_2)-B(y,t_1)]^T\}=\rho(x-y)\,(t_2-t_1)\,I_2\,,\ \ \forall x,y\in\mathbf{R}^3.$$

where ρ: $\mathbf{R}^2\to\mathbf{R}$ is a function with $\rho(0)=1$, $\rho(\Delta x)\to 0$ as $\Delta x\to\infty$.

Aircraft-to-Aircraft Conflict

- Two aircraft come too close to each other

dim=4

$$\boxed{\begin{aligned} dX_1(t) &= u_1(t)dt + f(X_1,t)dt + \sigma dB(X_1,t) \\ dX_2(t) &= u_2(t)dt + f(X_2,t)dt + \sigma dB(X_2,t) \end{aligned}}$$

$X = X_2-X_1$, $v = u_2-u_1$ $\Downarrow$ Assume $f(x,t)=R(t)x+d(t)$

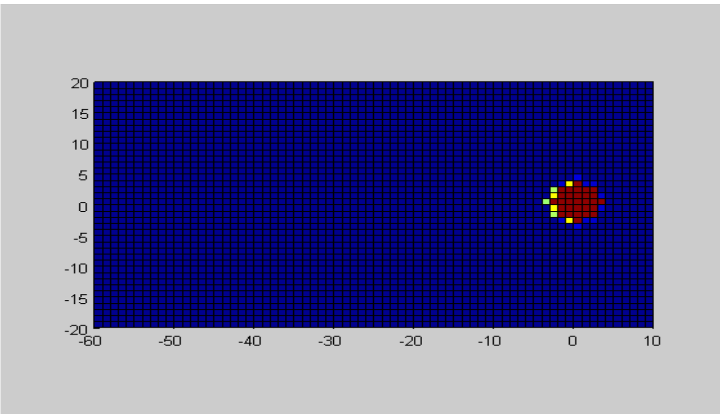
$$\boxed{dX(t) = v(t)dt + R(t)X(t)\,dt + \sigma d[B(X_2,t)-B(X_1,t)]}$$

$$\Downarrow$$

dim=2

$$\boxed{dX(t) = v(t)dt + R(t)X(t)\,dt + [2(1-\rho(Y))]^{1/2}\,\sigma\, dW(t)}$$

Conflict occurs when $X\in S_f$, where S_f is a circle



Aircraft-to-aircraft conflict
Time horizon t_f=20; No nominal wind; Relative velocity v(t)=(2,0); Spatial correlation $\rho(x)=\exp(-0.2\|x\|)$

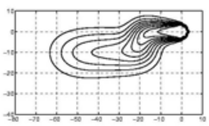
Example

Aircraft-to-aircraft (t_f =40)

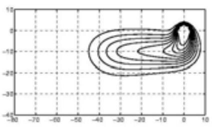
No wind

Relative velocity $v(t)=\begin{cases}(2,0), & t\in[0,10]\\(0,1), & t\in[10,20]\\(2,0), & t\in[20,40]\end{cases}$

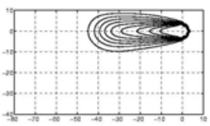
Spatial correlation $\rho(x)=\exp(-0.2\|x\|)$



t=0



t=10

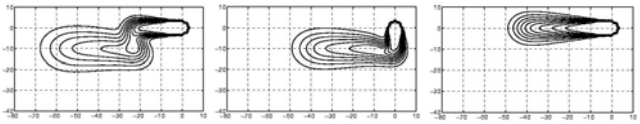


t=20

Example (more correlation)

Aircraft-to-aircraft (t_f =40)

