Sensitive State-Space Exploration

Thao Dang, Alexandre Donzé, Oded Maler, Noa Shalev

VERIMAG Laboratory, University Joseph Fourier

December 11, 2008
Context

- Nonlinear systems with non-deterministic inputs (external disturbances, measurement errors etc)
- Safety specification: can the system reach a bad state?
- Reachability analysis untractable in general

Alternative solution

Simulation using efficient exploration heuristics

+ Sensitivity analysis to get local reachability information
Outline

Reachability Using Sensitivity for Systems with Inputs

Sensitive RRT-Based Exploration

Experimental Results
Outline

Reachability Using Sensitivity for Systems with Inputs

Sensitive RRT-Based Exploration

Experimental Results
System and Trajectories

Dynamical system of the form

\[
\dot{x}(t) = f(x(t), u(t))
\]  

◮ with trajectories of the form \( \xi : t \mapsto \xi(x, u(t), t) \) s.t. 
\( \xi(x, u(0), 0) = x \) and \( \xi \) satisfies (1)

◮ We discretize time with a fixed time step \( h > 0 \) and consider 
piecewise constant non-deterministic inputs:

\[
\forall t \in [t_i, t_i + h[, u(t) = u_i \in U
\]

◮ For a given a state \( x \) and input \( u \), we note 
\( x' = \xi(x, u) = \xi(x, u, h) \) the state reached from \( x \) using a 
constant input \( u \)
Sensitivity Analysis

We consider two trajectory points \( \xi(x, u) \) and \( \xi(y, u) \).
Sensitivity Analysis

We consider two trajectory points $\xi(x, u)$ and $\xi(y, u)$. The Taylor expansion of $\xi(x, u)$ w.r.t. $x$ gives

$$\xi(y, u) = \xi(x, u) + \frac{\partial\xi(x, u)}{\partial x} (y - x) + \varphi(h, y - x)$$

where $\varphi(h, y - x) = O\|y - x\|^2$.
Sensitivity Analysis

\[ S_p = \frac{\partial \xi(x, u)}{\partial x} \] is called the sensitivity matrix. We get the estimate

\[ \hat{\xi}(y, u) = \xi(x, u) + S_x(y - x) \]
Sensitivity Analysis

\[ S_p = \frac{\partial \xi(x, u)}{\partial x} \] is called the sensitivity matrix. We get the estimate

\[ \hat{\xi}(y, u) = \xi(x, u) + S_x(y - x) \]

Note  The estimate is exact if the the dynamics is affine
Let the set reachable at time $h$ be

$$
\mathcal{R}_h(\mathcal{X}_0) = \bigcup_{y \in \mathcal{X}_0} \xi(y, u)
$$
Reachable Set Estimation

Then by extension,

\[ \hat{\mathcal{R}}_h(x_0) = \bigcup_{y \in x_0} \hat{\xi}_y = \xi_x \oplus S_x(x_0 - x), \]

where \( \oplus \) is the Minkowski sum, is an estimate of \( \mathcal{R}_h(x_0) \), exact if the dynamics is affine.
Sensitivity to Non-Deterministic Inputs

Similarly, let \( S_u = \frac{\partial \xi(x,u)}{\partial u} \) and \( u \) and \( v \) two inputs then an estimate of \( \xi(y,v) \) is

\[
\hat{\xi}(y,v) = \xi(x,u) + S_x(y-x) + S_u(v-u)
\]

and the set reachable from \( X_0 \) using inputs in \( U \) can be estimated by

\[
\hat{R}_h(X_0,U) = \{\xi(x,u)\} \oplus S_x(X_0-x) \oplus S_u(U-u)
\]

Note that \( S_x \) and \( S_u \) can be computed efficiently during the simulation.
Reachable Set Around a Trace

Let \( \tau \) be a trace of the form \( \tau : x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \cdots \xrightarrow{u_N} x_N \)

Let \( S_{x_0}(ih) = \frac{\partial x_i}{\partial x_0} \) and \( S_{u_k}(ih) = \frac{\partial x_i}{\partial u_k} \) where \( i \geq k \), then

\[
\hat{R}_{Nh}(x_0, U) = \{x_N\} \oplus S_{x_0}(Nh)(x_0 - x_0) \oplus \bigoplus_{k=1}^{N} S_{u_k}(Nh)(U - u_k)
\]

is an estimate of the reachset at time \( T = Nh \) from \( x_0 \) using inputs in \( U \).
Reachable Set Around a Trace

Let $\tau$ be a trace of the form $\tau : x_0 \overset{u_1}{\rightarrow} x_1 \overset{u_2}{\rightarrow} \cdots \overset{u_N}{\rightarrow} x_N$

Let $S_{x_0}(i h) = \frac{\partial x_i}{\partial x_0}$ and $S_{u_k}(i h) = \frac{\partial x_i}{\partial u_k}$ where $i \geq k$, then

$$\hat{R}_{Nh}(X_0, U) = \{x_N\} \oplus S_{x_0}(Nh)(X_0 - x_0) \oplus \bigoplus_{k=1}^{N} S_{u_k}(Nh)(U - u_k)$$

is an estimate of the reachset at time $T = Nh$ from $X_0$ using inputs in $U$.

