On Maximal Permissiveness in Partially-Observed Discrete Event Systems: Verification and Synthesis

Xiang Yin and Stéphane Lafortune

EECS Department, University of Michigan

13th WODES, May 30-June 1, 2016, Xi’an, China
Control Engineering Perspective

- $E = E_c \cup E_{uc} = E_o \cup E_{uo}$
- Supervisor: $S: E_o^* \rightarrow 2^E$; Disable events in $E_c$ based on its observations
- Closed-loop Behavior: $L(S/G)$
• $G = (X, E, f, x_0)$ is a deterministic FSA
  - $X$ is the finite set of states
  - $E$ is the finite set of events
  - $f : X \times E \rightarrow X$ is the partial transition function
  - $x_0$ is the initial state

• Safety specification automaton: $L(H) \subseteq L(G)$
Introduction

- $G = (X, E, f, x_0)$ is a deterministic FSA
  - $X$ is the finite set of states
  - $E$ is the finite set of events
  - $f: X \times E \to X$ is the partial transition function
  - $x_0$ is the initial state

- Safety specification automaton: $L(H) \subseteq L(G)$

We say that a supervisor $S: E_o^* \to 2^E$ is

- **Safe**, if $L(S/G) \subseteq L(H)$
- **Maximally Permissive**, if for any safe supervisor $S'$, we have $L(S/G) \nsubseteq L(S'/G)$. 
Introduction

1. $G = (X, E, f, x_0)$ is a deterministic FSA
   - $X$ is the finite set of states
   - $E$ is the finite set of events
   - $f: X \times E \to X$ is the partial transition function
   - $x_0$ is the initial state

2. Safety specification automaton: $L(H) \subseteq L(G)$

We say that a supervisor $S: E_o^* \to 2^E$ is
   - Safe, if $L(S/G) \subseteq L(H)$
   - Maximally Permissive, if for any safe supervisor $S'$, we have $L(S/G) \not\subseteq L(S'/G)$. 

\[ L(H) \subseteq L(G) \]

\[ Max_1 \cap Max_2 \]

\[ L(H) \]

\[ L(G) \]

- Supremal normal and controllable solution


- Supremal normal and controllable solution
- Solutions larger than supremal normal and controllable solution

- Supremal normal and controllable solution
- Solutions larger than supremal normal and controllable solution
- These solutions are sound but not complete
Literature Review


- Supremal normal and controllable solution
- Solutions larger than supremal normal and controllable solution
- These solutions are sound but not complete


- Solutions are both sound and complete
- A certain class of maximal policies


- Supremal normal and controllable solution
- Solutions larger than supremal normal and controllable solution
- These solutions are sound but not complete


- Solutions are both sound and complete
- A certain class of maximal policies

\[ Max \subseteq L(H) \]
• **Supervisor Verification Problem.**

Given a safe supervisor $S_R : E^*_o \rightarrow 2^E$, verify whether or not $S_R$ is maximal.
• **Supervisor Verification Problem.**

Given a safe supervisor \( S_R : E_o^* \rightarrow 2^E \), verify whether or not \( S_R \) is maximal.

• **Supervisor Synthesis Problem.**

Given a non-maximal safe supervisor \( S_R : E_o^* \rightarrow 2^E \), find a safe supervisor \( S \) such that \( L(S_R / G) \subseteq L(S / G) \).
• **Supervisor Verification Problem.**

Given a safe supervisor $S_R : E^*_o \rightarrow 2^E$, verify whether or not $S_R$ is maximal.

• **Supervisor Synthesis Problem.**

Given a non-maximal safe supervisor $S_R : E^*_o \rightarrow 2^E$, find a safe supervisor $S$ such that $L(S_R/G) \subseteq L(S/G)$.