Spatial correlation $\rho(x)=\exp(-0.05||x||)$



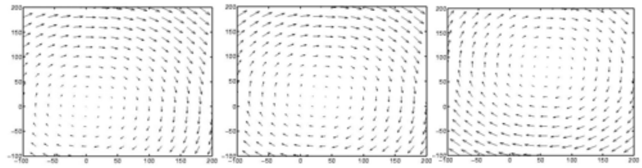
t=0 t=10 t=20

Spatial correlation does affect the probability of conflict

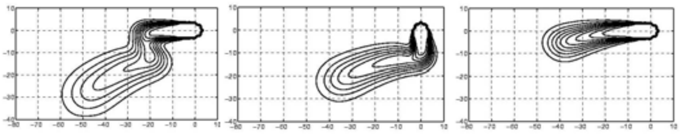
Larger spatial correlation results in P_c more concentrated along the projected collision course, and extended longer

Example

The effect of a swirling wind field

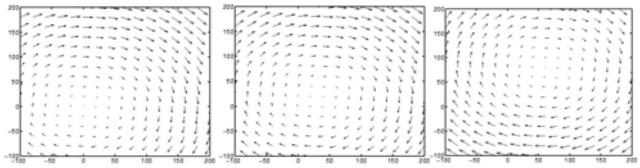


t=0 t=10 t=20

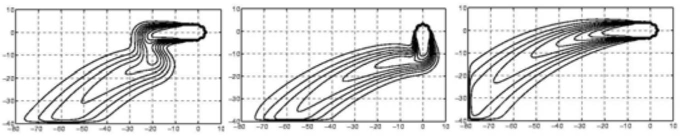


Example

Infinite horizon case ($t_f=\infty$)

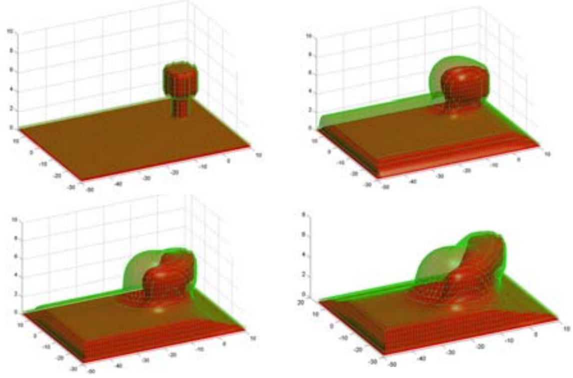


t=0 t=10 t=20



3D Forbidden Zone

Iso-probability surfaces (green 0.2, red 0.7)

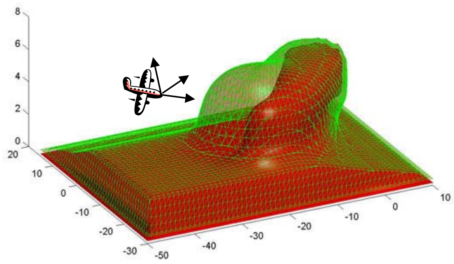


What can be done with the probability?

Of course, safety alert. But anything else?

Assist in designing feedback control to ensure safety

"Slide along a certain iso-surface"



References

"Reachability Analysis for Probabilistic Hybrid Systems with Application to Air Traffic Management"
Deliverable of the HYBRIDGE project (http://www.nlr.nl/public/hosted-sites/hybridge/)

J. Hu, M. Prandini
"Aircraft conflict detection: a method for computing the probability of conflict based on Markov chain approximation"
European Control Conference, Cambridge, UK, Sept. 2003

J. Hu, M. Prandini, S. Sastry
"Aircraft conflict prediction in presence of a spatially correlated wind field"
IEEE Trans. on Intelligent Transportation Systems, to appear.

H.J. Kushner, P.G. Dupuis
"Numerical methods for stochastic control problems in continuous time"
Springer-Verlag 2001.

X. D. Koutsoukos
"Optimal control of stochastic hybrid systems based on locally consistent Markov decision processes"
2005 IEEE Int. Symp. on Intelligent Control (ISIC '05), Cyprus, June, 2005.

Baier, B. Haverkort, Holger Hermanns, J-P. Katoen
"Automated performance and dependability evaluation using model checking"
Tutorial Proc. PERFORMANCE 2002, Springer LNCS 2459, 2002