Assuring Autonomy in Contested Environments

Attack-Resilient Design

Miroslav Pajic

Cyber-Physical Systems Lab (CPSL)

Pratt School of Engineering

Duke University

Security-Aware Design of Autonomous Systems

Duke PRATT SCHOOL *of* ENGINEERING

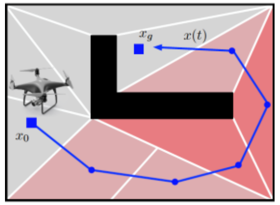
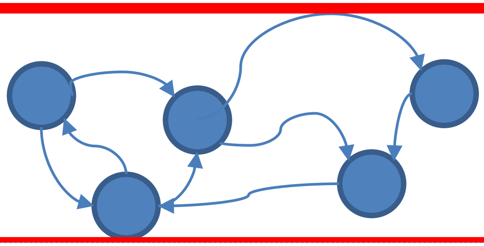
- Physical world abides by the laws of physics!
- Physical interfaces introduce new attack vectors!

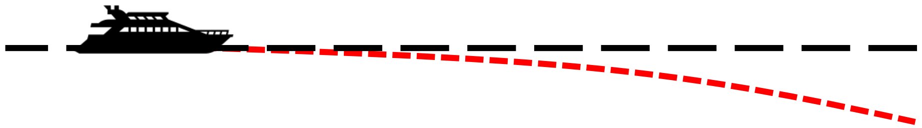
- How can we exploit ***limited*** knowledge of laws of physics (system model) for control and attack detection/identification

- Attack-Resilient design with *uncertainty, resource/platform constraints*, as well as varying (especially high) levels of autonomy
  - How much can the attacker exploit modeling limitation?
  - How can we effectively exploit physics to improve guarantees in the presence of attacks?

Security-Aware Control for Autonomous Systems

Duke PRATT SCHOOL of ENGINEERING

Control Stack	Control view	Modeling view	Adding Resiliency
Mission Planner	Long-horizon views		[ICRA19, ICRA20a, ICRA20b*, CAV'19a, THMS19]
Tactical Planner	Short-horizon views		[CDC19a,CDC19b, TAC19*, TII19]
Low-level Control	Continuous/discrete control with constraints	$f_r(x(t)) = \int_0^T \varrho(x(t),h(t))dt + \int_0^T \|x(t)\|^2 dt,$ $\min \; f_r(x_r(t)) + f_h(x_h(t))$ $\text{s.t.} \; x_r(t) = x_h(t), \; u_r(t) = u_h(t),$	[TAC19a ,TAC19b, TCPS20*, ACC20*, AUT20a*, AUT19*,AUT18, TECS17, RTSS17,TCNS17, CSM17, CDC17,CDC18,…]
Vehicle			



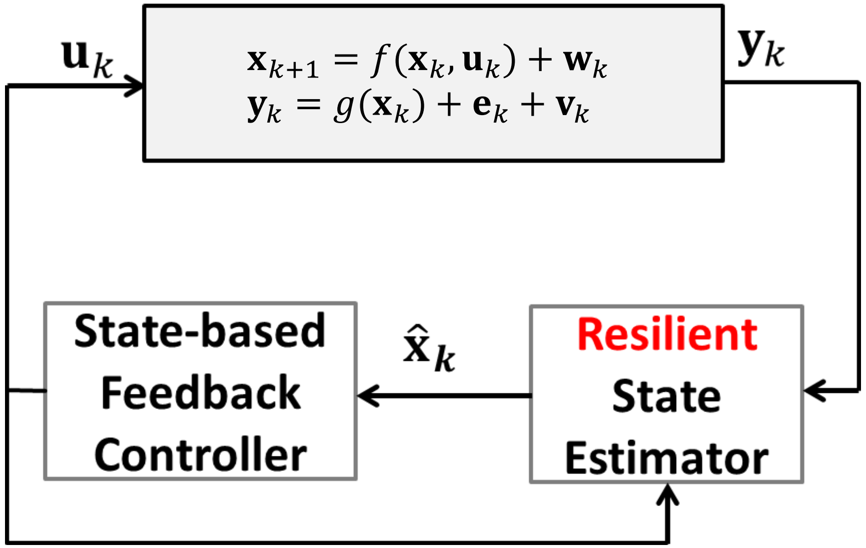
Our Goal: Add resiliency to controls across different/all levels of control stack

Attack-resilient State Estimation

Duke
PRATT SCHOOL of ENGINEERING

- Attack-resilient control of Cyber-Physical Systems
  - Idea: Design attack-resilient state estimators

- Attack model
  - Goal: force the system into an unsafe state by creating a discrepancy between states and the estimates
  - Attacker has the ability to inject any signal using the compromised sensors
  - Attacker has full system knowledge and unlimited computational power



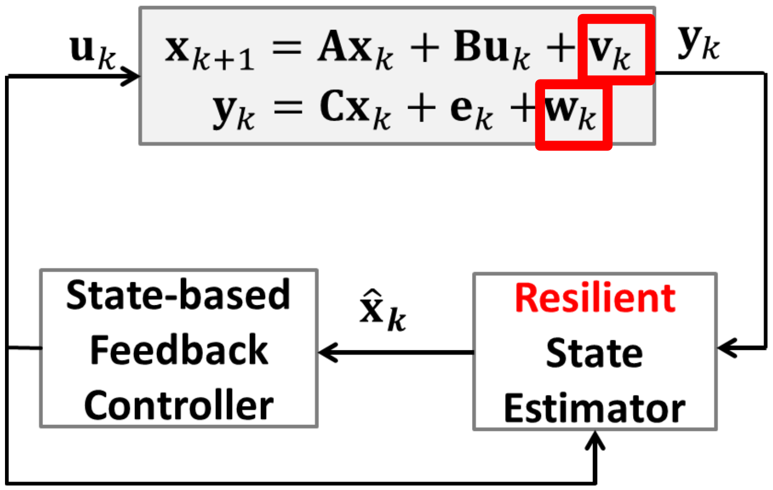
- Attacks on sensors in $\mathcal{K} = \{s_{i_1}, \ldots, s_{i_q}\} \subseteq \mathcal{S}$
  - modeled with attack vector $\mathbf{e}_k$
  - $\mathbf{e}_{k,i} \neq 0 \iff$ sensor s_i is under attack at time k

Attack-resilient State Estimation

Duke PRATT SCHOOL of ENGINEERING

- Attack-resilient control of Cyber-Physical Systems
  - Idea: Design attack-resilient state estimators

- Attack model
  - Goal: force the system into an unsafe state by creating a discrepancy between states and the estimates
  - Attacker has the ability to inject any signal using the compromised sensors
  - Attacker has full system knowledge and unlimited computational power



