Model Predictive Control from Signal Temporal Logic Specifications: A Case Study

Vasu Raman\(^1\)
Alexandre Donzé\(^2\) and Mehdi Maasoumy\(^2\)

\(^1\)California Institute of Technology
\(^2\)University of California at Berkeley

CyPhy
14 April 2014
Modern Cyber-Physical Systems

- Operate **autonomously**
- Fulfill **complex** requirements
- Easy to **specify** and **enforce** guarantees

Vasu Raman (Caltech)
Formal Methods: Two Perspectives

Verification

System Model $\mathcal{M}$

Specification $\varphi$

Proof Engine

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

“satisfies”

Synthesis

Specification $\varphi$

Synthesis Engine

System $\mathcal{M}$ such that $\mathcal{M} \models \varphi$
**Formal Methods: Two Perspectives**

**Verification**
- System Model $\mathcal{M}$
- Specification $\varphi$
- Proof Engine
- Proof whether $\mathcal{M} \models \varphi$

**Synthesis**
- Specification $\varphi$
- Synthesis Engine
- System $\mathcal{M}$ such that $\mathcal{M} \models \varphi$

Model Checkers, Theorem Provers, etc.
Formal Methods: Two Perspectives

Verification

System Model $\mathcal{M}$

Specification $\varphi$

Proof Engine

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

“satisfies”

Synthesis

Specification $\varphi$

Synthesis Engine

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

Model Checkers, Theorem Provers, etc.
Formal Methods: Two Perspectives

Verification

System Model $\mathcal{M}$

Specification $\varphi$

Proof Engine

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

“satisfies”

Synthesis

Specification $\varphi$

Synthesis Engine

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

Optimal control input

Model Checkers, Theorem Provers, etc.

Signal Temporal Logic

MILP solver

Formal Methods: Two Perspectives

Vasu Raman (Caltech)
Temporal Logic Synthesis for CPS
(Related Work)

• Robotics
  – Kress-Gazit, Fainekos and Pappas, ICRA 2007
  – Kloetzer and Belta, TAC 2008
  – Karaman and Frazzoli, CDC 2009
  – Bhatia, Kavraki and Vardi, ICRA 2010

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

• Aircraft Electric Power Systems
  – Nuzzo et al, IEEE Access 2013
Temporal Logic Synthesis for CPS (what is lacking?)

• Usually requires discrete abstraction
  – “If temperature falls below 20°C, get it back above 20°C in the next time step”

\[ \square(T_{\text{less than 20}} \implies \Diamond(\neg T_{\text{less than 20}})) \]
Temporal Logic Synthesis for CPS (what is lacking?)

- Temporal duration is often cumbersome
  - “Infinitely often visit $A$ and no more than 5 time steps later visit $B$”
  \[
  \Box\Diamond (A \land \bigcirc B \lor \bigcirc \bigcirc B \lor \bigcirc \bigcirc \bigcirc B \lor \bigcirc \bigcirc \bigcirc \bigcirc B \lor \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc B)
  \]
  - “All visits to $A$ and $B$ should be no more than 5.1s apart”
  \[
  \Box (A \implies \Diamond (\text{clock\_less\_than\_5.1} \land B))
  \]
Signal Temporal Logic (STL)

- Continuous predicates: $\mu(x) > 0$
- Boolean Operators: $\land, \lor, \implies, \neg$
- Bounded Temporal Operators:
  - $\square_{[a,b]} \varphi$
  - $\Diamond_{[a,b]} \varphi$
  - $\varphi_1 \mathcal{U}_{[a,b]} \varphi_2$

  $\varphi$ holds at all $t \in [a, b]$  
  $\varphi$ holds at some $t \in [a, b]$

- Synthesis undecidable for dense time
  - We’ll restrict to discrete time (but continuous systems)
Signal Temporal Logic (STL)

Syntax

\[ \varphi ::= \mu \mid \neg \mu \mid \varphi \land \psi \mid \varphi \lor \psi \mid \Box_{[a,b]} \psi \mid \varphi \mathcal{U}_{[a,b]} \psi \]

