Formal Methods: Two Perspectives

**Verification**

- System Model $\mathcal{M}$
- Specification $\varphi$

Proof whether $\mathcal{M} \models \varphi$

"satisfies"

**Synthesis**

- Specification $\varphi$

Synthesis Engine

System $\mathcal{M}$ such that $\mathcal{M} \models \varphi$

Vasu Raman (Caltech)
Reactive Synthesis

Most systems do not operate in isolation!

Given \( \varphi = \varphi_e \implies \varphi_s \)

- environment assumptions
- system guarantees

synthesize a strategy \( \mathcal{M} \) such that \( \mathcal{M} \models \varphi \)
Contributions

Specification
\( \varphi = \varphi_e \implies \varphi_s \)

Synthesis Engine

System \( \mathcal{M} \) such that \( \mathcal{M} \models \varphi \)

Signal Temporal Logic

MILP solver

Control input maximizing satisfaction

Vasu Raman (Caltech)
Highlights

• Counterexample-Guided Synthesis

• Model Predictive Control

• Semi-decision procedure for reactive synthesis from STL
Reactive Synthesis from Temporal Logic

• Robotics
  – Kress-Gazit, Fainekos and Pappas, ICRA 2007

• Autonomous Cars
  – Wongpiromsarn, Topcu and Murray, HSCC 2010

• Aircraft Electric Power Systems
  – Nuzzo et al, IEEE Access 2013

• Switched Systems
  – Liu, Ozay, Topcu and Murray, IEEE TAC 2013
LTL Synthesis for Hybrid Systems

1. Create discrete abstraction
2. Synthesize correct discrete controller
3. Continuously implement discrete solution
LTL Synthesis for Hybrid Systems

Create discrete abstraction → Synthesize correct discrete controller → Continuously implement discrete solution

Dynamical system → Labeled transition system

(Figure from BeltaIP04)

[AlurHLP00, BeltaH06, HabetsCS06, KaramanF09, Kress-GazitFP07, KloetzerB08, TabuadaP06, WongpiromsarnTM12,…]
LTL Synthesis for Hybrid Systems

Create discrete abstraction \rightarrow \text{Synthesize correct discrete controller} \rightarrow \text{Continuously implement discrete solution}

**BUT**

- Discrete abstraction is expensive
- LTL is inconvenient for specifying
  - properties of \textit{continuous signals}
  - temporal duration of events

Vasu Raman (Caltech)
Signal Temporal Logic (STL) [Maler and Nickovic 04]

• Continuous predicates:
  $\mu(x) > 0$
  No need to discretize the state space!

• Boolean operators:
  $\land$ (and) $\lor$ (or) $\neg$ (not) $\implies$ (implies)

• Temporal operations:
  $\varphi \mathcal{U}_{[a,b]} \psi$ (until)
  $\square_{[a,b]} \varphi$ (always)
  $\Diamond_{[a,b]} \varphi$ (eventually)

“At all times between a and b from now”
Quantitative Semantics for STL

[Donzé and Maler ‘10]

Robustness function $\rho^\varphi : \mathcal{X} \times \mathbb{N} \rightarrow \mathbb{R}$

$$(x, t) \models \varphi \equiv \rho^\varphi(x, t) > 0$$

$|x'_t - x_t| < \rho^\varphi(x, t)$

$\Rightarrow (x', t) \models \varphi$
Example: STL Quantitative Semantics

\[ \mu_1 \equiv x - 3 > 0 \quad \varphi = \Box [0,2] \mu_1 \]

- Define \( \rho^\varphi : \mathcal{X} \times \mathbb{N} \to \mathbb{R} \)
such that \((x, t) \models \varphi \equiv \rho^\varphi(x, t) > 0\)

