Decomposition of planning for multi-agent systems under LTL specifications

Jana Tumova and Dimos V. Dimarogonas
KTH Royal Institute of Technology

December 14, 2015
General Motivation and Approach

Correct-by-design mission and motion planning for multi-agent systems under complex high-level specifications
General Motivation and Approach

Correct-by-design mission and motion planning for multi-agent systems under complex high-level specifications

Approach

- Linear Temporal Logic (LTL) as a specification language
  - Expressive power
  - Rigorousness
  - Some resemblance to natural language

- Hierarchical approach
  1. Abstraction of a dynamical system
  2. Discrete planning
  3. Implementation
Challenge: Computational Feasibility of Step 2

Correct-by-design mission and motion planning for multi-agent systems under complex high-level specifications
Challenge: Computational Feasibility of Step 2

Correct-by-design mission and motion planning for multi-agent systems under complex high-level specifications

Example
A straightforward extension of the automata-based approach
Related Work: A Top-Down Approach


- Split the team formula into independent ones
- Works for a subclass of LTL only
Aim: A Bottom-Up Approach

- A team $\mathcal{N} = \{1, \ldots, N\}$ of agents
  - Abstraction of action and motion capabilities: a finite discrete transition system $\mathcal{T}_i$
  - Synchronization capabilities
- High-level behavior specification for each agent
  - Independent motion specification over the states
  - Dependent task specification over the inputs/actions
- Efficiently synthesize plans fulfilling the tasks
  - A satisfying trace of each $\mathcal{T}_i$
  - Necessary synchronizations
Model and Specification

- An agent model
  - A finite discrete transition system $\mathcal{T}_i = (S_i, s_{init,i}, A_i, T_i, \Pi_i, L_i)$
    - A transition $s_j \xrightarrow{\alpha_j} s_{j+1}$ has an unknown execution duration
  - Services $\Sigma_i$ and a labeling function $L_i : A_i \rightarrow 2^{\Pi_i} \cup 2^{E_i}$
    - $E_i = \{\epsilon_i\}$ is a special silent service
  - Synchronization requests $Sync_i = \{sync_i(I) \mid \{i\} \subseteq I \subseteq \mathcal{N}\}$
  - Behavior $\beta_i = (\tau_i = s_{i,1} \alpha_{i,1} s_{i,2} \alpha_{i,2} \ldots, \gamma_i = r_{i,1} r_{i,2} \ldots, T_i = t_{s_{i,1}} t_{\alpha_{i,1}} \ldots)$
  - Strategy $(\tau_i, \gamma_i)$
  - A collection of strategies induce a set of collective behaviors

- Specification
  - *Motion* specification: an LTL\(\neg X\) formula $\phi_i$ over $\Pi_i$
  - *Task* specification: an LTL formula $\psi_i$ over $\Sigma = \bigcup_{i \in \mathcal{N}} \Sigma_i$
Specification Semantics

- **Motion** specification: an LTL\(\not\forall\) formula \(\phi_i\) over \(\Pi_i\)
  - Interpreted on \(L_i(s_1)L_i(s_2)\ldots\) of a behavior of a single agent
    \(\beta_i = (\tau_i = s_{i,1}\alpha_{i,1}s_{i,2}\alpha_{i,2}\ldots, \gamma_i = r_{i,1}r_{i,2}\ldots, T_i = t_{s_{i,1}}t_{\alpha_{i,1}}\ldots)\)

- **Task** specification: an LTL formula \(\psi_i\) over \(\Sigma = \bigcup_{i \in \mathcal{N}} \Sigma_i\)
  - Special notion of local LTL satisfaction interpreted on a collective behavior \(\mathcal{B} = \beta_1, \ldots, \beta_N\) from agent \(i\)’s viewpoint
    \(w_{\tau_i} = L_i,\tau_1 L_i,\tau_2 \ldots\) is the subsequence of non-silent elements of
    \(v_{\tau_i} = L_i(\alpha_{i,1})L_i(\alpha_{i,2})\ldots\)
    \(\mathcal{T}_i(w_{\tau_i}) = t_{i,\tau_1}t_{i,\tau_2} \ldots\) is the subsequence of \(\mathcal{T}_i\) corresponding to \(w_{\tau_i}\)
  - \(v_{\tau_i}(t)\) is the service set that agent \(i\) provides at time \(t\)
  - \(w_{\mathcal{B}_i} = \omega_{i,\tau_1}\omega_{i,\tau_2} \ldots\), where \(\omega_{i,\tau_j} = (\bigcup_{i' \in \mathcal{N}} v_{\tau_{i'}}(t_{i,\tau_j})) \cap \Sigma\)
  - Interpreted on \(w_{\mathcal{B}_i}\)
Local LTL Satisfaction Example