Semantics

\[(x, t) \models \mu \quad \Leftrightarrow \quad \mu(x(t)) > 0\]

\[(x, t) \models \neg \mu \quad \Leftrightarrow \quad \neg((x, t) \models \mu)\]

\[(x, t) \models \varphi \land \psi \quad \Leftrightarrow \quad (x, t) \models \varphi \land (x, t) \models \psi\]

\[(x, t) \models \varphi \lor \psi \quad \Leftrightarrow \quad (x, t) \models \varphi \lor (x, t) \models \psi\]

\[(x, t) \models \Box_{[a,b]} \varphi \quad \Leftrightarrow \quad \forall t' \in [t + a, t + b], (x, t') \models \varphi\]

\[(x, t) \models \varphi \mathcal{U}_{[a,b]} \psi \quad \Leftrightarrow \quad \exists t' \in [t + a, t + b] \text{ s.t. } (x, t') \models \psi \land \forall t'' \in [t, t'], (x, t'') \models \varphi.\]
Examples

• If temperature falls below 20°C, get it back above 20°C within 5 time steps
  \( \Box (T_{\text{less than 20}} \implies \lozenge (\neg T_{\text{less than 20}})) \)

• Infinitely often visit A and no more than five time steps later visit B
  \( \Box \Diamond (A \land \lozenge B \lor \lozenge \lozenge B \lor \lozenge \lozenge \lozenge \lozenge B \lor \lozenge \lozenge \lozenge \lozenge \lozenge \lozenge \lozenge B) \)

• All visits to A and B should be no more than 5.1 seconds steps apart
  \( \Box (A \implies \Diamond (\text{clock} \text{ less than 5.1} \land B)) \)
Examples

• If temperature falls below 20°C, get it back above 20°C within 5 time steps
  \( \square(T < 20 \iff \lozenge_{[0,5]}(T > 20)) \)

• Infinitely often visit A and no more than five time steps later visit B
  \( \square \lozenge (A \land \lozenge_{[0,5]} B) \)

• All visits to A and B should be no more than 5.1 seconds steps apart
  \( \square (A \implies \lozenge_{[0,5.1]} B) \)

Vasu Raman (Caltech)
Optimal Control Synthesis from STL

Given:
Discrete time continuous system \( x_{t+1} = f(x_t, u_t) \)
STL specification \( \varphi \)
Initial state \( x_0 \)
Cost function \( J \) on runs of the system

Compute:
\[
\text{arg min}_{u} \quad J(x(x_0, u), u) \\
\text{s.t.} \quad x(x_0, u) \models \varphi
\]
Model Predictive Control from STL

Given:
Discrete time continuous system $x_{t+1} = f(x_t, u_t)$
STL specification $\varphi$
Initial state $x_0$
Cost function $J$ on runs of the system
Horizon $H$

Compute:
$$\arg \min_{u_t^H} J(x^H(x_t, u_t^H), u_t^H)$$
subject to $x(x_0, u) = \varphi$,
Finite Trajectory Parametrization

• Lasso-shaped parametrization for infinite executions

• Common approach in Bounded Model Checking
STL Synthesis for Control (Overview)

- System Dynamics
- STL Specification $\varphi$
- Finite trajectory parametrization

Mixed Integer Linear Program (constraints + objective)

MILP solver

Optimal control input enforcing $\varphi$

Vasu Raman (Caltech)

Raman et al, in submission (2014)
STL Synthesis for Control (Overview)

System Dynamics \rightarrow \text{STL Specification } \varphi \rightarrow \text{Finite trajectory parametrization}

\text{Mixed Integer Linear Program (constraints + objective)} \rightarrow \text{MILP solver} \rightarrow \text{Optimal control input enforcing } \varphi

Vasu Raman (Caltech)
STL to MILP constraints

Given a formula $\psi$ with subformulas denoted by $\varphi$

Introduce $z^\varphi_t$

Constrained such that $z^\varphi_t = 1 \iff (x, t) \models \varphi$

Enforce $z^\psi_0 = 1$

Recursively generate the MILP constraints corresponding to $z^\psi_0$. 
## STL to MILP constraints