- Attacks on sensors in $\mathcal{K} = \{s_{i_1}, \ldots, s_{i_q}\} \subseteq \mathcal{S}$
  - modeled with attack vector $\mathbf{e}_k$
  - $\mathbf{e}_{k,i} \neq 0 \iff$ sensor s_i is under attack at time k

$$\mathcal{K} = \{s_2, s_5\}$$

$$\mathbf{e}_k = \begin{bmatrix} 0 \\ 1.7 \\ 0 \\ 0 \\ -9 \end{bmatrix}$$

Attack-Resilient State Estimation for Noisy Dynamical Systems

- Consider an initial state $\mathbf{x}_0$ and attack vectors from $\tilde{\mathbf{e}}$ $\quad \tilde{\mathbf{y}} = \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_p \end{bmatrix}, \tilde{\mathbf{e}} = \begin{bmatrix} \tilde{\mathbf{e}}_1 \\ \vdots \\ \tilde{\mathbf{e}}_p \end{bmatrix}, \tilde{\mathbf{w}} = \begin{bmatrix} \tilde{\mathbf{w}}_1 \\ \vdots \\ \tilde{\mathbf{w}}_p \end{bmatrix}, \quad \mathbf{O} = \begin{bmatrix} \mathbf{O}_1 \\ \vdots \\ \mathbf{O}_p \end{bmatrix}$

$$P_0: \quad \min_{\tilde{\mathbf{e}},\mathbf{x}} \|\tilde{\mathbf{e}}\|_{l_2,l_0} \quad s.t. \quad \tilde{\mathbf{y}} - \mathbf{O}\mathbf{x}_0 - \tilde{\mathbf{e}} = \mathbf{0}$$

$$P_{0,\omega}: \quad \min_{\tilde{\mathbf{e}},\mathbf{x}} \|\tilde{\mathbf{e}}\|_{l_2,l_0} \quad s.t. \quad \tilde{\mathbf{y}} - \mathbf{O}\,\mathbf{x}_0 - \tilde{\mathbf{e}} = \tilde{\mathbf{w}}, \quad \tilde{\mathbf{w}} \in \Omega$$

$$P_{1,\omega}: \quad \min_{\tilde{\mathbf{e}},\mathbf{x}} \|\tilde{\mathbf{e}}\|_{l_2,l_1} \quad s.t. \quad \tilde{\mathbf{y}} - \mathbf{O}\,\mathbf{x}_0 - \tilde{\mathbf{e}} = \tilde{\mathbf{w}}, \quad \tilde{\mathbf{w}} \in \Omega$$

- Goal: guarantees for $P_{0,\omega}$ and $P_{1,\omega}$ based estimators
  - Bounds on the state estimation errors
  - Sound attacked sensor identification

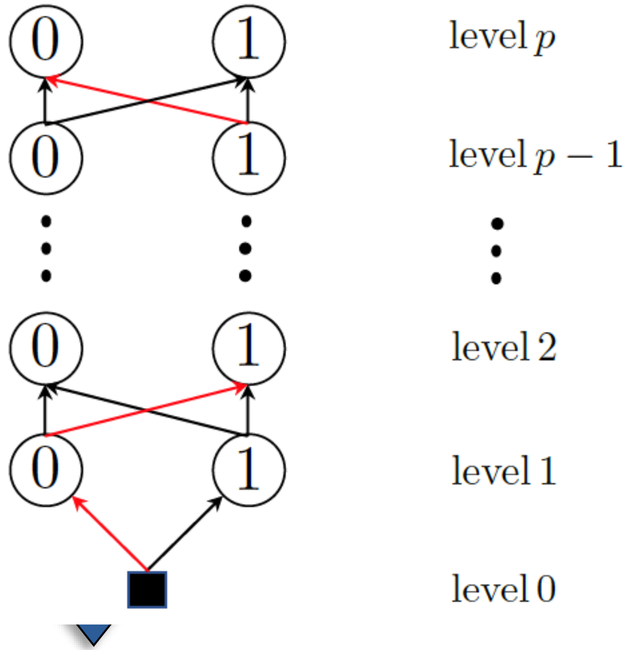
[ICCPS'14 – **Best paper award**, CDC15, IEEE CSM'17, IEEE TCNS'17]

Scalable and Optimal Graph-Search Method for RSE

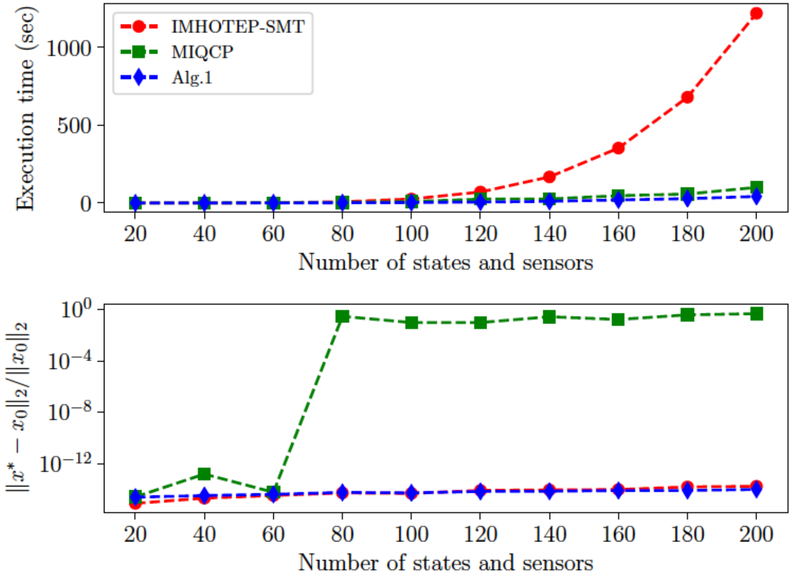
- Consider an initial state $\mathbf{x}_0$ and attack vectors from $\tilde{\mathbf{e}}$

$$P_0: \quad \min_{\tilde{\mathbf{e}},\mathbf{x}} \|\tilde{\mathbf{e}}\|_{l_2,l_0} \quad s.t. \quad \tilde{\mathbf{y}} - \mathbf{O}\mathbf{x}_0 - \tilde{\mathbf{e}} = \mathbf{0}$$

$$P_{0,\omega}: \quad \min_{\tilde{\mathbf{e}},\mathbf{x}} \|\tilde{\mathbf{e}}\|_{l_2,l_0} \quad s.t. \quad \tilde{\mathbf{y}} - \mathbf{O}\,\mathbf{x}_0 - \tilde{\mathbf{e}} = \tilde{\mathbf{w}}, \quad \tilde{\mathbf{w}} \in \Omega$$