**Motivation:**

• Lower bound behavior $L_r$

• $L(S_R/G) = L_r \downarrow^{CO}$, the infimal controllable and observable super-language

• Achieve both the lower bound and permissiveness
**Information State**: a set of states, $I := 2^X$

**BTS**: A bipartite transition system $T$ w.r.t. $G$ is a 7-tuple

$$T = (Q^T_Y, Q^T_Z, h^T_{YZ}, h^T_{ZY}, E, \Gamma, y_0)$$

where
- $Q^T_Y \subseteq I$ is the set of Y-states;
- $Q^T_Z \subseteq I \times \Gamma$ is the set of Z-states so that $z = (I(z), \Gamma(z));$
- $h^T_{YZ}: Q^T_Y \times \Gamma \rightarrow Q^T_Z$ represents the unobservable reach;
- $h^T_{ZY}: Q^T_Z \times E \rightarrow Q^T_Y$ represents the observation transition;

Information State: a set of states, $I := 2^X$
**Information State:** a set of states, \( I := 2^X \)

**BTS:** A bipartite transition system \( T \) w.r.t. \( G \) is a 7-tuple

\[
T = (Q_Y^T, Q_Z^T, h_{YZ}^T, h_{ZY}^T, E, \Gamma, y_0)
\]

where

- \( Q_Y^T \subseteq I \) is the set of Y-states;
- \( Q_Z^T \subseteq I \times \Gamma \) is the set of Z-states so that \( z = (I(z), \Gamma(z)) \);
- \( h_{YZ}^T: Q_Y^T \times \Gamma \rightarrow Q_Z^T \) represents the unobservable reach;
- \( h_{ZY}^T: Q_Z^T \times E \rightarrow Q_Y^T \) represents the observation transition;

\[
\begin{align*}
E_c &= \{c_1, c_2\}, \\
E_o &= \{a, b\}
\end{align*}
\]

**The given supervisor \( S_R \) can be realized as a BTS \( T_R \)**
Definition. (AIC).

The All Inclusive Controller

\( \mathcal{AIC}(G) = (Q_Y^{\text{AIC}}, Q_Z^{\text{AIC}}, h_Y^{\text{AIC}}, h_Z^{\text{AIC}}, E, \Gamma, y_0) \)

is defined as the largest BTS such

1. For any \( y \in Q_Y^{\text{AIC}} \), there exists at least one control decision
2. For any \( z \in Q_Z^{\text{AIC}} \), we have
   2.1. all feasible observable events are defined
   2.2. \( I(z) \) only contains legal states

\[ E_c = \{c_1, c_2\}, E_o = \{a, b\} \]
Definition. (AIC).
The All Inclusive Controller
\( \mathcal{AIC}(G) = (Q^\text{AIC}_Y, Q^\text{AIC}_Z, h^\text{AIC}_{YZ}, h^\text{AIC}_{ZY}, E, \Gamma, y_0) \),
is defined as the largest BTS such
1. For any \( y \in Q^\text{AIC}_Y \), there exists at least one control decision
2. For any \( z \in Q^\text{AIC}_Z \), we have
   2.1. all feasible observable events are defined
   2.2. \( I(z) \) only contains legal states

- The AIC contains all safe supervisors
Verification of Maximality: Basic Idea and Difficulties

$T_R$: realizes the given supervisor

$\text{AIC: includes all safe supervisors}$

X.Yin & S.Lafortune (UMich)
Verification of Maximality: Basic Idea and Difficulties

$T_R$: realizes the given supervisor

$\text{AIC}$: includes all safe supervisors

• “Compare” $T_R$ with the AIC
Verification of Maximality: Basic Idea and Difficulties

\( T_R \): realizes the given supervisor
\( AIC \): includes all safe supervisors

- “Compare” \( T_R \) with the AIC
- How to compare?
- The effect of enabling an event depends on future information
Verification of Maximality: Basic Idea and Difficulties

$T_R$: realizes the given supervisor

$\text{AIC}$: includes all safe supervisors

- “Compare” $T_R$ with the AIC
- How to compare?
- The effect of enabling an event depends on future information

$E_c = \{c\}, E_o = \{o\}$
Verification of Maximality: Basic Idea and Difficulties

**Verification of Maximality**

**Basic Idea and Difficulties**

**X.Yin** & **S.Lafortune**

(UMich) May 2016

WODES 2016

**$T_R$:** realizes the given supervisor

**AIC:** includes all safe supervisors

- “Compare” $T_R$ with the AIC
- How to compare?
- The effect of enabling an event depends on future information

