Modeling Interdependent Critical Infrastructures using Open Hybrid Automata

Constantinos Heracleous, Christos G. Panayiotou, Marios M. Polycarpou, Georgios Ellinas

✉ heracleous.constantinos@ucy.ac.cy

Researcher at KIOS Research Center for Intelligent Systems and Networks, University of Cyprus

This work is supported by the European Research Council Advanced Grant FAULT-ADAPTIVE (ERC-AdG-291508).
Introduction

There are key Interdependencies between Critical Infrastructures usually classified as:

- physical
- cyber
- geographic
- logical


Modeling and Simulation (M&S) methodologies are typically used to study and analyze CIs interdependencies, and to generate information regarding the consequences of failures and disruptions to multiple CIs


M&S approaches:
- Inoperability Input-Output Model
- Agent-based approaches

Unfortunately, this is a complex problem and existing approaches lack generalization and usually difficult to merged together into a unified framework

Agent-based approaches consider CIs as complex adaptive systems and represent their components as agents, each with a set of rules that direct their goals.

Other approaches include:
- Fault trees
- Reliability block diagrams
- System dynamics
- Mixed Hanoi-configuration (MHC) framework
- Geographical Information Systems (GIS)

Network-based approaches
- Conflict network
- Network flow models

In this approach, usually models denote different infrastructural components and links denote their physical and relational connections.
M&S approaches:

Inoperability Input-Output Model

\[ x = Ax + c \]

IIM considers each infrastructure as an atomic entity, whose level of operability depends on the availability of resources supplied by the other infrastructures.

Agent-based approaches consider CIs as complex adaptive systems and represent their components as agents, each with a set of rules that direct their goals.

Unfortunately, this is a complex problem and existing approaches lack generalization and usually difficult to merged together into a unified framework.

Other approaches include:

- Federated simulations (e.g. high-level architecture (HLA), Open Modelling Interface (OpenMI))
- Bayesian Networks
- General Stochastic Petri Nets
- Reliability Block Diagrams
- System Dynamics
- Mixed Holistic-Reductionistic (MHR) framework
- Geographical Information Systems (GIS)

Network-based approaches

- Graph theory based
- Complex networks
- Network flow models

In this approaches usually nodes denote different infrastructure components and links denote their physical and relational connections.
Hybrid Systems and Interdependent Critical Infrastructures Modeling

Hybrid Systems combine both continuous-time and discrete-event dynamics into a single model:

They can serve as models of complex large scale systems and they have been already used to model individual CIs such as:
- Electric Power Systems
- Communication Networks
- Transportation Systems
- Natural Gas Transmission Network

They haven't been exploited yet for modeling multiple critical infrastructures and their interdependencies.

Use hybrid automata to model constituent components of different infrastructures which can influence each other.

Final step, is to compose together all the CI components open hybrid automata models, creating a single model that represents the interdependent CIs under investigation.

We do this by using composition properties, which basically connect some inputs/outputs variables of one hybrid automaton with some outputs/inputs variables of another.

For some infrastructure components the internal input variables are used, while for different infrastructure components the dependency inputs are used.

- We can represent this behavior by using open (or IO) hybrid automata to model CI components.
- We make the formulation more specific for CI components by dividing the input variables into internal inputs and dependency inputs.

\[ C = (Q, X, Y, J, f, h, e, E, F, W) \]

where:
- \( Q \) is a set of discrete states
- \( X \) is a set of continuous state variables
- \( Y \) is a set of continuous output variables
- \( J \) is a set of discrete output variables
- \( f \) is a function \( f: Q \times X \times J \rightarrow \mathbb{R} \)
- \( h \) is a function \( h: Q \times X \times J \rightarrow \mathbb{R} \)
- \( e \) is a vector \( e: Q \times X \times J \rightarrow \mathbb{R} \)
- \( E \) is the set of edges
- \( F \) is the set of transitions
- \( W \) is the set of modes

- Internal variables and dependency variables that influence the state transitions and continuous state evolution are combined together.
- \( Q, X, Y, J, f, h, e, E, F, W \) are the output functions in each discrete state.
- \( f \) is a function of the set of inputs that drive the system and \( h \) is each continuous state.
- \( e \) is the set of edges that represent transitions between discrete states.
- \( W \) is a directed graph for each edge that defines the discrete transition function.
- \( f, h, e \) are functions of the continuous state variables.
Hybrid Systems combine both continuous-time and discrete-event dynamics into a single model:

They can serve as models of complex large scale systems and they have been already used to model individual CIs such as:

- Electric Power Systems
- Communication Networks
- Transportation Systems
- Natural Gas Transmission Network

They haven't been exploit yet for modeling multiple critical infrastructures and their interdependencies
Use hybrid automata to model constituent components of different infrastructures which can influence each other.