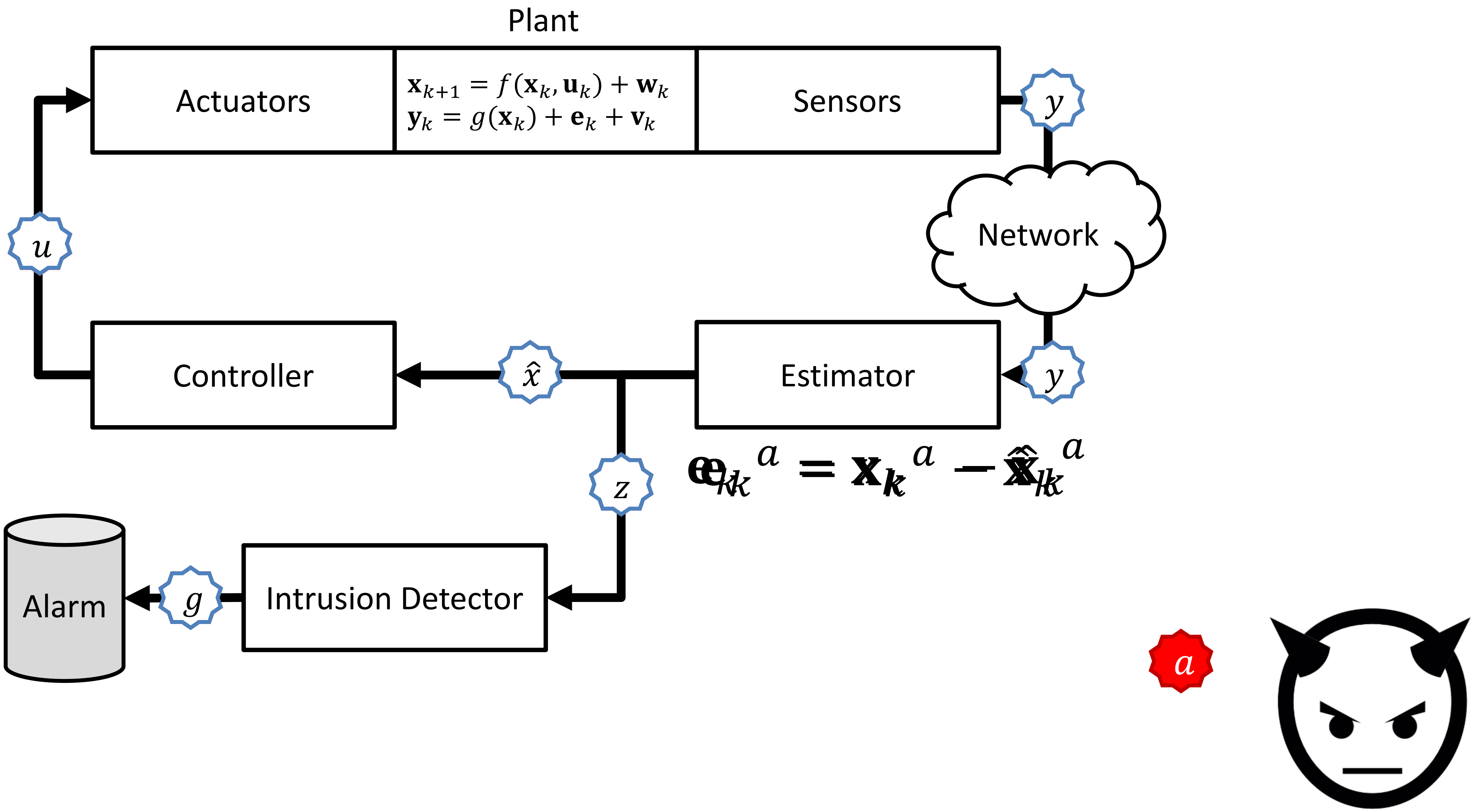


Graph capturing possible sensor attack assignments



X. Luo, M. Pajic, and M. Zavlanos, “A Scalable and Optimal Graph-Search Method for Secure State Estimation”, Automatica, submitted.

System Model With Attacks



Can Attacker Reach Any State?

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{a}_k + \boxed{\mathbf{v}_k}$$

$$supp(\mathbf{a}_k) = \mathcal{K}$$
$$\mathbf{a}_{k,i} = 0, \forall i \in \mathcal{K}^C$$

Theorem 1 [1,2,3,4*]:
A system presented above is perfectly attackable if and only if it is unstable, and at least one eigenvector **v** corresponding to an unstable mode satisfies $supp(\mathbf{C}\mathbf{v}) \subseteq \mathcal{K}$ and **v** is a reachable state of the dynamic system.

Physical detectors cannot always protect us from an intelligent attacker...

Can data authentication help?

[1] Y. Mo and B. Sinopoli, "*False data injection attacks in control systems,*" in First Workshop on Secure Control Systems, 2010

[2] C. Kwon, W. Liu, and I. Hwang, "*Analysis and design of stealthy cyber attacks on unmanned aerial systems*", Journal of Aerospace Information Systems, 1(8), 2014

[3] I. Jovanov and M. Pajic, "*Relaxing Integrity Requirements for Attack-Resilient Cyber-Physical Systems*", IEEE Trans. on Automatic Control, 2019

[4] Amir Khazraei, Miroslav Pajic, "*Perfect Attackability of Linear Dynamical Systems with Bounded Noise,*" ACC, submitted.

Can Attacker Reach Any State?

Duke PRATT SCHOOL of ENGINEERING

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{a}_k + \mathbf{v}_k$$

$$supp(\mathbf{a}_k) = \mathcal{K}$$
$$\mathbf{a}_{k,i} = 0, \forall i \in \mathcal{K}^C$$

Theorem 1 [1,2,3,4*]:
A system presented above is perfectly attackable if and only if it is unstable, and at least one eigenvector **v** corresponding to an unstable mode satisfies $supp(\mathbf{C}\mathbf{v}) \subseteq \mathcal{K}$ and **v** is a reachable state of the dynamic system.

Theorem: A system Σ with a global data integrity police $\mu(L)$ is not perfectly attackable.

