Lecture 8
Receding Horizon Temporal Logic Planning & Compositional Protocol Synthesis

Ufuk Topcu
Nok Wongpiromsarn    Richard M. Murray

EECI, 18 May 2012

Outline:
• Receding horizon temporal logic planning (RHTLP)
  • Basic idea & main result
  • Discussion of the key details of implementation
  • Hierarchical control architecture
  • Autonomous driving examples
• Compositional control protocol synthesis and its application to smart camera networks and resource allocation
Problem: Design control protocols, that...

- Handle mixture of discrete and continuous dynamics
- Account for both high-level specs and low-level constraints
- Reactively respond to changes in environment,

... with “correctness certificates.”

\[
\left[ (\varphi_{init} \land \varphi_{env}) \rightarrow (\varphi_{safety} \land \varphi_{goal}) \right]
\]

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

\[
g(x, u) \geq 0
\]
**Preview**

**Multi-layer approach**

- Use optimal trajectory generation to create a discrete abstraction that captures the dynamics at a simplified level
- Reactive planner based on GR(1) synthesis (possibly RHC)
- High level planner sends specifications to reactive planner
- Online versus offline decisions at each level

**Alice’s navigation stack**

- Mission Planner
  - Traffic Planner
  - Path Planner
  - Vehicle Actuation

**Different views**

- “long-horizon specification”
- “short-horizon specification”
- continuous dynamics & constraints

**Multi-scale models**

\[ \mathcal{W}_0 \prec \ldots \prec \mathcal{W}_{L-1} \prec \mathcal{W}_L \]

\[
\min \int_{t_0}^{T} L(x, u) dt \\
\text{s.t. } \dot{x} = f(x, u) \\
g(x, u) \leq 0
\]

**Hierarchical control architecture**

- Goal Generator
- Trajectory Planner
- Continuous Controller
Each of these cells may be occupied by an obstacle.

The vehicle can be in any of these cells.

\((2L)\left(2^{2L}\right)\) possible states!
Receding Horizon Control

\[
\min_{u[t, t+T]} \int_t^{t+T} C(x(\tau), u(\tau)) d\tau + V(x(t+T))
\]

subject to:
\[
\dot{x} = f(x, u), \quad x(t) \text{ given} \\
g(x, u) \leq 0
\]

- Reduces the computational cost by solving smaller problems.
- Real-time (re)computation improves robustness.
Receding Horizon Control

- If not implemented properly, global properties, e.g., stability, are not guaranteed.
- Increasing $T$ helps for stability at the expense of increased computational cost.

- If the terminal cost is chosen as a control Lyapunov function, i.e., $V$ is (locally) positive definite and satisfy (for some $r>0$)
  \[
  \min_u (\dot{V} + C(x, u)) < 0, \quad \forall x \in \{x : V(x) \leq r^2\}
  \]
  then stability is guaranteed.

- Alternative (related) approach, imposed contractiveness constraints in short-horizon problems.
Receding Horizon for LTL Synthesis

Global (long-horizon) specification:

\[(\varphi_{\text{init}} \land \varphi_{\text{env}}) \rightarrow (\varphi_{\text{safety}} \land \varphi_{\text{goal}})\]

Basic idea:

- Partition the state space into a partially ordered set \(\{W_j\}, \preceq_{\varphi} \)
- Goal-induced partial order

Short-horizon specification: For each \(i\),

\[
\left(\nu \in W_i\right) \land (\varphi_{\text{env}} \land \varphi) \rightarrow (\Box \varphi \land \varphi_{\text{safety}} \land \Diamond (\nu \in F_i(W_i)))
\]

Plan from the current cell on

Receding horizon invariant: rules out “corner” cases

Get closer to goal rather than reaching. \(F\): “horizon” length

Theorem: Receding horizon implementation of the short-horizon strategies ensures the correctness of the global specification.

Trade-offs:

- computational cost vs. horizon length vs. strength of invariant vs. conservatism

[TAC’11(submitted), HSCC’10]
How to come up with a partial order, $\mathcal{F}$ and $\Phi$?

• In general, problem-dependent and requires user guidance.
• Partial automation is possible (discussed later).
• Partial order: “measure of closeness” to the goal, i.e., to the states satisfying.
• The map $\mathcal{F}$ determines the “horizon length.