  - \( \rho^{\mu_1}(x, 0) = x(0) - 3 \)
  - \( \rho^{\mu_1 \wedge \mu_2}(x, t) = \min(\rho^{\mu_1}, \rho^{\mu_2}) \)
  - \( \rho^\varphi(x, t) = \min_{t \in [0,2]} \rho^{\mu_1}(x, t) = \min_{t \in [0,2]} x(t) - 3 \)
Reactive Synthesis from STL

\[ \varphi = \varphi_e \quad \iff \quad \varphi_s \]

evironment \ assumptions

\[ \min \quad \max \]

\[ (x, t) \models \varphi \equiv \rho^{\varphi}(x, t) > 0 \]
Preliminaries

• Continuous time hybrid system

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

\[x \in \mathcal{X} \subseteq (\mathbb{R}^{n_c} \times \{0, 1\}^{m_l})\]

\[u \in U \subseteq (\mathbb{R}^{m_c} \times \{0, 1\}^{m_l})\]

\[w \in W \subseteq (\mathbb{R}^{e_c} \times \{0, 1\}^{e_l})\]

• Assume discrete-time approximation

\[x(t_{k+1}) = f_d(x(t_k), u(t_k), w(t_k))\]

\[t_{k+1} - t_k = \Delta t > 0\]

• A run

\[\xi(x_0, u, w) = (x_0 u_0 w_0)(x_1 u_1 w_1)(x_2 u_2 w_2)\ldots\]

\[\xi(x_0, u^N, w^N) = (x_0 u_0 w_0)(x_1 u_1 w_1)(x_2 u_2 w_2)\ldots(x_N u_N w_N)\]
Finite Run Parametrization

• Bounded-length $N$ based on formula

\[ x(t_{k+1}) = f_d(x(t_k), u(t_k), w(t_k)) \]

\[ N \geq \text{Bound}(\varphi) \]

e.g. \( \text{Bound}(\square_{[0,10]} \diamond_{[1,6]} \psi) = 16 \)

• Inspired by bounded model checking

[Biere et al. 99, Biere et al. 06, Clarke et al. 01]
STL Reactive Synthesis

Given:
Continuous-time system \( \dot{x} = f(x, u, w) \)
STL specification \( \varphi = \varphi_e \implies \varphi_s \)
Initial state \( x_0 \)
Cost function \( J \) on runs

Compute:
\[
\arg\min_{u^N} \max_{w^N \in \{w \in W^N | w \models \varphi_e \}} J(\xi(x_0, u^N, w^N)) \\
s.t. \quad \forall w^N \in W^N, \quad \xi(x_0, u^N, w^N) \models \varphi
\]
CEGIS: Counterexample-Guided Inductive Synthesis

[Solar-Lezama et al, ASPLOS 2006]
CEGIS: Counterexample-Guided Inductive Synthesis

1: procedure CEGIS(ξ, x₀, N, ϕ, J)
2: Let \( w₀ = (w₁⁰, w₂⁰, \ldots w_{N-1}⁰) \), s.t. \( w^N \models \phi_e \)
3: \( W_{cand} = \{ w⁰ \} \)
4: while True do
5: \( u⁰ \leftarrow \text{argmin}_{u \in U^N} \max_{w^0 \in W_{cand}} (J(ξ(x₀, u, w₀))) \)
   s.t. \( \forall w^0 \in W_{cand}, \xi(x₀, u, w₀) \models \phi_s \),
6: if \( u⁰ == \text{null} \) then
7: Return \text{INFEASIBLE}  
8: end if
9: \( w¹ \leftarrow \text{argmin}_{w \in W^N} \rho^\varphi(ξ(x₀, u⁰, w), 0) \)
   s.t. \( w¹ \models \phi_e \),
10: if \( \rho^\varphi(ξ(x₀, u⁰, w¹)) > 0 \) then
11: Return \( u⁰ \),
12: else
13: \( W_{cand} \leftarrow W_{cand} \cup \{ w¹ \} \)
14: end if
15: end while
16: end procedure
Algorithm 1

CEGIS Algorithm for Problem

1: procedure CEGIS(⇠, x₀, N, 'J)

2: Let w₀ = (w₀₁, w₀₂, ..., w₀N₁)