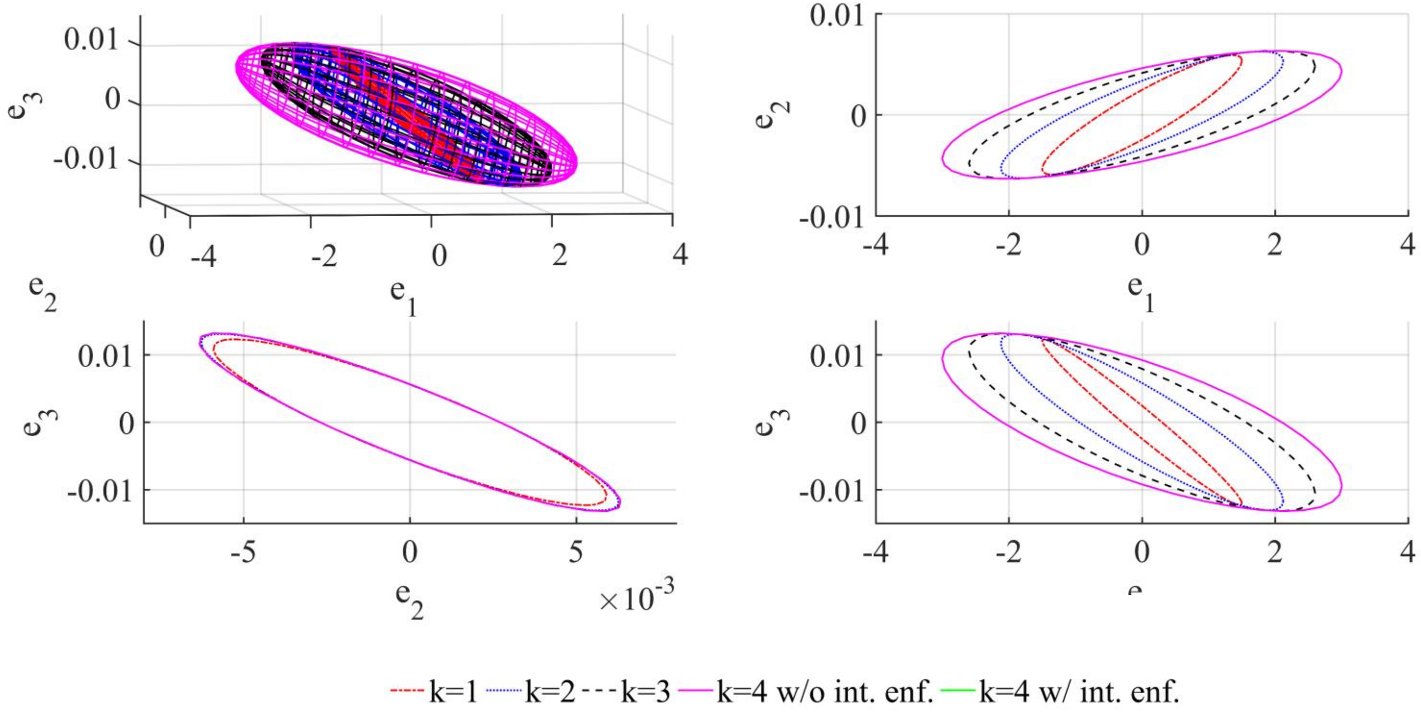
State Estimation Error
In the Presence of Stealthy Attacks

Reachable region of the state estimation error under attack [1,2,3]

$$\mathcal{R}[k] = \left\{ \boldsymbol{e} \in \mathbb{R}^{\boldsymbol{n}} \middle| \begin{array}{c} \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}} \preccurlyeq E\left[\boldsymbol{e}^a[k]\right]E\left[\boldsymbol{e}^a[k]\right]^{\mathrm{T}} + \gamma Cov(\boldsymbol{e}_k^a) \\ \boldsymbol{e}^a[k] = \boldsymbol{e}_k^a(\mathbf{a}_{1\ldots k}), \mathbf{a}_{1\ldots k} \in \mathcal{A}_k \end{array} \right\}$$

$\mathbf{a}_{1\ldots k} = [\mathbf{a}[1]^{\mathrm{T}} \ldots \mathbf{a}[k]^{\mathrm{T}}]^{\mathrm{T}}$
$\mathcal{A}_k$ is the set of all stealthy attacks

$\boldsymbol{e}_k^a(\mathbf{a}_{1\ldots k})$ is the estimation error evolution due to attack $\mathbf{a}_{1\ldots k}$