• The invariant $\Phi$ (in this example) rules out the states that render the short horizon problems unrealizable.
• In the example above, it is the conjunction of the following propositional formulas on the initial states for each subproblem:
  • no collision in the initial state
  • vehicle cannot be in the left lane unless there is an obstacle in the right lane in the initial state
  • vehicle is able to progress from the initial state
Navigation of point-mass omnidirectional vehicle

Nondimensionalized dynamics:
\[
\begin{align*}
\ddot{x} + \dot{x} &= q_x(t) \\
\ddot{y} + \dot{y} &= q_y(t) \\
\ddot{\theta} + \frac{2mL^2}{J} \dot{\theta} &= q_\theta
\end{align*}
\]

Conservative bounds on control authority to decouple the dynamics:
\[
\begin{align*}
|q_x(t)|, |q_y(t)| &\leq \sqrt{0.5} \\
|q_\theta(t)| &\leq 1
\end{align*}
\]

Reasons for the non-intuitive trajectories:
- Synthesis: feasibility rather than “optimality.”
- Specifications are not rich enough.

Partition (in two consecutive cells):
Example: Navigation In Urban-Like Environment

Dynamics: \[ \dot{x}(t) = u_x(t) + d_x(t), \quad \dot{y}(t) = u_y(t) + d_y(t) \]

Actuation limits: \[ u_x(t), u_y(t) \in [-1, 1], \quad \forall t \geq 0 \]

Disturbances: \[ d_x(t), d_y(t) \in [-.1,.1], \quad \forall t \geq 0 \]

Traffic rules:
• No collision
• Stay in right lane unless blocked by obstacle
• Proceed through intersection only when clear

Environment assumptions:
• Obstacle may not block a road
• Obstacle is detected before it gets too close
• Limited sensing range (2 cells ahead)
• Obstacle does not disappear when the vehicle is in its vicinity
• Obstacles don’t span more than certain # of consecutive cells in the middle of the road
• Each intersection is clear infinitely often
• Cells marked by star and adjacent cells are not occupied by obstacle infinitely often

Goals: Visit the cells with *'s infinitely often.
Navigation In Urban-Like Environment

Setup:
- **Dynamics**: Fully actuated with actuation limits and bounded disturbances
- **Specifications**:
  - Traffic rules
  - Assumptions on obstacles, sensing range, intersections,...
- **Goals**: Visit the two stars infinitely often

Results:
- **Without receding horizon**: $1e87$ states (hence, not solvable)
- **Receding horizon**:
  - Partial order: From the top layer of the control hierarchy
  - Horizon length $= 2$  \( (\mathcal{F}(\mathcal{W}^i_j) = \mathcal{W}^i_{j-2}) \)
  - Invariant: Not surrounded by obstacles. If started in left lane, obstacle in right lane.
  - $1e4$ states in the automaton.
  - $\sim 1.5$ sec for each short-horizon problem
  - Milliseconds for partial order generation
What is $\Phi$?

- A propositional formula (that we call receding horizon invariant).
- Used to exclude the initial states that render synthesis infeasible, e.g., states from which collision is unavoidable.

**Short-horizon specification:**

$$(((\nu \in W_i) \land \Phi \land \varphi_{env}) \rightarrow (\Box \Phi \land \varphi_{safety} \land \Diamond (\nu \in F_i(W_i))))$$

Given partial order and $F$, computation of the invariant can be automated:

- Check realizability
- If realizable, done.
- If not, collect violating initiation conditions. Negate them and put in $\Phi$.
- Repeat until all subproblems or all possible states are excluded (in the latter case, either the global problem is infeasible or RHTLP with given partial order and $F$ is inconclusive.)
Generalization to multiple “goals”

General form of LTL specifications considered in reactive control protocol synthesis:

\[
\left( \psi_{init} \land \Box \psi_e \land \left( \bigwedge_{i\in I_f} \Box \Diamond \psi_{f,i} \right) \right) \rightarrow \left( \left( \bigwedge_{i\in I_s} \Box \psi_{s,i} \right) \land \left( \bigwedge_{i\in I_g} \Box \Diamond \psi_{g,i} \right) \right)
\]

Each partial order covers the discrete (system) state space. For each \( \nu \in \mathcal{W}_0^{i,j} \), one can find a cell in the “proceeding” partial order that \( \nu \) belongs to.