$L(S_R/G)$

$E_c = \{c\}, E_o = \{o\}$
Verification of Maximality: Basic Idea and Difficulties

\[ T_R : \text{realizes the given supervisor} \]

\[ \text{AIC: includes all safe supervisors} \]

- "Compare" \( T_R \) with the AIC
- How to compare?
- The effect of enabling an event depends on future information

\[ E_c = \{c\}, E_o = \{o\} \]
Verification of Maximality: Basic Idea and Difficulties

\( T_R \): realizes the given supervisor
\( AIC \): includes all safe supervisors

- “Compare” \( T_R \) with the AIC
- How to compare?
- The effect of enabling an event depends on future information

\[ L(S_R/G) \]

\( E_c = \{c\}, E_o = \{o\} \)
Verification of Maximality: Basic Idea and Difficulties

$T_R$: realizes the given supervisor

$AIC$: includes all safe supervisors

- “Compare” $T_R$ with the AIC
- How to compare?
- The effect of enabling an event depends on future information

$E_c = \{c\}, E_o = \{o\}$

$L(S_R/G)$ Conflict!
Definition. (Control Simulation Relation)

Let $T_1$ and $T_2$ between BTSs. A relation $\Phi = \Phi_Y \cup \Phi_Z \subseteq (Q_{Y}^{T_1} \times Q_{Y}^{T_2}) \times (Q_{Z}^{T_1} \times Q_{Z}^{T_2})$ is said to be a control simulation relation from $T_1$ to $T_2$ if the following conditions hold:

1. $(y_0, y_0) \in \Phi_Y$;

2. For every $(y_1, y_2) \in \Phi_Y$, we have that: for any $y_1 \xrightarrow{\gamma_1} z_1$ in $T_1$, there exists $y_2 \xrightarrow{\gamma_2} z_2$ such that $(z_1, z_2) \in \Phi_Z$ and $\gamma_1 \subseteq \gamma_2$.

3. For every $(z_1, z_2) \in \Phi_Z$, we have that: for any $z_1 \xrightarrow{\sigma} y_1$ in $T_1$, there exists $z_2 \xrightarrow{\sigma} y_2$ such that $(y_1, y_2) \in \Phi_Y$. 
Definition. (Control Simulation Relation)

Let $T_1$ and $T_2$ between BTSs. A relation $\Phi = \Phi_Y \cup \Phi_Z \subseteq (Q_Y^{T_1} \times Q_Y^{T_2}) \times (Q_Z^{T_1} \times Q_Z^{T_2})$ is said to be a control simulation relation from $T_1$ to $T_2$ if the following conditions hold:

1. $(y_0, y_0) \in \Phi_Y$;
2. For every $(y_1, y_2) \in \Phi_Y$, we have that: for any $y_1 \xrightarrow{\gamma_1} z_1$ in $T_1$, there exists $y_2 \xrightarrow{\gamma_2} z_2$ such that $(z_1, z_2) \in \Phi_Z$ and $\gamma_1 \subseteq \gamma_2$.
3. For every $(z_1, z_2) \in \Phi_Z$, we have that: for any $z_1 \xrightarrow{\sigma} y_1$ in $T_1$, there exists $z_2 \xrightarrow{\sigma} y_2$ such that $(y_1, y_2) \in \Phi_Y$.

• There exists a unique maximal CSR $\Phi^*(T_1, T_2)$ from $T_1$ to $T_2$ if one exists
• $\Phi^*(T_1, T_2)$ can be computed by

$$\Phi^*(T_1, T_2) = \lim_{k\to\infty} F^k((Q_Y^{T_1} \times Q_Y^{T_2}) \cup (Q_Z^{T_1} \times Q_Z^{T_2}))$$
Verification of Maximality: Solution

\[ \Phi^* \]

\[ TR \]

\[ AIC \]
Replacement.
Let $y$ be a Y-state in $T_R$ and $c_{T_R}(y)$ be the control decision defined at $y$.
We say that control decision $\gamma$ replaces $c_{T_R}(y)$ at $y$ if
1. $\gamma$ is defined at $y$ in the AIC
2. $c_{T_R}(y) \subset \gamma$
3. $(z, z') \in \Phi^*(T_R, AIC)$, where $y \xrightarrow{c_{T_R}(y)} z$ and $y \xrightarrow{\gamma} z'$
Verification of Maximality: Solution

