Automaton-guided Controller Synthesis for Nonlinear Systems with Temporal Logic

Eric M. Wolff$^1$
Ufuk Topcu$^2$ and Richard M. Murray$^1$
$^1$Caltech and $^2$UPenn

IROS
6 November 2013
Modern Autonomous Systems

• How to specify complex tasks?
• How to handle high-dimensional and nonlinear dynamics?

Caltech

NASA/JPL

http://www.andrewalliance.com/

US Navy
Main Contributions

• Trajectory generation for high-dimensional (10+ dim) and nonlinear systems with complex tasks

• Solves problems that discrete abstraction techniques cannot
Nonlinear System Model

- Discrete-time nonlinear system
  \[ x_{t+1} = f(x_t, u) \]
  \[ x \in X \subseteq \mathbb{R}^n, \quad u \in U \subseteq \mathbb{R}^m \]

- Labels: \( L : X \rightarrow 2^{AP} \)

- Trajectory:
  \[ x = x(x_0, u) = x_0 x_1 x_2 \ldots \]
  \[ x_{t+1} = f(x_t, u) \text{ for some } u \in U \text{ for } t = 0, 1, \ldots \]

- Word: \( L(x) = L(x_0) L(x_1) L(x_2) \ldots \)
Temporal Logic

- A logic for reasoning about how properties change over time
- Reason about infinite sequences $\sigma = s_0s_1s_2 \ldots$ of states
- Propositional logic: $\land$ (and), $\lor$ (or), $\implies$ (implies), $\neg$ (not)
- Temporal operators: $\mathcal{U}$ (until), $\bigcirc$ (next), $\Box$ (always), $\Diamond$ (eventually)
**Spec:** Avoid obstacles, pick-up supplies at region A and then do surveillance on regions B and C.
Problem Statement

• **Given:**
  – a discrete-time nonlinear system,
  – an initial state $x_0$,
  – a temporal logic task $\varphi$

• **Goal:** Find a control input sequence $u$ such that $L(x(x_0,u)) \models \varphi$. 
Related Work

• **Discrete abstractions** (Alur00, Belta06, Habets06, Gol12, Kloetzer08, Pappas06, Tabuada06, Wongpiromsarn10, Yordanov13)

**Low dimensional systems (<= 6)**
Related Work

- **Discrete abstractions** (Alur00, Belta06, Habets06, Gol12, Kloetzer08, Pappas06, Tabuada06, Wongpiromsarn10, Yordanov13)
  - Low dimensional systems ($\leq 6$)

- **Counter-example guided abstraction refinement** (Alur03, Clarke00, Stursberg05)
  - Limited task search

- **Hierarchical LTL motion planning** (Bhatia11, Plaku10)
  - Limited specs & only sampling-based
From Logic to Automaton

Informal Task

Avoid obstacles, pick-up supplies at region A and then do surveillance on regions B and C.

Automatic translation from logic to automaton!

Gastin, Oddoux: http://www.lsv.ens-cachan.fr/~gastin/ltl2ba/
Automaton-guided Solution

• **Main idea:** Logic automaton guides a series of constrained reachability computations
Automaton-guided Solution

- Automaton path: $q_0(q_1q_2q_3)^\omega$
- Reachability: $(S,A) [(S,B)(S,C)(\emptyset,S)]^\omega$
Automaton-guided Solution

Environment

Logic Automaton

- Automaton path: $q_0(q_1q_2q_3)\omega$
- Reachability: $(S,A) [(S,B)(S,C)(\emptyset,S)]\omega$
Automaton-guided Solution

Environment

Logic Automaton

- Automaton path: $q_0(q_1q_2q_3)^\omega$
- Reachability: $(S,A) [(S,B)(S,C)(\emptyset,S)]^\omega$
Constrained Reachability

- **Given:** safe set $X_1 \subseteq X$, reach set $X_2 \subseteq X$

- **Goal:** Find a control input $u$ and a horizon length $N$ such that $x_1, \ldots, x_{N-1} \in X_1, x_N \in X_2$, and $x_{t+1} = f(x_t, u)$ for $t = 1, \ldots, N - 1$

- **CstReach($X_1, X_2$)**
  - robotic motion planning (LaValle)
  - optimization-based methods (Betts, Milam)
  - PDE-based methods (Mitchell)
Solving Constrained Reachability

• CstReach($X_1, X_2$) can be encoded as a mixed-integer linear program

• Enforce that state is in $X_1$ until $X_2$ (big-M)

\[ H_i x \leq K_i + M (1 - z_i), \quad z_i \in \{0,1\}, \quad \sum z_i = 1 \]

• Dynamics are independent of state constraints
What if a Path is Infeasible?

- **Issue:** Lots of paths in automaton to check

- **Fixes:**
  1) Parallel constrained reachability computations
  2) Update ordering on paths
Examples

• Systems
  – Quadrotor (10 dim)
  – Chained integrators (4, 12, 20 dim)
  – Car-like robot (nonlinear + drift)

• Specifications
  – Visit $n$ goals
  – Repeatedly visit $n$ goals
Examples

Model: car-like robot (5 dim)
Spec: Repeatedly visit 3 regions and avoid obstacles
Examples

**Model**: quadrotor (10 dim)
**Spec**: Repeatedly visit 4 regions and avoid obstacles
Examples

Model: quadrotor (10 dim)
Spec: Visit all 4 regions and avoid obstacles
Solver Time: Goals

Abstractions
2-dim: seconds
4-dim: minutes
6-dim: hours
8-dim: ??
Solver Time: Surveillance

Abstractions
2-dim: seconds
4-dim: minutes
6-dim: hours
8-dim: ??

Solver time (sec)

\(|n| = 2, 4, 6, 8|

- car
- chain2
- chain6
- chain10
- quadrotor
Conclusions

• Contributions
  – Temporal logic motion planning for **high dimensional** and **nonlinear** systems
  – Significant improvement on standard techniques

• Future work
  – Varying levels of abstraction
  – Improved discrete search
  – Multi-agent
Thank you!

• **Contact:** Eric M. Wolff
  – Email: ewolff@caltech.edu
  – Web: www.cds.caltech.edu/~ewolff/

• **Funding:** NDSEG fellowship, Boeing, AFOSR