Given a formula $\psi$ with subformulas denoted by $\varphi$

### Predicates

$\mu(x_t) \leq M_t z^\mu_t + \epsilon_t$

$-\mu(x_t) \leq M_t (1 - z^\mu_t) - \epsilon_t$

### Conjunction

$\psi = \land_{i=1}^m \varphi_i$

$z^\psi_t \leq z^\varphi_i \ , \ i = 1, \ldots, m,$

$z^\psi_t \geq 1 - m + \sum_{i=1}^m z^\varphi_i$

### Disjunction

$\psi = \lor_{i=1}^m \varphi_i$

$z^\psi_t \geq z^\varphi_i \ , \ i = 1, \ldots, m,$

$z^\psi_t \leq \sum_{i=1}^m z^\varphi_i$

---

Vasu Raman (Caltech)
STL to MILP constraints

Given a formula $\psi$ with subformulas denoted by $\varphi$

Always
$\psi = \square[a,b] \varphi$

Eventually
$\psi = \Diamond[a,b] \varphi$

Until
$\psi = \varphi_1 \mathcal{U}[a,b] \varphi_2$

$$a_t^N = \min(t + a, N), \quad b_t^N = \min(t + b, N)$$

$$z_t^\psi = \bigvee_{i=a_t^N}^N z_i^\varphi \land (\bigvee_{j=1}^N I_j \land \bigwedge_{i=j+\hat{a}_t^N}^{j+b_t^N} z_i^\varphi)$$

$$z_t^\psi = \bigwedge_{i=a_t^N}^N z_i^\varphi \land (\bigvee_{j=1}^N I_j \land \bigwedge_{i=j+\hat{a}_t^N}^{j+b_t^N} z_i^\varphi)$$

$$\varphi_1 \mathcal{U}[a,b] \varphi_2 = \square[0,a] \varphi_1 \land \Diamond[a,b] \varphi_2 \land \Diamond[a,a](\varphi_1 \mathcal{U} \varphi_2)$$
Quantitative Semantics for STL

- How much can we **vary the signal** and still satisfy $\varphi$?
- Robustness function $\rho^\varphi : \mathcal{X} \times \mathbb{N} \to \mathbb{R}$

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

\[
\begin{align*}
\rho^\mu(x, t) &= \mu(x(t)) \\
\rho^{-\mu}(x, t) &= -\mu(x(t)) \\
\rho^\varphi \land^\psi(x, t) &= \min(\rho^\varphi(x, t), \rho^\psi(x, t)) \\
\rho^\varphi \lor^\psi(x, t) &= \max(\rho^\varphi(x, t), \rho^\psi(x, t)) \\
\rho^{\square_{[a, b]} \varphi}(x, t) &= \min_{t' \in [t+a, t+b]} \rho^\varphi(x, t') \\
\rho^\varphi \cup_{[a, b]}^\psi(x, t) &= \max_{t' \in [t+a, t+b]} \left( \min(\rho^\psi(x, t'), \min_{t'' \in [t, t']} \rho^\varphi(x, t'')) \right)
\end{align*}
\]
Quantitative Semantics for STL

• How much can we **vary the signal** and still satisfy $\varphi$?
• Robustness function $\rho^\varphi : \mathcal{X} \times \mathbb{N} \to \mathbb{R}$
  $$(x, t) \models \varphi \equiv \rho^\varphi(x, t) > 0$$

• Examples: $\mu_1 \equiv x - 3 > 0 \quad \varphi = [0,2] \mu_1$

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

Given:
Discrete time continuous system \( x_{t+1} = f(x_t, u_t) \)
STL specification \( \varphi \)
Initial state \( x_0 \)
Robustness function \( \rho^\varphi : X \times \mathbb{N} \rightarrow \mathbb{R} \)

Compute:
\[
\arg \max_u \rho^\varphi(x_0, 0) \\
s.t. \ x(x_0, u) \models \varphi
\]
# STL to MILP constraints

Given a formula $\psi$ with subformulas denoted by $\varphi$

<table>
<thead>
<tr>
<th>Introduce</th>
<th>Boolean encoding</th>
<th>Robustness encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constrained such that</td>
<td>$z^\varphi_t = 1 \iff (x, t) \models \varphi$</td>
<td>$r^\varphi_t &gt; 0 \iff (x, t) \models \varphi$</td>
</tr>
<tr>
<td>Enforce</td>
<td>$z^\psi_0 = 1$</td>
<td>$r^\psi_0 &gt; 0$</td>
</tr>
</tbody>
</table>

Recursively generate the MILP constraints corresponding to $z^\psi_0$ or $r^\psi_0$

Vasu Raman (Caltech)
STL Synthesis for Control (Overview)

System Dynamics → STL Specification $\varphi$ → Finite trajectory parametrization

Mixed Integer Linear Program (constraints + objective)

MILP solver

Optimal control input enforcing $\varphi$

Two encodings:
- Boolean
- Quantitative
STL Synthesis for Control (Overview)