Replacement.
Let $y$ be a $Y$-state in $T_R$ and $c_{T_R}(y)$ be the control decision defined at $y$.
We say that control decision $\gamma$ replaces $c_{T_R}(y)$ at $y$ if
1. $\gamma$ is defined at $y$ in the AIC
2. $c_{T_R}(y) \subset \gamma$
3. $(z, z') \in \Phi^*(T_R, AIC)$, where $y \xrightarrow{c_{T_R}(y)} z$ and $y \xrightarrow{\gamma} z'$

Theorem.
Supervisor $S_R$ is maximal iff no control decision in $T_R$ can be replaced
Synthesis of Larger Supervisor: Basic Idea and Difficulties

• Construct a new BTS: replace the original control decision in $T_R$ by a larger one
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- Construct a new BTS: replace the original control decision in $T_R$ by a larger one
- Such a BTS may not exist!
• Construct a new BTS: replace the original control decision in $T_R$ by a larger one
• Such a BTS may not exist!
Construct a new BTS: replace the original control decision in $T_R$ by a larger one

Such a BTS may not exist!
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- Construct a new BTS: replace the original control decision in $T_R$ by a larger one
- Such a BTS may not exist!
• Construct a new BTS: replace the original control decision in $T_R$ by a larger one
• Such a BTS may not exist!
• Information Merge Phenomenon: Information is lost
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- Construct a new BTS: replace the original control decision in $T_R$ by a larger one
- Such a BTS may not exist!
- Information Merge Phenomenon: Information is lost
- $2^X$ may not be sufficient for the synthesis problem!
The Role of Strict Sub-automaton

- The information merge phenomenon \textbf{will not occur} under the assumption that $R$ is a \textit{strict sub-automaton} of $G$, where $L(R) = L(S_R/G)$.
The Role of Strict Sub-automaton

- The information merge phenomenon will not occur under the assumption that \( R \) is a strict sub-automaton of \( G \), where \( L(R) = L(S_R/G) \).
- **Strict sub-automaton**: if a string goes outside, then it stays outside forever.
The information merge phenomenon **will not occur** under the assumption that $R$ is a strict sub-automaton of $G$, where $L(R) = L(S_R/G)$.

- **Strict sub-automaton**: if a string goes outside, then it stays outside forever.
- We can always obtain strict sub-automaton [Cho & Marcus, 1989]
The Role of Strict Sub-automaton

\[ R \rightarrow G \]

Sub-automaton
The Role of Strict Sub-automaton

Refine

\[ \mathcal{R}, \mathcal{G} \rightarrow \mathcal{G}' \]
The Role of Strict Sub-automaton

\[ R \rightarrow G \rightarrow G' \]

Refine

Strict sub-automaton
The Role of Strict Sub-automaton

---

Strict sub-automaton

Refine

AIC for the refined system
The Role of Strict Sub-automaton

**Strict sub-automaton**

\[ R \rightarrow G \rightarrow G' \]

Refine

**AIC for the refined system**

\[ \{3\}, \{\} \rightarrow \{3\} \rightarrow \{0\}, \{\} \rightarrow \{0\} \rightarrow \{0,1\}, \{c_1\} \rightarrow \{3',4'\} \rightarrow \{3',4',\} \rightarrow \{3',4',6'\}, \{c_2\} \rightarrow \{3',5,5'\}, \{c_1\} \]

\[ {a',b'} \rightarrow \{3\} \rightarrow \{0\} \rightarrow \{0\} \rightarrow \{0,1\}, \{c_1\} \rightarrow \{3',4'\} \rightarrow \{3',4',\} \rightarrow \{3',4',6'\}, \{c_2\} \rightarrow \{3',5,5'\}, \{c_1\} \]
Synthesis Algorithm