3: W_cand = {w₀}

4: while True do

5: u₀ = argmin_u∈U_max w₀ ∈ W_cand (J(⇠(x₀, u, w₀)))

6: s.t. ∀w₀ ∈ W_cand, ⇠(x₀, u, w₀) |= φ_s,

7: if u₀ == null then

8: Return INFEASIBLE

9: end if

10: w₁ = argmin_w∈W⇠(⇠(x₀, u₀, w₁), 0)

11: s.t. w₁ |= φ_e

12: if ⇠(⇠(x₀, u₀, w₁)) > 0 then

13: Return u₀

14: end if

15: end while

16: end procedure

OPENLOOP and FALSIFY

OpenLoop

\[ u^0 \leftarrow \arg\min_{u \in U^N} \max_{w^0 \in W_{\text{cand}}} (J(\xi(x_0, u, w^0))) \]

s.t. \( \forall w^0 \in W_{\text{cand}}, \xi(x_0, u, w^0) \models \varphi_s \),

FALSIFY

\[ w^1 \leftarrow \arg\min_{w \in W^N} \rho^\varphi(\xi(x_0, u^0, w), 0) \]

s.t. \( w^1 \models \varphi_e \)

Problem: find a bounded-time, open-loop sequence of inputs/disturbances
• satisfying an STL formula
• minimizing a cost
Open-Loop Synthesis

Problem: find a bounded-time sequence of inputs/disturbances
• satisfying an STL formula
• minimizing a cost

Solution: Encode everything (including the formula) as a Mixed Integer Linear Program

[Raman et al CDC 2014]
Example: STL Encoding

\((x, 0) \models \varphi \iff r^\varphi_0 > 0\)

\(\varphi = \Box[0,10] \lozenge [0,5] \((T > 20) \land (T < 30)\)\)

\(r_t^{(T>20)} = T - 20\)

\(r_t^{(T<30)} = 30 - T\)

\(r_t^{((T>20) \land (T<30))} = \min(r_t^{(T>20)}, r_t^{(T<30)})\)

\(r_t^{[0,5]((T>20) \land (T<30))} = \max_{i=0,...,5} (r_i^{((T>20) \land (T<30))})\)

\(r_t^{[0,10] \lozenge [0,5]((T>20) \land (T<30))} = \min_{i=0,...,10} (r_i^{[0,5]((T>20) \land (T<30))})\)
**CEGIS: Counterexample-Guided Inductive Synthesis**

1: `procedure CEGIS(ξ, x₀, N, φ, J)`
2: `Let w₀ = (w₁₀, w₂₀, ...wₙ₋₁₀), s.t. wᴺ |= φₑ`
3: `W_{cand} = {w₀}`
4: `while True do`
5: `u₀ ← argmin_{u ∈ Uᴺ} max_{w₀ ∈ W_{cand}} (J(ξ(x₀, u, w₀)))`
   `s.t. ∀w₀ ∈ W_{cand}, ξ(x₀, u, w₀) |= φₛ,`
6: `if u₀ == null then`
7: `Return INFEASIBLE`
8: `end if`
9: `w₁ ← argmin_{w ∈ Wᴺ} ρφ(ξ(x₀, u₀, w), 0)`
   `s.t. w¹ |= φₑ`
10: `if ρφ(ξ(x₀, u₀, w¹)) > 0 then`
11: `Return u₀`
12: `else`
13: `W_{cand} ← W_{cand} ∪ {w¹}`
14: `end if`
15: `end while`
16: `end procedure`
Receding Horizon Synthesis

• Open-loop: generate a finite signal

\[ x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_{N-1} \]

• Receding horizon (MPC): do this repeatedly

\[ x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_H \rightarrow x_{H+1} \rightarrow x_{H+2} \]