**Strategy:** While in \( \mathcal{W}_j^i \) implement (in a receding horizon fashion) the controller that realizes

\[
\left( (\nu \in \mathcal{W}_j^i) \land \Phi \land \Box \psi_e \land \bigwedge_{k\in I_f} \Box \Diamond \psi_{f,k} \right)
\]

\[\rightarrow \left( \bigwedge_{k\in I_s} \Box \psi_{s,k} \land \Box \Diamond (\nu \in \mathcal{F}^i(\mathcal{W}_j^i)) \land \Box \Phi \right)\]
Computational complexity & completeness

For Generalized Reactivity [1] formulas, the computation time of synthesis is $O(mn|\Sigma|^3)$, where $|\Sigma|$ is the number of discrete states.

Receding horizon implementation...

- reduces the computational complexity by restricting the state space considered in each subproblem; and
- is not complete, i.e., the global problem may be solvable but the choice of $\{W_j\}$, the partial order, the maps $F_i$, and $\Phi$ may not lead to a solution.

Choose $F_i$ to give “longer horizon”:

- Subproblems in RHTLP are more likely to be realizable.
- Computational cost is higher.

E.g., for urban-like driving example is infeasible with horizon length of one.

Global synthesis problem

$$(\varphi_{\text{init}} \land \varphi_{\text{env}}) \rightarrow (\varphi_{\text{safety}} \land \varphi_{\text{goal}})$$

Subproblems in RHTLP

$$((v \in W_i) \land \Phi \land \varphi_{\text{end}}) \rightarrow (\varphi_{\text{safety}} \land \lozenge(v \in F_i(W_i) \land \Box \Phi))$$
**ShortHorizonProb**: a class for defining a short horizon problem

- **computeLocalPhi()**: compute $\phi$ that makes this short horizon problem realizable.

**RHTLPProb**: a class for defining a receding horizon temporal logic planning problem

- Contains a collection of short-horizon problems
- Useful methods
  - **computePhi()**: compute $\phi$ for this RHTLP problem if one exists.
  - **validate()**: validate that the sufficient conditions for applying RHTLP hold
Hierarchical control structure

models of varying fidelity

\[ \mathcal{W}_L \cup \mathcal{W}_{L-1} \cup \ldots \cup \mathcal{W}_0 \]

Abstraction procedure and bisimulations relate models of different fidelity level.

\[
\begin{align*}
\ddot{x} + \dot{x} &= q_x(t) \\
\ddot{y} + \dot{y} &= q_y(t) \\
\ddot{\theta} + \frac{2mL^2}{J} \dot{\theta} &= q_\theta
\end{align*}
\]

\(|q_x(t)|, |q_y(t)| \leq \sqrt{0.5}
\]

\(|q_\theta(t)| \leq 1\)
Decompositions in the state space

Decompositions induced by ...

<table>
<thead>
<tr>
<th>receding horizon</th>
<th>goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>distributed synthesis</td>
<td>underlying network</td>
</tr>
</tbody>
</table>
**Goal:** synthesize control protocols for PTZ to ensure that one high resolution image of each target is captured at least once.
Synthesis of protocols for active surveillance

System:
- region of view of PTZs
- governed by finite state automata

Additional requirement:
- Zoom-in the corner cells infinitely often.

Environment specifications:
- At most N targets at a time.
- Every target remains at least T time steps and eventually leaves.
- Can only enter/exit through doors.
- Can only move to neighbors.
Centralized vs. decentralized control architecture

How to design control protocols that can be
• synthesized
• implemented
in a decentralized way?

What information exchange & interface models are needed?
**Compositional Synthesis**

**Goal:** Find control protocols for PTZ-1 & PTZ-2 so that

\[ \varphi_e \rightarrow \varphi_s \] holds.