**Synthesis Steps:**
1. Construct BTS $T_R$ and $AIC(G)$ (make sure $R$ is a strict sub-automaton of $G$)
2. Compute the maximal CSR $\Phi^*$ from $T_R$ to $AIC(G)$
3. Find a Y-state $y$ in $T_R$ such that its control decision can be replaced by $\gamma$
4. Construct BTS $T^*$ by
   - For Y-state $y$, choose $\gamma$ which is larger than $c_{T_R}(y)$
   - For other Y-states, choose the same control decisions in $T_R$
Synthesis Steps:
1. Construct BTS $T_R$ and $\mathcal{AIC}(G)$ (make sure $R$ is a strict sub-automaton of $G$)
2. Compute the maximal CSR $\Phi^*$ from $T_R$ to $\mathcal{AIC}(G)$
3. Find a Y-state $y$ in $T_R$ such that its control decision can be replaced by $\gamma$
4. Construct BTS $T^*$ by
   - For Y-state $y$, choose $\gamma$ which is larger than $c_{T_R}(y)$
   - For other Y-states, choose the same control decisions in $T_R$
Synthesis Algorithm

Synthesis Steps:
1. Construct BTS $T_R$ and $AIC(G)$ (make sure $R$ is a strict sub-automaton of $G$)
2. Compute the maximal CSR $\Phi^*$ from $T_R$ to $AIC(G)$
3. Find a Y-state $y$ in $T_R$ such that its control decision can be replaced by $\gamma$
4. Construct BTS $T^*$ by
   - For Y-state $y$, choose $\gamma$ which is larger than $c_{T_R}(y)$
   - For other Y-states, choose the same control decisions in $T_R$
Synthesis Algorithm

Synthesis Steps:
1. Construct BTS $T_R$ and $\mathcal{AIC}(G)$ (make sure $R$ is a strict sub-automaton of $G$)
2. Compute the maximal CSR $\Phi^*$ from $T_R$ to $\mathcal{AIC}(G)$
3. Find a Y-state $y$ in $T_R$ such that its control decision can be replaced by $\gamma$
4. Construct BTS $T^*$ by
   - For Y-state $y$, choose $\gamma$ which is larger than $c_{T_R}(y)$
   - For other Y-states, choose the same control decisions in $T_R$
Synthesis Algorithm

Synthesis Steps:
1. Construct BTS $T_R$ and $AIC(G)$ (make sure $R$ is a strict sub-automaton of $G$)
2. Compute the maximal CSR $\Phi^*$ from $T_R$ to $AIC(G)$
3. Find a Y-state $y$ in $T_R$ such that its control decision can be replaced by $\gamma$
4. Construct BTS $T^*$ by
   - For Y-state $y$, choose $\gamma$ which is larger than $c_{T_R}(y)$
   - For other Y-states, choose the same control decisions in $T_R$
Synthesis Algorithm

Synthesis Steps:
1. Construct BTS $T_R$ and $AIC(G)$ (make sure $R$ is a strict sub-automaton of $G$)
2. Compute the maximal CSR $\Phi^*$ from $T_R$ to $AIC(G)$
3. Find a Y-state $y$ in $T_R$ such that its control decision can be replaced by $\gamma$
4. Construct BTS $T^*$ by
   - For Y-state $y$, choose $\gamma$ which is larger than $c_{T_R}(y)$
   - For other Y-states, choose the same control decisions in $T_R$

$\mathbf{2^X \text{ is sufficient!}}$
Achieve More Permissiveness

A maximal supervisor

Yes

Is $T_R$ maximal?

No

Refine the state-space of $R$ s.t. $R \subseteq G$

Find a supervisor $S_R'$ strictly larger than $S_R$

$S_R \leftarrow S'_R$

$L(R) \leftarrow L(S_R/G)$
Achieve More Permissiveness

Is $T_R$ maximal?

Yes

A maximal supervisor

No

Refine the state-space of $R$ s.t. $R \subseteq G$

Find a supervisor $S'_R$ strictly larger than $S_R$

$S_R \leftarrow S'_R$

$L(R) \leftarrow L(S_R/G)$

Iteration may not converge!
Conclusion

Contribution

• Verify whether or not a given supervisor is maximal
• The notion of control simulation relation
• Synthesis a new supervisor that is strictly more permissive
• Information Merge Phenomenon & Strict sub-automaton
Conclusion

Contribution

• Verify whether or not a given supervisor is maximal
• The notion of control simulation relation
• Synthesis a new supervisor that is strictly more permissive
• Information merge phenomenon & Strict sub-automaton

On Going Work

• Synthesize a maximal supervisor that contains the given one