Behavior $\beta_1$

- $\tau_1 = s_{1,1} \alpha_{1,1} s_{1,2} \alpha_{1,2} s_{1,3} \alpha_{1,3} \ldots$
- $\gamma_1 = \text{sync}_1(\{1, 2\}) \text{sync}_1(\{1\}) \text{sync}_1(\{1, 2\}) \ldots$

Behavior $\beta_2$

- $\tau_2 = s_{2,1} \alpha_{2,1} s_{2,2} \alpha_{2,2} s_{2,3} \alpha_{2,3} \ldots$
- $\gamma_2 = \text{sync}_2(\{1, 2\}) \text{sync}_2(\{2\}) \text{sync}_2(\{1, 2\}) \ldots$
Local LTL Satisfaction Example

Behavior $\beta_1$

- $\tau_1 = s_{1,1}\alpha_{1,1}s_{1,2}\alpha_{1,2}s_{1,3}\alpha_{1,3} \ldots$
- $\gamma_1 = sync_1(\{1, 2\})sync_1(\{1\})sync_1(\{1, 2\}) \ldots$
- $\Delta_{\alpha_{1,1}} = 1$, $\Delta_{\alpha_{1,2}} = 3$, $\ldots$

Behavior $\beta_2$

- $\tau_2 = s_{2,1}\alpha_{2,1}s_{2,2}\alpha_{2,2}s_{2,3}\alpha_{2,3} \ldots$
- $\gamma_2 = sync_2(\{1, 2\})sync_2(\{2\})sync_2(\{1, 2\}) \ldots$
- $\Delta_{\alpha_{2,1}} = 2$, $\Delta_{\alpha_{2,2}} = 1$, $\ldots$
Local LTL Satisfaction Example

Behavior $\beta_1$
- $\tau_1 = s_1, \alpha_1, s_1, \alpha_2 s_1, \alpha_3, \ldots$
- $\gamma_1 = \text{sync}_1(\{1, 2\}) \text{sync}_1(\{1\}) \text{sync}_1(\{1, 2\}) \ldots$
- $\Delta_{\alpha_1, 1} = 1, \Delta_{\alpha_1, 2} = 3, \ldots$
- $T_1 = 0 0 1 1 4 4 \ldots$

Behavior $\beta_2$
- $\tau_2 = s_2, \alpha_2, s_2, \alpha_3 s_2, \alpha_3 \alpha_2, \ldots$
- $\gamma_2 = \text{sync}_2(\{1, 2\}) \text{sync}_2(\{2\}) \text{sync}_2(\{1, 2\}) \ldots$
- $\Delta_{\alpha_2, 1} = 2, \Delta_{\alpha_2, 2} = 1, \ldots$
- $T_2 = 0 0 2 2 3 4 \ldots$
Local LTL Satisfaction Example

Behavior $\beta_1$

1. $\tau_1 = s_{1,1}\alpha_{1,1}s_{1,2}\alpha_{1,2}s_{1,3}\alpha_{1,3} \ldots$
2. $\gamma_1 = \text{sync}_1(\{1, 2\})\text{sync}_1(\{1\})\text{sync}_1(\{1, 2\})\ldots$
3. $\Delta_{\alpha_{1,1}} = 1, \Delta_{\alpha_{1,2}} = 3, \ldots$
4. $T_1 = 0 0 1 1 4 4 \ldots$
5. $v_{\tau_1} = E_1\{\sigma_{1,1}\}E_1\{\sigma_{1,1}, \sigma_{1,2}\} \ldots$
6. $w_{\tau_1} = \{\sigma_{1,1}\}\{\sigma_{1,1}, \sigma_{1,2}\} \ldots$
7. $T_i(w_{\tau_1}) = 1 4 \ldots$
8. $w_{B_1} = \{\sigma_{1,1}\}\{\sigma_{1,1}, \sigma_{1,2}, \sigma_{2,1}\} \ldots$