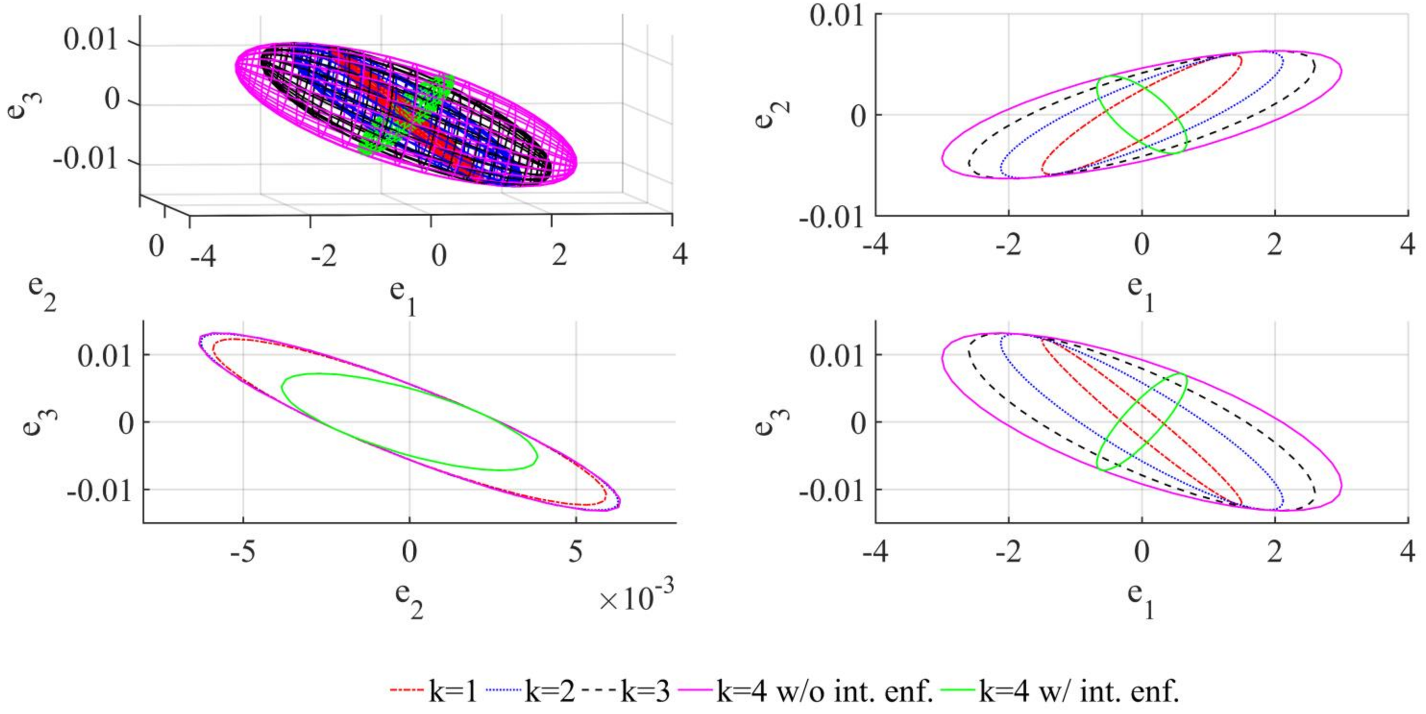


Integrity Enforcement Policy

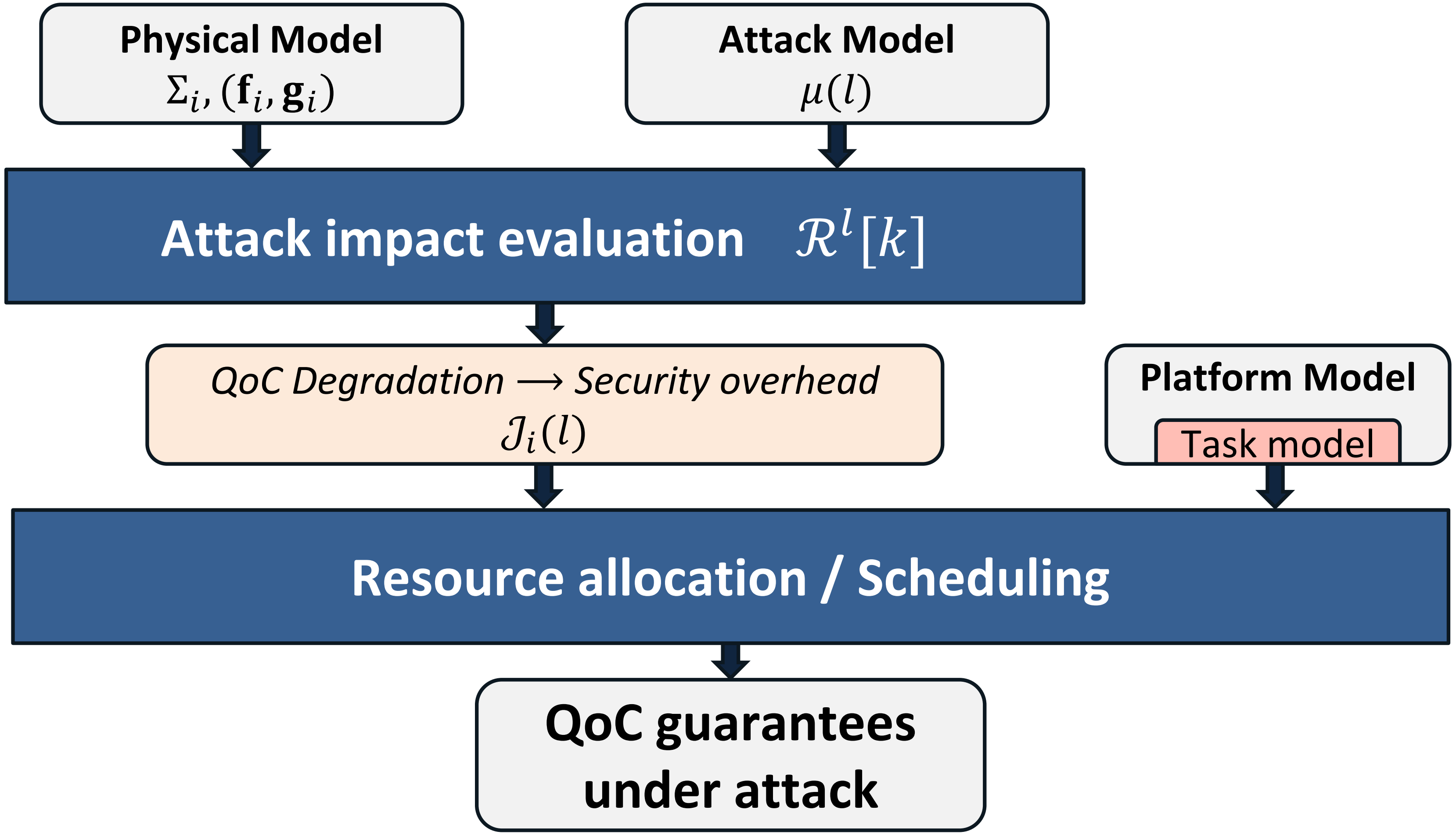
Integrity enforcement policy ensures attacker's influence is zeroed at enforcement points

Data integrity enforcement policy (μ, l) where $\mu = \{t_k\}_{k=0}^{\infty}$, with $t_{k-1} < t_k, \forall k > 0$
and $l = \sup_{k>0} t_k - t_{k-1}$ ensures that $\mathbf{a}_{1\ldots k} = 0, \forall k \geq 0$

This means that at points of authentication $\boldsymbol{y}_i^{net,a}[k] = \boldsymbol{y}_i^{a}[k]$