Vasu Raman (Caltech)
STL Receding Horizon Reactive Synthesis

Given:
Continuous-time system \( \dot{x} = f(x, u, w) \)
STL specification \( \varphi = \varphi_e \Rightarrow \varphi_s \)
Initial state \( x_0 \)
Cost function \( J \) on runs
Horizon \( H \)

Compute:
\[
\arg\min_{u^{H,k}} \max_{w^{H,k} \in \{w \in W^H \mid w \models \varphi_e\}} J(\xi(x_k, u^{H,k}, w^{H,k}))
\]
\[
\text{s.t.} \quad \forall w \in W^\omega, \quad \xi(x_0, u, w) \models \varphi
\]
Receding Horizon Synthesis

Time step

0

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_N \rightarrow x_{N+1} \rightarrow x_{2N-1}

1

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_N \rightarrow x_{N+1} \rightarrow x_{2N-1}

2

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_N \rightarrow x_{N+1} \rightarrow x_{2N-1}

3

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_N \rightarrow x_{N+1} \rightarrow x_{2N-1}

\ldots

N

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_N \rightarrow x_{N+1} \rightarrow x_{2N-1}

N+1

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_N \rightarrow x_{N+1} \rightarrow x_{2N-1} \rightarrow x_{2N}

N+2

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_N \rightarrow x_{N+1} \rightarrow x_{2N-1} \rightarrow x_{2N} \rightarrow x_{2N+1}

\ldots

known (constrained)
computation
past (forgotten)

“Transient” phase

“Stationary” phase

Vasu Raman (Caltech)
Case Study: HVAC Control

Minimize the input (air flow) subject to

“If the occupancy of a room is > 0, the temperature should be above the comfort level”

\[
\begin{align*}
\varphi_e &= \square(|w - w^{\text{ref}}| < 5) \\
\varphi_s &= \square((\text{occ}_t > 0) \implies (T_t > T_t^{\text{comfort}}))
\end{align*}
\]
Case Study: HVAC Control

![Graph showing HVAC control parameters](graph.png)
Case Study: HVAC Control
Case Study: HVAC Control
Case Study: Autonomous Driving

\[ \begin{bmatrix} x^{\text{ego}} \\ y^{\text{ego}} \\ v^{\text{ego}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^{\text{ego}} \\ y^{\text{ego}} \\ v^{\text{ego}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \]

\[ \begin{bmatrix} x^{\text{adv}} \\ y^{\text{adv}} \\ v^{\text{adv}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^{\text{adv}} \\ y^{\text{adv}} \\ v^{\text{adv}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} w \]

\[ \varphi_e = \Box (|w - w^{\text{ref}}| < 0.1) \]

\[ \varphi_s = \Box (|y^t - x^t| < 2) \Rightarrow \Box [0,2] (|v^t| < 0.1) \]

\[ J(\xi(x_t, u^H, w^H)) = \sum_{l=0}^{H-1} |v^t_l - 1| \]
Case Study: Autonomous Driving

![Graph showing distance and velocity of vehicles over time]

- **Distance of Ego vehicle from intersection**
- **Distance of Adversary vehicle from intersection**
- **Velocity of Ego Vehicle**
- **Velocity of Adversary Vehicle**
Recap + Discussion

- **Optimization-based reactive synthesis for Signal Temporal Logic**
  - No discrete abstraction
  - CEGIS scheme

- **Receding Horizon Control** for top-level unbounded “always”

- **Future Work:**
  - Other unbounded formulas
  - Synthesis for stochastic systems
Try it Out!

**BluSTL**: Controller Synthesis from Signal Temporal Logic Specifications *(presented at ARCH 2015)*

[https://github.com/BluSTL/BluSTL](https://github.com/BluSTL/BluSTL)  
(BSD License)

[See Zoolander, 2001]
Thanks!

Reactive Synthesis from Signal Temporal Logic Specifications

Vasu Raman, Alexandre Donzé, Dorsa Sadigh, Richard M. Murray, Sanjit A. Seshia

Contact: vasu@caltech.edu
https://github.com/BluSTL/BluSTL

HSCC
16 April 2015