**Simple & not very useful composition:**

<table>
<thead>
<tr>
<th>Statement</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any execution of the env’t, satisfying ( \varphi_e ), also satisfies ( \varphi_{e_1} \land \varphi_{e_2} )</td>
<td>( \varphi_{e_1} \land \varphi_{e_2} )</td>
</tr>
<tr>
<td>Any execution of the system, satisfying ( \varphi_{s_1} \land \varphi_{s_2} ), also satisfies ( \varphi_s )</td>
<td>( \varphi_{s_1} \land \varphi_{s_2} )</td>
</tr>
<tr>
<td>No common controlled variables in ( \varphi_{s_1} ) and ( \varphi_{s_2} )</td>
<td></td>
</tr>
<tr>
<td>There exist control protocols that realize ( \varphi_{e_1} \rightarrow \varphi_{s_1} ) &amp; ( \varphi_{e_2} \rightarrow \varphi_{s_2} )</td>
<td></td>
</tr>
</tbody>
</table>

\[ \varphi_e \rightarrow \varphi_s \] is realized.
Central

Compositional

\( e, \varphi_e \overset{\text{Sys}}{\longrightarrow} s, \varphi_s \)

\( e_1, \varphi_{e_1} \overset{\text{Sys}_1}{\longrightarrow} s_1, \varphi_{s_1} \)

\( e_2, \varphi_{e_2} \overset{\text{Sys}_2}{\longrightarrow} s_2, \varphi_{s_2} \)

\( P_1 \)

\( P_2 \)

\( \phi_1 \)

\( \phi_2 \)

\( \phi'_1 \)

\( \phi'_2 \)

\( \wedge \)

\( \Rightarrow \)

\( \Rightarrow \)

Synthesis of Embedded Control Software
(Refined) Compositional Synthesis

As before:

Any execution of the env’t, satisfying $\varphi_e$, also satisfies $\varphi_{e1} \land \varphi_{e2}$

Any execution of the system, satisfying $\varphi_{s1} \land \varphi_{s2}$, also satisfies $\varphi_s$

No common controlled variables in $\varphi_{s1}$ and $\varphi_{s2}$

Refined interfaces:

There exist control protocols that realize

$$\left(\varphi'_2 \land \varphi_{e1}\right) \rightarrow \left(\varphi_{s1} \land \varphi_1\right) \quad \& \quad \left(\varphi'_1 \land \varphi_{e2}\right) \rightarrow \left(\varphi_{s2} \land \varphi_2\right)$$

For soundness and to avoid circularity:

$$\Box (\varphi_i \rightarrow \diamond \varphi'_i) \quad \text{for } i = 1, 2$$

$\varphi_e \rightarrow \varphi_s$ is realized.
Application to a (very simple) smart camera network

IsZoomed & StepsInZone

$\phi_1$ and $\phi'_1$

limit the number of unzoomed targets entering zone 2 from zone 1
Case Study: Synthesis of Protocols for Electric Power Management

**Multiple criticality levels:**
- flight controllers
- active de-icing
- environmental control

**Environment variables:**
- wind gust (w)
- outside temperature (T)

**Controlled variables:**
- altitude
- power supply to different components

For environment & control variables, use crude discretization over their respective ranges. For example, 
\[ T \in \{ \text{low, low-medium, medium-high, high} \} \]
representing the range of \([-22^\circ F, 32^\circ F]\)

**Dependent (state) variables:**
- level of ice accumulation
- state-of-charge of the batteries
- cabin pressure level

Modeling & The Dependent Variables

Use models based on finite transitions systems from a combination of empirical data and first principles.

<table>
<thead>
<tr>
<th>Icing Level</th>
<th>Airspeed Reduction</th>
<th>Power Increase to Regain Airspeed</th>
<th>Climb-rate Reduction</th>
<th>Reduction in Control Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>&lt; 10 knots</td>
<td>&lt; 10%</td>
<td>&lt; 10%</td>
<td>No effect</td>
</tr>
<tr>
<td>Light</td>
<td>10 – 19 knots</td>
<td>10 – 19%</td>
<td>10 – 19%</td>
<td>No effect</td>
</tr>
<tr>
<td>Moderate</td>
<td>20 – 39 knots</td>
<td>20 – 39%</td>
<td>≥ 20%</td>
<td>Slow or overly sensitive response</td>
</tr>
<tr>
<td>Severe</td>
<td>≥ 40 knots</td>
<td>Unable</td>
<td>Unable</td>
<td>Limited or no response</td>
</tr>
</tbody>
</table>