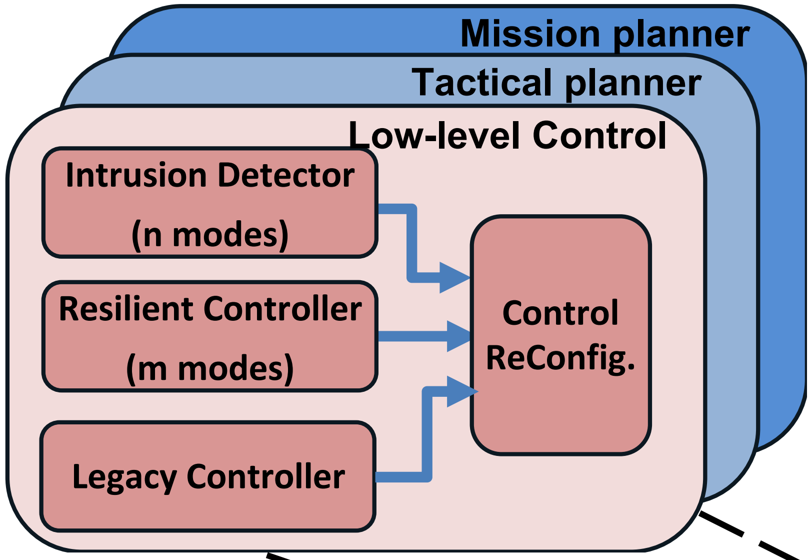
Security-Aware Design Framework



Platform-aware Execution/Integration of Cyber-Physical Security Components

Duke PRATT SCHOOL of ENGINEERING

Control view

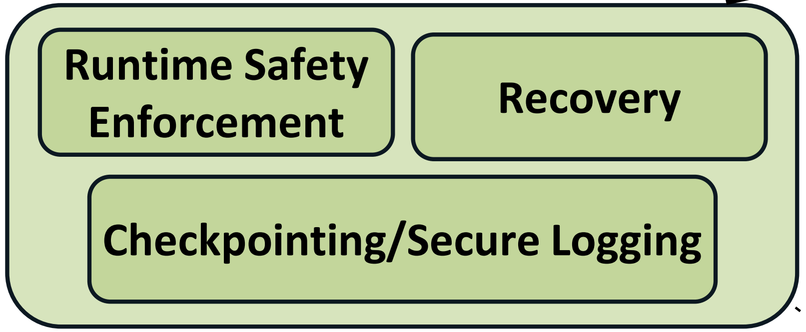


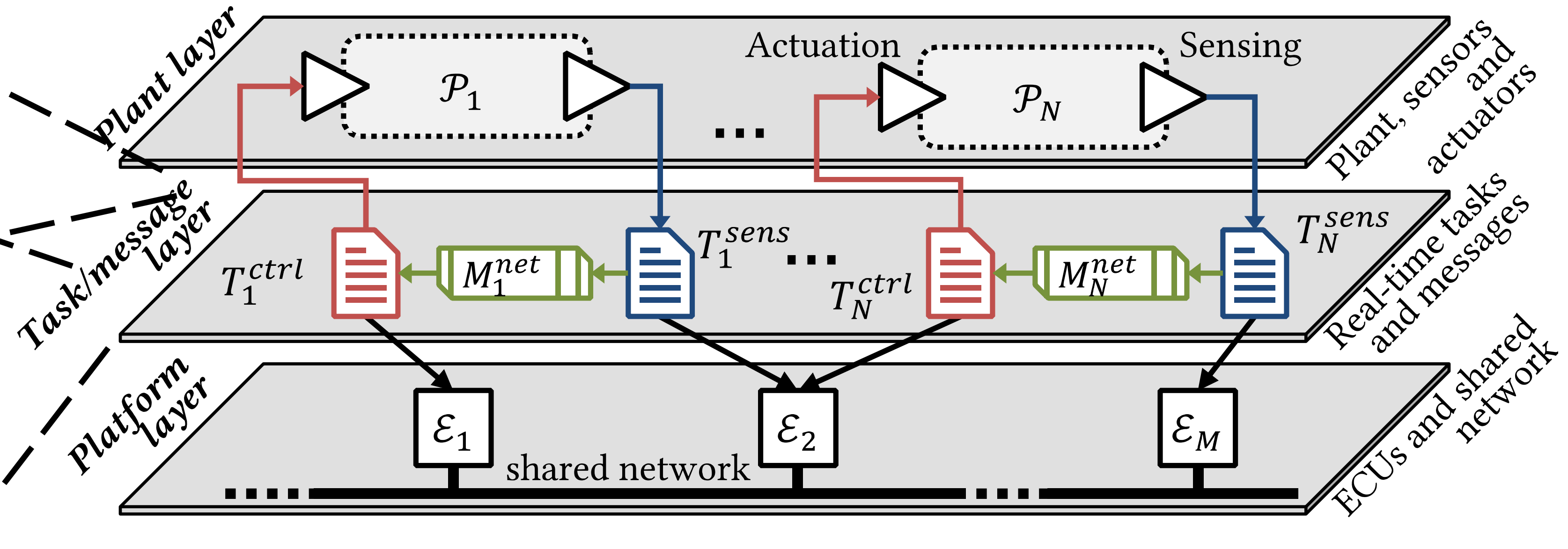
Constrained computation and communication resources limit the full use of developed cyber-physical techniques

Our Goal: Provide quantitative tradeoff procedure to map security-aware modules onto available architecture

[CMS17, TECS/EMSOFT17, RTSS17, TCPS*19]