In case of dependencies and interdependencies discrete transitions and continuous state variations in *Infrastructure A component* can influence *Infrastructure B component* and vice-versa.
• We can represent this behavior by using open (or IO) hybrid automata to model CI components.
• We make the formulation more specific for CI components by dividing the input variables into \textit{internal} inputs and \textit{dependency} inputs.

\textit{Definition (Open hybrid automaton):} An open hybrid automaton model $C_j^l$ for a component $j = 1,2,...$ of an infrastructure $l = A,B,...$ is a collection:

$$C_j^l = (Q, X, V, Y, \text{Init}, f, h, \text{Inv}, E, G, R)$$

where:
• $Q$ is a set of discrete states;
• $X$ is a set of continuous state variables;
• $V$ is a set of input variables, continuous or discrete, divided into two subsets $V = \overline{V} \cup \overline{V}$ with:
  • $\overline{V}$ denoting \textit{internal} inputs to the model, for connections between component models of the same infrastructure;
  • $\overline{V}$ denoting \textit{dependency} inputs to the model, for connections between components of different infrastructures;
• $Y$ is a set of output variables, continuous and discrete;
• $f: Q \times X \times V \rightarrow R^n$ are the state equations that describe the evolution of the continuous state variables over time in each discrete state;
• $h: Q \times X \times V \rightarrow Y$ are the output functions in each discrete state;
• $\text{Inv}$: invariant set that denotes the set of values that $x$ and $v$ take in each discrete state $q$;
• $E$: collection of edges that represent transitions between discrete states;
• $G$: guard function for each edge that denotes the discrete transition firing condition.
• $R$: reset relation for continuous states before discrete transition.

CI component graphical representation

Guard relation, function of $x$ and $u$

Reset function
Final step, is to composed together all the CI components open hybrid automata models, creating a single model that represents the interdependent CIs under investigation.

We do this by using composition properties, which basically connect some inputs/outputs variables of one hybrid automaton with some outputs/inputs variables of another.

For same infrastructure components the internal input variables are used, while for different infrastructure components the dependency inputs are used.
Hybrid Systems and Interdependent Critical Infrastructures Modeling

Hybrid Systems combine both continuous-time and discrete-event dynamics into a single model:

They can serve as models of complex large scale systems and they have been already used to model individual CIs such as:
- Electric Power Systems
- Communication Networks
- Transportation Systems
- Natural Gas Transmission Network

They haven't been exploited for modeling multiple critical infrastructures and their interdependencies.

Use hybrid automata to model constituent components of different infrastructures which can influence each other.

Final step, is to composed altogether all the CI components open hybrid automata models, creating a single model that represents the interdependent CIs under investigation.

We do this by using composition properties, which basically connect some inputs/outputs variables of one hybrid automaton with some outputs/inputs variables of another.

For some infrastructure components the internal input variables are used, while for different infrastructure components the dependency inputs are used.

- We can represent this behavior by using open (or IO) hybrid automata to model CI components.
- We make the formulation more specific for CI components by dividing the input variables into internal inputs and dependency inputs.

Hybrid Automata representation: An open hybrid automaton model C, for a component, is defined by an infrastructure \( V \in A \), is a collection:

\[ C = (Q, X, Y, J, f, h, x_0, E, R) \]

where:
- \( Q \) is a set of discrete states
- \( X \) is a set of continuous state variables
- \( Y \) is a set of system outputs, continuous or discrete
- \( J \) is a set of external inputs, continuous or discrete
- \( f \) is an evolution function, continuous or discrete
- \( h \) is a dependence function, continuous or discrete
- \( x_0 \) is the initial state
- \( E \) is an edge set
- \( R \) is a set of edges that represent transitions between discrete states
- A dashed line for each edge that denotes the discrete transition relationship

A more detailed explanation is found in the paper by Ela et al.
Modeling Example

Simulation Results

Simulation of 24 hour period where fault/inputs introduced to each component:

1. @10:35 Power overload occurs to the substation
   - The Communication Station switches to the emergency UPS, while SCADA notifies the operator.
   - @10:45 Once the safety period passes the power is restored and all the components return to their initial state.

2. @10:30 Technical fault occurs to the Communication Station
   - SCADA loses communication with the Power Substation, and the power demand drops.
   - @11:40 The technical fault is resolved and all the components return to their initial state.

3. @18:00 The SCADA operator commands the opening of the Substation switches due to schedule maintenance.
   - The Substation transitions to the Switch Off state, and the Communication Station switches to the emergency UPS.
   - @19:00 The UPS support period ends, and the communication station stops operating.
   - SCADA loses communication with the Power Substation, not able to control the substation and switch it automatically ON.
Power Distribution Substation

- Receives power from the transmission network.
- Supplies power to the area, including the communication station.
- Has overload protection.
- Is monitored and controlled by the SCADA system.

SCADA

- Provides communication services to the area.
- Receives power from the substation, but is also equipped with a UPS in case of emergency.