We can show that

$$S_{u_k}(Nh) = S_{u_k}((k + 1)h)S_{u_{k+1}}(Nh)$$

⇒ This makes it possible to compute $\hat{R}_{Nh}$ by induction in linear time w.r.t. to $N$

▶ But to get a precise reachset estimate, we need to partition $U$ into smaller subsets which produces an exponential number of traces
Outline

Reachability Using Sensitivity for Systems with Inputs

Sensitive RRT-Based Exploration

Experimental Results
The g-RRT algorithm

Definition (Exploration Trees)

An exploration tree for a dynamical system \((\mathcal{X}, U, \xi, \{x_0\})\) is a labeled tree \(\mathcal{T} = (S, E, x_0)\) where \(S \subseteq \mathcal{X}\) is a set of nodes, \(E \subseteq S \times U \times S\) is a set of labeled edges, and \(x_0\) is the root of the tree. An edge \((x, u, x')\) appears in the tree only if \(x' = \xi(x, u)\).
The g-RRT algorithm

Definition (Exploration Trees)

An exploration tree for a dynamical system \((X, U, \xi, \{x_0\})\) is a labeled tree \(T = (S, E, x_0)\) where \(S \subseteq X\) is a set of nodes, \(E \subseteq S \times U \times S\) is a set of labeled edges, and \(x_0\) is the root of the tree. An edge \((x, u, x')\) appears in the tree only if \(x' = \xi(x, u)\).

Procedure \(g\text{-RRT}(X, J)\)

\(T = (S = \{x\}, E = \emptyset, x)\) \hspace{2cm} \triangleright \text{initial tree}

for \(j = 1, 2 \ldots, J\) do

\(x' := \text{AddNew}(T, X)\) \hspace{2cm} \triangleright \text{explore a new node}

if \(x' \in B\) then \hspace{2cm} \triangleright \text{new state is bad}

return unsafe

end if

end for

return no violation found
The g-RRT algorithm

**Procedure** \( \text{AddNew}(T, \mathcal{X}) \)

\[x_g := \text{GuidedSampling}(S, \mathcal{X})\]
\[x_* := \arg \min_{x \in S} |x - x_g|\]
\[u_* := \arg \min_{u \in U} |\xi(x_*, u) - x_g|\]
\[x' := \xi(x_*, u_*)\]
\[S := S \cup \{x'\}\]
\[E := E \cup \{(x_*, u_*, x')\}\]

**return** \((x')\)
The g-RRT algorithm

**Procedure** \texttt{ADDNEW}(\mathcal{T}, \mathcal{X})

\begin{align*}
x_g &:= \text{GUIDEDSAMPLING}(S, \mathcal{X}) \\
x_* &:= \arg \min_{x \in S} |x - x_g| \\
u_* &:= \arg \min_{u \in U} |\xi(x_*, u) - x_g| \\
x' &:= \xi(x_*, u_*) \\
S &:= S \cup \{x'\} \\
E &:= E \cup \{(x_*, u_*, x')\} \\
\text{return} \ (x')
\end{align*}
The g-RRT algorithm

Procedure $\text{ADDNEW}(T, x')$

$x_g := \text{GUIDEDSAMPLING}(S, x)$

$x_* := \arg\min_{x \in S} |x - x_g|$

$u_* := \arg\min_{u \in U} |\xi(x_*, u) - x_g|$

$x' := \xi(x_*, u_*)$

$S := S \cup \{x'\}$

$E := E \cup \{(x_*, u_*, x')\}$

return $(x')$
The g-RRT algorithm

**Procedure** \texttt{ADDNEW}(T, x')

\begin{align*}
x_g & := \text{GUIDEDSAMPLING}(S, x) \\
x_* & := \arg \min_{x \in S} |x - x_g| \\
u_* & := \arg \min_{u \in U} |\xi(x_*, u) - x_g| \\
x' & := \xi(x_*, u_*) \\
S & := S \cup \{x'\} \\
E & := E \cup \{(x_*, u_*, x')\}
\end{align*}

**return** (x')
The g-RRT algorithm

**Procedure** \texttt{ADDNEW}(T, \mathcal{X})

\begin{align*}
x_g &:= \textbf{GUIDEDSAMPLING}(S, \mathcal{X}) \\
x_* &:= \arg \min_{x \in S} |x - x_g| \\
u_* &:= \arg \min_{u \in U} |\xi(x_*, u) - x_g| \\
x' &:= \xi(x_*, u_*) \\
S &:= S \cup \{x'\} \\
E &:= E \cup \{(x_*, u_*, x')\} \\
\text{return} \ (x')
\end{align*}