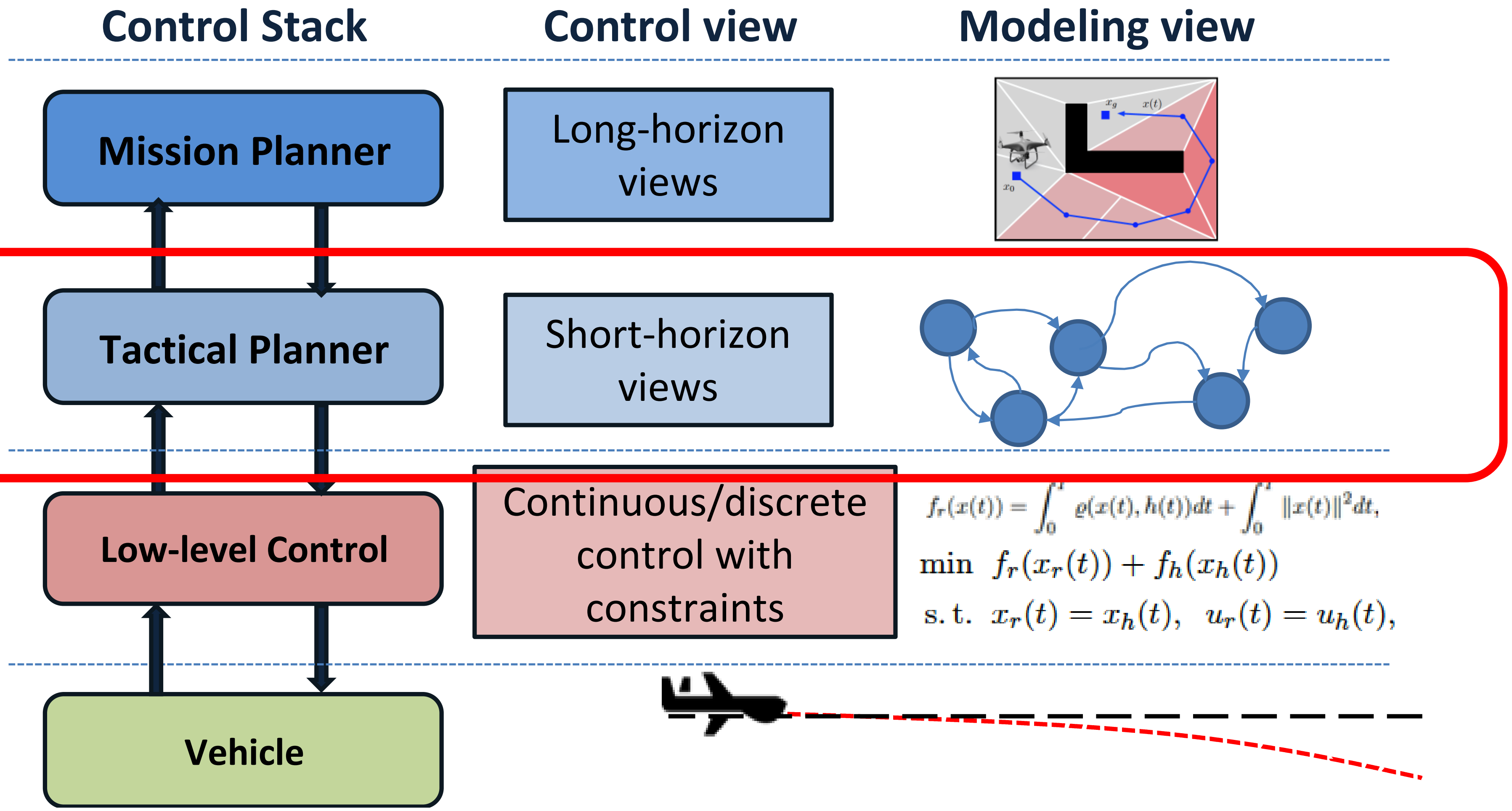
Runtime/platform support





Security-Aware Control for Autonomous Systems

Duke
PRATT SCHOOL of ENGINEERING



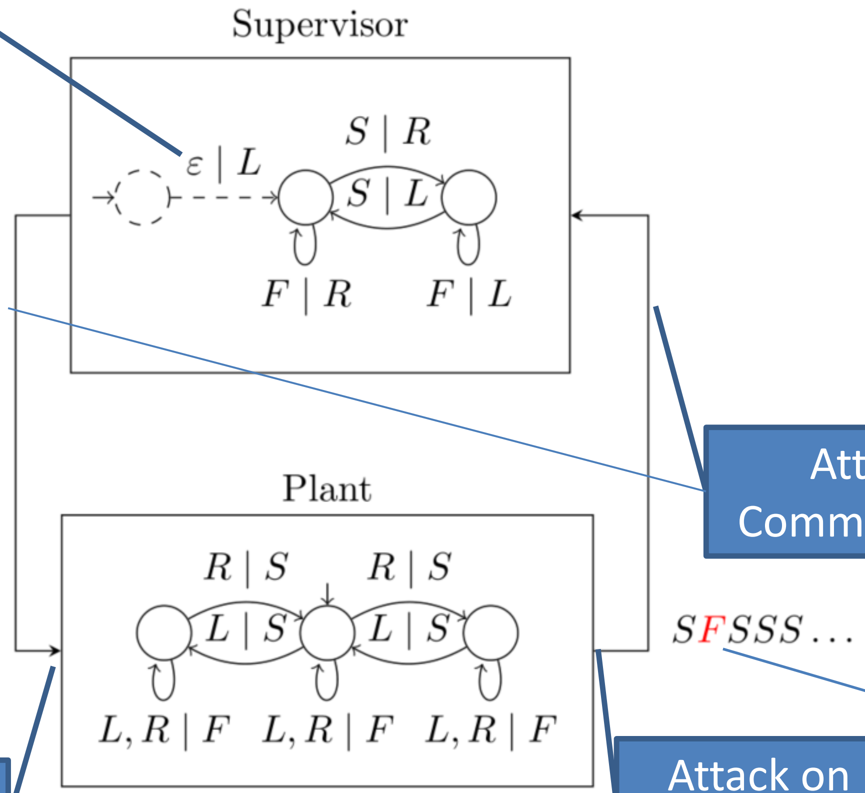
Our Goal: Add resiliency to controls across different/all levels of control stack

Security From A Supervisory Control PerspectivE

Duke PRATT SCHOOL of ENGINEERING

On the higher level, CPS is abstracted by discrete event systems, namely, finite state models driven by discrete events.

Empty symbol to start



Idea: Model Attacks as ***Finite State Transducers (FSTs)***

Violation if an attacker removes L

Input: $LRLR$ ...

Attack on Communications

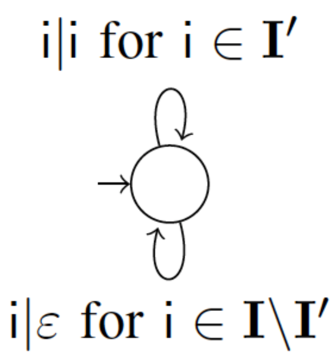
Attack on Actuators

Attack on Sensors

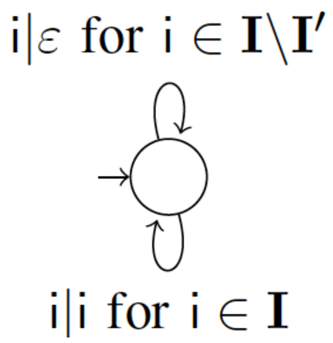
Violation if F is replaced by S

Desired Model $\mathcal{D} \subseteq \mathcal{P}$ is **controllable** without attacks