- System Dynamics
- STL Specification \( \varphi \)
- Finite trajectory parametrization

Mixed Integer Linear Program (constraints + objective)

- MILP solver

Optimal control input enforcing \( \varphi \)

Two encodings:
- Boolean
- Quantitative

This is open loop... what about model predictive control?
MPC/Receding Horizon Control
(for **bounded** formulas)

- Pick $H$ based on $\varphi$
  - conservative bound on trajectory length to decide satisfiability
  - e.g. for $\Box [0,10] \Diamond [1,6] \varphi$ use $H \geq 10 + 6 = 16$

- Open-loop synthesis at each time step
  - STL constraints apply on the length-$H$ prefix

- Store history of states and inputs
  - ensures $\varphi$ is satisfied over the length-$H$ prefix

- Extends to certain unbounded formulas
  - e.g. $\varphi = \Box (\varphi_{MPC})$ for bounded $\varphi_{MPC}$.
Example: Grid regulation

Controlling ancillary service power flow for grid frequency regulation

Minimize control input

subject to

“If the Area Control Error (ACE) increases above 0.01, it will decrease below 0.01 within $\tau$ time steps”

$$\varphi_t = \neg(|\text{ACE}^1| < .01)) \Rightarrow (\diamond_{[0, \tau]}(|\text{ACE}^1| < .01) \wedge \neg(|\text{ACE}^2| < .01)) \Rightarrow (\diamond_{[0, \tau]}(|\text{ACE}^2| < .01)$$

Vasu Raman (Caltech)
Example: Grid regulation

\[
\min_{U_{\text{anc}}[k]} J(\text{ACE}, U_{\text{anc}}) + \| x[k + H] - x_{\text{ref}} \|_Q
\]

s.t.

\[
x[k + j + 1] = Ax[k + j] + B_2 u_{\text{anc}}[k + j] + Ed[k + j]
\]

\[
\underline{u_{\text{anc}}} \leq u_{\text{anc}}[k + j] \leq \bar{u}_{\text{anc}}
\]

\[
| u_{\text{anc}}[k + j + 1] - u_{\text{anc}}[k + j] | \leq \lambda
\]

\[
x[k + H] \in \mathcal{X}[H]
\]

\[
x[k] \models \varphi
\]

\[
J(\text{ACE}, U_{\text{anc}}) = \| U_{\text{anc}} \|_{\ell_2} = \sum_{i=1}^{2} \sum_{j=0}^{H-1} (U_{\text{anc}}^i[k + j])^2
\]

\[
\varphi = \Box(\varphi_t)
\]

\[
\varphi_t = \neg(|\text{ACE}^1| < .01) \Rightarrow (\Box_{[0, \tau]}(|\text{ACE}^1| < .01) \land \neg(|\text{ACE}^2| < .01) \Rightarrow (\Box_{[0, \tau]}(|\text{ACE}^2| < .01)
\]

Vasu Raman (Caltech)

Raman et al, in submission (2014)
Example: Grid regulation

Vasu Raman (Caltech)

Raman et al, in submission (2014)
Example: HVAC system

Minimize the input (total air flow) subject to

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

\[ \varphi = \square_{[0,H]}((\text{occ}_t > 0) \Rightarrow (T_t > T_{t}^{\text{conf}})) \]
Example: HVAC system

\[ \varphi = \min_{\vec{u}_t} \sum_{k=0}^{H-1} \|u_{t+l}\| \quad \text{s.t.} \]

\[ x_{t+k+1} = f(x_{t+k}, u_{t+k}, d_{t+k}), \]

\[ x_t \models \varphi \]

\[ u_{t+k} \in \mathcal{U}_{t+k}, \quad k = 0, \ldots, H - 1 \]
Future Work

• Receding Horizon framework for unbounded STL properties
  – ties to online monitoring of STL properties
  – formalize connection with reactive synthesis

• Contract-based framework for specifying and designing components (e.g. of the smart-grid) and their interactions
Thank You!

Model Predictive Control from Signal Temporal Logic Specifications: A Case Study

Vasu Raman\textsuperscript{1}
Alexandre Donzé\textsuperscript{2} and Mehdi Maasoumy\textsuperscript{2}

\textsuperscript{1}California Institute of Technology
\textsuperscript{2}University of California at Berkeley

Email: vasu@caltech.edu