State-of-charge evolves with:

\[ b[t + 1] = \min \{ B, b[t] + \bar{P} - p_f[t] - p_d[t] - p_e[t] \} \]

Transitions model the gap between requested and supplied power for each functionality.
Sample Specifications

Resource constraint: \( \Box(p_f + p_d + p_e \leq \bar{P} + b) \)
Prioritization: \( \Box(p_f \geq r_f) \)
\( \Box(p_f = \text{high} \land p_d = \text{high} \Rightarrow p_e = \text{low}) \)

Safety: Altitude cannot change too much between to consecutive instants, e.g.,
\( \Box(h = \text{low} \Rightarrow (\Diamond h \neq \text{medium-high} \land \Diamond h \neq \text{high})) \)
Ice accumulation limits allowable altitude change, e.g.,
\( \Box(a = \text{severe} \Rightarrow \Diamond h = h) \)
Ice accumulation cannot be severe: \( \Box(a \neq \text{severe}) \)

Performance: Cabin pressure does not exceed the level at 8000 ft.
Always go back to the desirable altitude: \( \Box \Diamond (h = \text{high}) \)

Assumptions: Wind gusts cannot be severe too many consecutive steps.
\( \Box(n_w \geq N_w \Rightarrow \Diamond(w \neq \text{severe}) \)
No abrupt change in outside temperature, e.g.,
\( \Box(T = \text{medium-low} \Rightarrow \Diamond T \neq \text{high}) \)

Notation may not be fully explained. Ask, if confused!!!
Dynamic power allocation allows reductions in peak power (i.e., generator weight) requirements.

Formulate as a temporal logic, reactive planning problem

\[
(\varphi_{\text{environment}} \land \varphi_{\text{initial}} \land \varphi_{\text{criticality}}) \downarrow (\varphi_{\text{performance}} \land \varphi_{\text{safety}})
\]

\[
N_w = 2, B = 3 \\
P = 5 \\
r_f, r_d \in \{0, 1, 2, 3\} \\
r_e \in \{0, 1, 2\}
\]
Conventional vs. Boeing 787 Electric Power Network Structure

pre-787

- E/E Bay
- 115 Vac Feeder
- External Power 115 Vac
- Generator 1 x 120 kVA
- 115 Vac or 28 Vdc Wire
- Loads
- APU Generator

787: distributed

- External Power 2 x 115 Vac, 90 kVA
- Forward E/E Bay
- 230 Vac Feeder
- Generator 2 x 250 kVA
- Aft E/E Bay
- External Power 2 x 115 Vac, 90 kVA
- Remote Power Distribution Unit
- APU Generator 2 x 225 kVA
- Loads
Distributed resource allocation

Controlled variables:
- Power supplies to each function
- Altitude

Environment variables:
- Wind gusts
- Outside temperature
- Generator health status

Dependent variables:
- Level of ice accumulation
- State-of-charge of the battery
- Cabin pressure & temperature

Interface refinements

\[
\psi_{12} = \square \Diamond (h = 1)
\]

\[
\psi_{21} = \square [\neg H_1 \rightarrow (p_{21} = 1)] \land (H_1 \rightarrow (p_{21} = 0))
\]
Compositional Synthesis of Distributed Protocols

\[ \varphi_{e_1} \rightarrow \varphi_{s_1} \]

\[ \varphi_{e_3} \rightarrow \varphi_{s_3} \]

\[ \Lambda i \varphi_{e_i} \rightarrow \varphi_e \rightarrow \varphi_s \rightarrow \Lambda i \varphi_{s_i} \]

“weaker”
environment assumptions

“stronger”
system requirements

Extra (mild) technical conditions: No common controlled variables & loops are well-posed.

Theorem: \( \varphi_e \rightarrow \varphi_s \) is realizable if every \( \varphi_{e_i} \rightarrow \varphi_{s_i} \) is realizable.

Contracts formalize the coupling and information exchange between subsystems.

Trade-offs:

- conservatism vs.
- expressiveness of contracts vs.
- need for coordination & computational cost