Behavior $\beta_2$

1. $\tau_2 = s_{2,1}\alpha_{2,1}s_{2,2}\alpha_{2,2}s_{2,3}\alpha_{2,3} \ldots$
2. $\gamma_2 = \text{sync}_2(\{1, 2\})\text{sync}_2(\{2\})\text{sync}_2(\{1, 2\})\ldots$
3. $\Delta_{\alpha_{2,1}} = 2, \Delta_{\alpha_{2,2}} = 1, \ldots$
4. $T_2 = 0 0 2 2 3 4 \ldots$
5. $v_{\tau_2} = \{\sigma_{2,1}\}E_2\{\sigma_{2,1}\} \ldots$
6. $w_{\tau_2} = \{\sigma_{2,1}\}\{\sigma_{2,1}\} \ldots$
7. $T_i(w_{\tau_1}) = 0 4 \ldots$
8. $w_{B_2} = \{\sigma_{2,1}\}\{\sigma_{1,1}, \sigma_{1,2}, \sigma_{2,1}\} \ldots$
Example 1

Agent 1 is a ground vehicle and has to avoid walls and obstacles. Agent 2 and Agent 3 are UAVs and their environment is obstacle-free except for the walls.

Motion specifications
Agent 1: Keep avoiding R1, \( \phi_1 = G \neg R_1 \).
Agent 2: Keep avoiding R2, \( \phi_2 = G \neg R_2 \).
Agent 3: Periodically survey R1 and R2, \( \phi_3 = GF R_1 \land GF R_2 \).

Task specifications
Agent 1: periodically load (−) with the help of agent 2 (−) and the assistance of agent 3 (−), then unload (|) with the help of agent 2 (−) or the assistance of agent 3 (−)

\[
\psi_1 = load \land help \land assist \land G (load \Rightarrow X (unload \land (help \lor assist))) \land \\
G (unload \Rightarrow X (load \land help \land assist))
\]

Agent 2: Periodically provide inform service (|), \( \psi_2 = GF inform \).
Agent 3: Nothing specific, \( \psi_3 = true \).
Problem Formulation

For each $i \in \mathcal{N}$, synthesize $\tau_i, \gamma_i$ so that
- the set of induced behaviors is nonempty
- the motion specification $\phi_i$ is satisfied
- the task specification $\psi_i$ is locally satisfied
Our Hierarchical Approach I

- Each $\phi_i$ is translated to a Büchi automaton $B_i^\phi$
- $N$ motion products $\mathcal{P}_i = T_i \otimes B_i^\phi$ are built
- Each motion product is reduced to $\bar{\mathcal{P}}_i$ by systematic removal of states, where no services of interest are available
- Each $\psi_i$ is translated to a Büchi automaton $B_i^\psi$
- $N$ task and motion products $\bar{\mathcal{P}}_i = \bar{\mathcal{P}}_i \otimes B_i^\psi$
- Each motion and task product is reduced to $\hat{\mathcal{P}}_i$ by systematic removal of states, where no dependent services are available
- A global product $\mathcal{P} = \hat{\mathcal{P}}_1 \otimes \ldots \otimes \hat{\mathcal{P}}_N$ containing only states relevant for planning of dependent tasks is constructed
Our Hierarchical Approach II

» An accepting run in the global product projected onto the original system gives
  » a motion plan
  » a task execution plan
  » a synchronization plan
for each agent $i$, that is correct-by-design with respect to $\phi_i$ and $\psi_i$. 
Reduction Step

Motion product $\mathcal{P}_i = \mathcal{T}_i \otimes B_i^\phi$

- A state $p$ of $\mathcal{P}_i$ is *significant* if it is either the initial state or if there exists a transition $(p, \sigma, p') \in \delta_i$, such that $\sigma \neq E_i$

- Insignificant states can be iteratively removed and replaced with transitions leading from all its predecessors to all its successors.

- The new transition is labeled with sequence of actions.
Example I Revisited

Agent 1 is a ground vehicle and has to avoid walls and obstacles. Agent 2 and Agent 3 are UAVs and their environment is obstacle-free except for the walls.

Motion specifications

Agent 1: Keep avoiding R1, $\phi_1 = G\neg R_1$.
Agent 2: Keep avoiding R2, $\phi_2 = G\neg R_2$.
Agent 3: Periodically survey R1 and R2, $\phi_3 = GF R_1 \land GF R_2$.