- The algorithm maintains a measure of the coverage of \( \mathcal{X} \) by \( T \)
- \textbf{GUIDEDSAMPLING} selects new points in order to improve this coverage
- We used the star-discrepancy measure defined in: 
  *Test Coverage for Continuous and Hybrid Systems*, T. Nahhal and T. Dang, CAV’07.
Combining g-RRT with Sensitivity

\[ \mathcal{T} := (x_0, \emptyset, x_0) \]

\[
\text{for } j = 1, 2 \ldots J \text{ do }
\]

\[
x' := \text{AddNew}(\mathcal{T}, \mathcal{X})
\]

\[
\text{if } x' \in B \text{ then }
\]

\[
\text{return unsafe}
\]

\[
\text{end if}
\]

\[
\text{if } |x' - B| \leq \epsilon \text{ then }
\]

\[
\tau := \text{TraceTo}(x')
\]

\[
\mathcal{R} := \text{ReachSens}(\tau)
\]

\[
\text{if } \mathcal{R} \cap B \neq \emptyset \text{ then }
\]

\[
\text{return not robustly safe}
\]

\[
\text{end if}
\]

\[
\text{end if}
\]

\[
\text{end for}
\]

\[
\text{return no violation found}
\]
Outline

Reachability Using Sensitivity for Systems with Inputs

Sensitive RRT-Based Exploration

Experimental Results
Experimental Results

Quadrirotor Helicopter with Input Disturbances

- 6-dimensional nonlinear model where the state variable is then \((\dot{x}, x, \dot{z}, z, \dot{\theta}, \theta)\)

- It is equipped with a linear quadratic regulator that drives it to a specified goal

- We assume that the velocity is estimated with an error \(\tilde{\dot{\theta}} = \dot{\theta} + w\)

- We want to verify that \(\theta\) remains less than .15
Quadritor Helicopter with Input Disturbances

(a) (b)
A Robot Navigation System Benchmark

- Dynamics of a car with 5 variables: \( \dot{x} = v \cos(\theta), \dot{y} = v \sin(\theta), \)
  \( \dot{\theta} = v \tan(\phi)/L, \dot{v} = u_0, \dot{\phi} = u_1 \)
where \( x, y \): position, \( \theta \): heading of the car, \( v \) is its speed and \( \phi \) is steering angle.
Control inputs \( u_0 \) and \( u_1 \): acceleration and steering.

- Hybrid control law with 3 modes
  - Mode 1 *RandomDriver*, the control inputs selected uniformly at random between their lower and upper bounds.
  - Mode 2 *StudentDrive*, when the speed is low, \( u_0 \) is randomly chosen as in first mode; otherwise, reduce the speed.
  - Mode 3, called *HighwayDrive*, reduce the speed when it is high and increase it when it is low.

- In our experiments, the system selects a mode randomly and use it for some fixed amount of time.
Experimental Results

Robot Navigation System - Results

- A safety property is specified by a bad set
  \[ x \in [20.0, 21.0] \land y \in [20.0, 21.0] \land \theta \in [1.2, 1.25], \]

- We compared the performance of the sg-RRT algorithm (with sensitivity) with that of the standard g-RRT algorithm on this example

- sg-RRT discovers that the system is not robustly safe after 7643 iterations in 0.56 minutes of computation time

- g-RRT needs 20751 iterations to detect the violation of the property in about 1 minute.
## High-dimensional Linear systems

<table>
<thead>
<tr>
<th>dim n</th>
<th>Nb of Iter until detection</th>
<th>Time</th>
<th>Time for 10000 iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sg-RRT</td>
<td>g-RRT</td>
<td>sg-RRT</td>
</tr>
<tr>
<td>10</td>
<td>7653</td>
<td>11090</td>
<td>7s</td>
</tr>
<tr>
<td>20</td>
<td>7720</td>
<td>10025</td>
<td>11s</td>
</tr>
<tr>
<td>100</td>
<td>9920</td>
<td>11152</td>
<td>36s</td>
</tr>
<tr>
<td>200</td>
<td>3035</td>
<td>11243</td>
<td>21s</td>
</tr>
<tr>
<td>200</td>
<td>125</td>
<td>&gt;50000</td>
<td>2s</td>
</tr>
</tbody>
</table>

Experimental results for random linear systems
Conclusion

- A new method for the verification of nonlinear systems with disturbances
- Combination of two methods that proved their efficiency

Future work

- Investigate other exploration strategies
- Implement optimization heuristics (e.g. merging of reachsets)
- Extend the method to find quickly concrete counter examples