- Monitors the substation by receiving power measurements through the network.
- Controls the substation by sending commands through the network.
Power Distribution Substation

Available Power

SCADA Control Signal

Area Power Demand

\[ \begin{bmatrix} \overline{\nu}_{1,1}^P \\ \overline{\nu}_{1,2}^P \end{bmatrix} \]

**Supply Power**

\[ \dot{x}_{1,1}^P = 0 \]
\[ y_{1,1}^P = \overline{\nu}_{1,2}^P \]
\[ [\overline{\nu}_{1,1}^P = 0 \land \overline{\nu}_{1,2}^P < \overline{\nu}_{1,1}^P] \]

\[ q_{1,1}^P \]

\[ \overline{\nu}_{1,1}^P = 1 \lor \overline{\nu}_{1,2}^P \geq \overline{\nu}_{1,1}^P \]

\[ \begin{bmatrix} y_{1,1}^P \end{bmatrix} \]

**Switch Off**

\[ \dot{x}_{1,1}^P = 1 \]
\[ y_{1,1}^P = 0 \]
\[ [x_{1,1}^P \geq 0] \]

\[ \begin{bmatrix} \overline{\nu}_{1,1}^P \end{bmatrix} = 0 \land x_{1,1}^P \geq T_{safe} \]
SCADA

$q_{2.1}^P : Close$
\[ x_{2.1}^P := 0 \]
\[ y_{2.1}^P = 0 \]
\[ [\overline{\mathcal{P}}_{2.1} > 0] \]
\[ \dot{x}_{2.1}^P = 1 \]
\[ \overline{\mathcal{P}}_{2.1} = -1 \]

$q_{2.2}^P : Open$
\[ x_{2.1}^P := 0 \]
\[ y_{2.1}^P = 1 \]
\[ [\overline{\mathcal{P}}_{2.1} = 0 \lor \overline{\mathcal{P}}_{2.1} = 1] \]
\[ \dot{x}_{2.1}^P = 1 \]
\[ \overline{\mathcal{P}}_{2.1} = -1 \]

$q_{2.3}^P : No Data$
\[ x_{2.1}^P = 0 \]
\[ y_{2.1}^P = -1 \]
\[ [\overline{\mathcal{P}}_{2.1} = -1] \]
\[ \dot{x}_{2.1}^P = 0 \]
\[ \overline{\mathcal{P}}_{2.1} = 0 \]
Communication Station and Network

- **Healthy** ($q_{1,1}^C$): $\dot{x}_{1,1}^C = 0$
  - $y_{1,k}^C(t) = \overline{v}_{1,k}^C(t + l_k)$
  - $y_{1,3}^C = P_{nom} [\overline{v}_{1,1}^C = 0 \land \overline{v}_{1,3}^C \geq P_{nom}]$

- **under UPS** ($q_{1,2}^C$): $\dot{x}_{1,1}^C = 1$
  - $y_{1,k}^C(t) = \overline{v}_{1,k}^C(t + l_k)$
  - $y_{1,3}^C = 0 [x_{1,1}^C \leq T_{UPS}]$

- **Fault** ($q_{1,3}^C$): $\dot{x}_{1,1}^C = 0$
  - $y_{1,k}^C = -1$
  - $y_{1,3}^C = 0 [\overline{v}_{1,1}^C = 1 \lor \overline{v}_{1,3}^C < P_{nom}]$
  - $k = 1, 2$
Composed Hybrid Automaton Model

Communication Station Power Demand

Physical Dependencies

Cyber Dependencies

Logical Dependencies

Matlab Simulink Implementation
Simulation Results

Simulation of 24 hour period where fault/events introduced to each component.

1. \( @02:35 \) Power overload occurs to the substation
   \[ \rightarrow \] The Communication Station switches to the emergency UPS, while SCADA notifies the operator.
   
   \( @02:45 \) Once the safety period passes the power is restored and all the components return to their initial state.

2. \( @10:20 \) Technical fault occurs to the Communication Station
   \[ \rightarrow \] SCADA loses communication with the Power Substation, and the power demand drops.
   
   \( @11:40 \) The technical fault is resolved and all the components return to their initial state.

3. \( @18:00 \) The SCADA operator commands the opening of the Substation switches due to schedule maintenance.
   \[ \rightarrow \] The Substation transitions to the Switch Off state, and the Communication Station switches to the emergency UPS.
   
   \( @19:00 \) The UPS support period ends, and the communication station stops operating.
   \[ \rightarrow \] SCADA loses communication with the Power Substation, not able to control the substation and switch it automatically ON.
Conclusions & Future Work

• We adopted open hybrid automata to model components of different CIs including their dependencies.
• The components models can have the necessary abstraction level and they can be composed together to represent interdependent CIs.
• The simulation results demonstrated that this approach can generate sequence of events on how a fault can cascade or escalate through dependencies and interdependencies into multiple infrastructures.

• In the future we plan to use this approach to implement more complex CI component models with more sophisticated dynamics and also,
• Exploit hybrid systems reachability analysis to discover the sets of inputs that can push the models to undesirable states.
THANK YOU