Task specifications

Agent 1: periodically load (−) with the help of agent 2 (−) and the assistance of agent 3 (−), then unload (|) with the help of agent 2 (−) or the assistance of agent 3 (−)

$$\psi_1 = load \land help \land assist \land G (load \Rightarrow X (unload \land (help \lor assist))) \land G (unload \Rightarrow X (load \land help \land assist))$$

Agent 2: Periodically provide inform service (|), $\psi_2 = GF inform$.
Agent 3: Nothing specific, $\psi_3 = true$. 
Example I Revisited

Centralized approach

- Each TS: 100 states
- Product TS: $100^3$ states
- $B_1^\phi, B_2^\phi, B_3^\phi, B_1^\psi, B_2^\psi, B_3^\psi$: 2, 2, 3, 2, 2, 1 states, respectively
- Intersection BA: $2 \cdot 2 \cdot 3 \cdot 2 \cdot 2 \cdot 1 \cdot 7 = 330$ states
- The overall product $P$: $\approx 330$ mil. states

Our approach:

- $P_1, P_2, P_3$: 200, 200, 300 states, respectively
- $\hat{P}_1, \hat{P}_2, \hat{P}_3$: 27, 17, 8 states, respectively
- The largest structure handled has cca 15000 states.
Remarks

- The worst-case complexity meets the complexity of the centralized solution
- Suitable for sparsely distributed services of interest and occasional needs for collaboration
- The bottleneck is still the product $\mathcal{P}$
Event-triggered Receding Horizon Approach

- Each $\phi_i$ is translated to a Büchi automaton $B_i^\phi$
- $N$ motion products $\mathcal{P}_i = T_i \otimes B_i^\phi$ are built
- Each motion product is reduced to $\mathcal{P}_i$ by systematic removal of states, where no services of interest are available
- Each $\psi_i$ is translated to a Büchi automaton $B_i^\psi$
- $N$ task and motion products $\mathcal{P}_i = \mathcal{P}_i \otimes B_i^\psi$
- Each motion and task product is reduced to $\mathcal{P}_i$ by systematic removal of states, where no dependent services are available
- A global product $\mathcal{P} = \mathcal{P}_1 \otimes \ldots \otimes \mathcal{P}_N$ containing only states relevant for planning of dependent tasks is constructed
Stepwise Receding Horizon

Agent models

Specifications

Dependency set $I_i \subseteq N$

Synchronized product $\bigotimes_{i \in I_i} P_i$ up to horizon $H$

Progressive function

Shortest path to a maximally progressive state

Found

Implement 1 step

Not found

Extend horizon $h$

Repeat

Product $\hat{P}_1 \otimes B_1^\psi$ up to horizon $h$

Dependency set $I_1 \subseteq N$

Product $\hat{P}_N \otimes B_N^\psi$ up to horizon $h$

Dependency set $I_M \subseteq N$

Horizon cannot be extended any more: Backtrack
Event-triggered Receding Horizon

Agent models

Specifications

\[ \hat{P}_1 \]

\[ \psi_1 \rightarrow B_1^{\psi} \]

Product \( \hat{P}_1 \otimes B_1^{\psi} \) up to horizon \( h \)

Dependency set \( I_1 \subseteq N \)

\[ \vdots \]

\[ \hat{P}_N \]

\[ \psi_N \rightarrow B_N^{\psi} \]

Product \( \hat{P}_N \otimes B_N^{\psi} \) up to horizon \( h \)

Dependency set \( I_M \subseteq N \)

Dependency set \( I_l \)

Synchronized product \( \bigotimes_{i \in I_l} P_i \) up to horizon \( H \)

Progressive function

Shortest path to a maximally progressive state

Found

Implement as much as you can

Not found

Extend horizon \( h \)

Repeat

Horizon cannot be extended any more: Backtrack
Example II

- Agent 1 can load \((l_H, l_A, l_B)\), carry, and unload \((u_H, u_A, u_B)\) a heavy object \(H\) or a light object \(A, B\), in the green cells.

\[
\psi_1 = \mathcal{F}(l_H \land h_H \land \mathcal{X} u_H \land \bigwedge_{i \in \{A, B\}} \mathcal{G}\mathcal{F}(l_i \land \mathcal{X} u_i)))
\]

- Agent 2 is capable of helping the agent 1 to load object \(H (h_H)\), and to execute simple tasks in the purple regions \((t_1 - t_5)\).

\[
\psi_2 = \mathcal{G}\mathcal{F}(t_1 \land \mathcal{X}(t_2 \land \mathcal{X}(t_3 \land \mathcal{X}(t_4 \land \mathcal{X} t_5 \land s_4))))
\]

- Agent 3 is capable of taking a snapshot of the rooms \((s_1 - s_5)\) when being present in there.

\[
\psi_3 = \bigwedge_{i \in \{2, 4, 5\}} \mathcal{G}\mathcal{F} s_i
\]
Example II

cca 3 mil. vs. hundreds to thousands of states
Remarks

- The worst-case complexity still the same as for the centralized case
- Suitable for collaborations executed in small (dynamically changing) subgroups
Future Work

- More interesting models and specifications
- Collision avoidance
- Decomposition of formulas

Thank you!