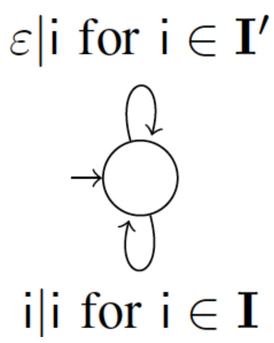
Using FSTs to Model Attacks



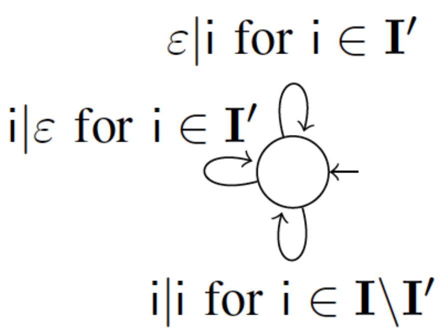
Projection Attack



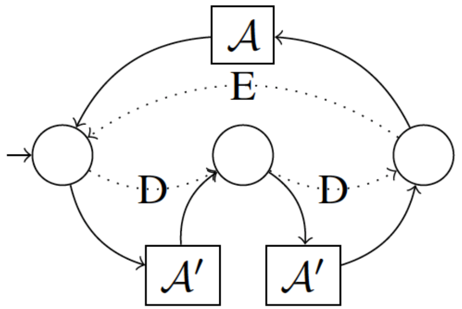
Deletion (DoS) Attack



Data Injection Attack

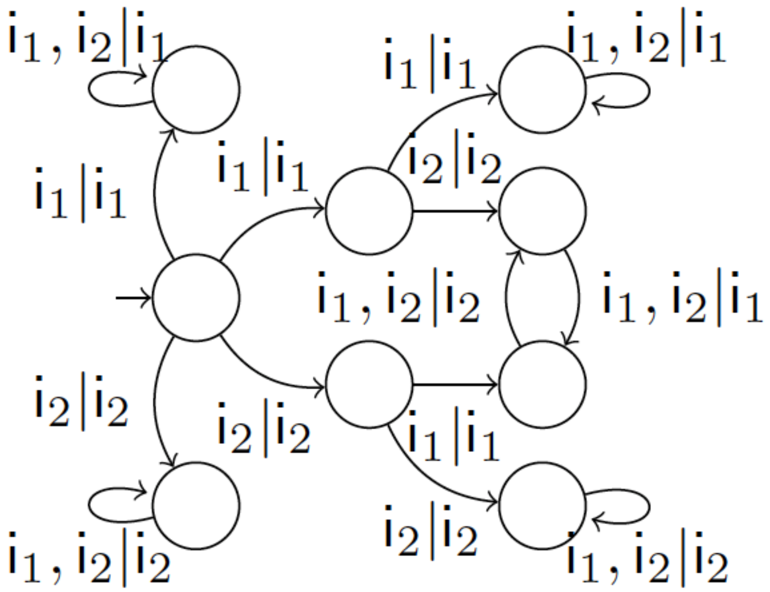


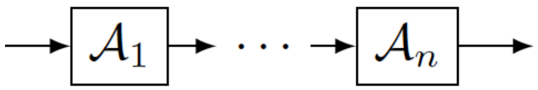
Injection-Removal Attack



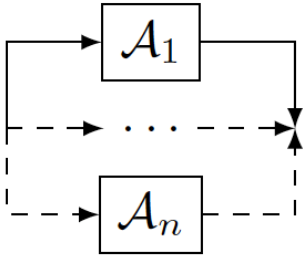
Modeling constraints on attacker

Replay Attack





(a) Serial Composition

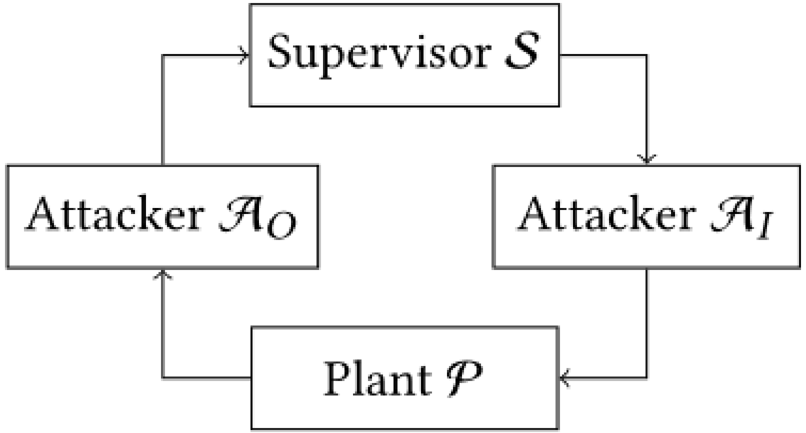


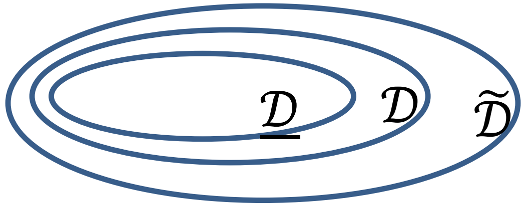
(b) Parallel Composition

1. Attacks usually have patterns.
2. All possible attacks captured with nondeterminism
3. FST models can be built from partial information on the attackers to overapproximate.
4. Attack models even unknown, may be inferred from executions.

Attack-Resiliency <=> Controllability Under Attacks

Not any desired model $\mathcal{D}$ is controllable!





Model subtraction $C = A \backslash B$ if $C \subseteq A$ and B, C share no common I/O sequences.

Controllability Theorem: For desired Model $\mathcal{D} \subseteq \mathcal{P}$

1. The minimal controllable model containing $\mathcal{D}$ is

$$\widetilde{\mathcal{D}} = \mathcal{A}_I^{-1} \circ \mathcal{A}_I \circ \mathcal{D}$$

achieved by the supervisor when observable

$$\mathcal{S} = \mathcal{A}_O^{-1} \circ \mathcal{D} \circ \mathcal{A}_I^{-1}.$$

2. The maximal controllable model contained in $\mathcal{D}$ is

$$\underline{\mathcal{D}} = \mathcal{D} \setminus \mathcal{A}_I^{-1} \circ \mathcal{A}_I \circ \left(\left(\mathcal{A}_I^{-1} \circ \mathcal{A}_I \right)^{\infty} \circ \mathcal{D} \right) \setminus \mathcal{D}),$$

achieved by the supervisor when observable

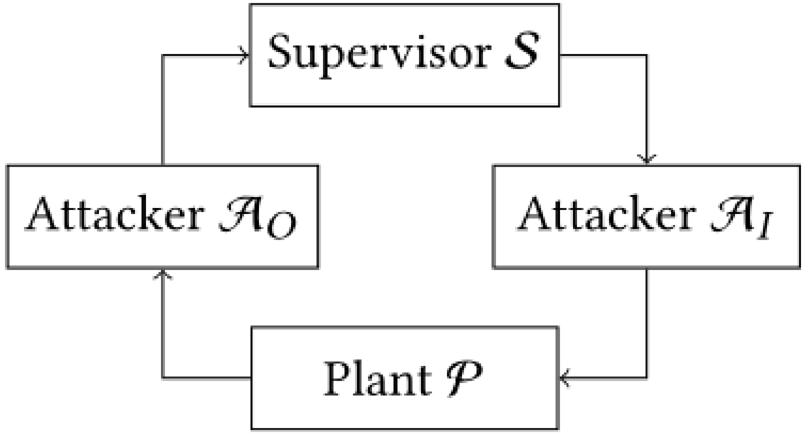
$$\mathcal{S} = \mathcal{A}_O^{-1} \circ \underline{\mathcal{D}} \circ \mathcal{A}_I^{-1}.$$

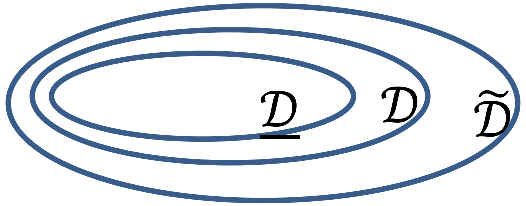
The desired model is controllable if and only if $\underline{\mathcal{D}} = \widetilde{\mathcal{D}} = \mathcal{D}$.

Y. Wang, A. Bozkurt, and M. Pajic, "Attack-Resilient Supervisory Control of Discrete Event Systems", *IEEE Transactions on Automatic Control*, submitted.
Z. Jakovljevic, V. Lesi, and M. Pajic, "Attacks on Distributed Sequential Control in Manufacturing Automation", *IEEE Transactions on Industrial Informatics*, submitted.
M. Elfar, Y. Wang, and M. Pajic, "Security-Aware Synthesis using Delayed Action Games", 31st CAV, 2019, submitted.
Y. Wang and M. Pajic, "Supervisory Control of Discrete Event Systems in the Presence of Sensor and Actuator Attacks", IEEE CDC, 2019.
Y. Wang and M. Pajic, " Attack-Resilient Supervisory Control with Intermittent Authentication", IEEE CDC, 2019.
V. Lesi, Z. Jakovljevic and M. Pajic, "Reliable Industrial IoT-Based Distributed Automation", 4th ACM/IEEE IoTDI, 2019.

Attack-Resiliency <=> Controllability Under Attacks

Not any desired model $\mathcal{D}$ is controllable!





Model subtraction $C = A \backslash B$ if $C \subseteq A$ and B, C share no common I/O sequences.

Controllability Theorem: For desired Model $\mathcal{D} \subseteq \mathcal{P}$

1. The minimal controllable model containing $\mathcal{D}$ is

$$\widetilde{\mathcal{D}} = \mathcal{A}_I^{-1} \circ \mathcal{A}_I \circ \mathcal{D}$$

achieved by the supervisor when observable

$$\mathcal{S} = \mathcal{A}_O^{-1} \circ \mathcal{D} \circ \mathcal{A}_I^{-1}.$$

2. The maximal controllable model contained in $\mathcal{D}$ is

$$\underline{\mathcal{D}} = \mathcal{D} \setminus \mathcal{A}_I^{-1} \circ \mathcal{A}_I \circ \left(\left(\mathcal{A}_I^{-1} \circ \mathcal{A}_I \right)^{\infty} \circ \mathcal{D} \right) \setminus \mathcal{D}),$$

achieved by the supervisor when observable

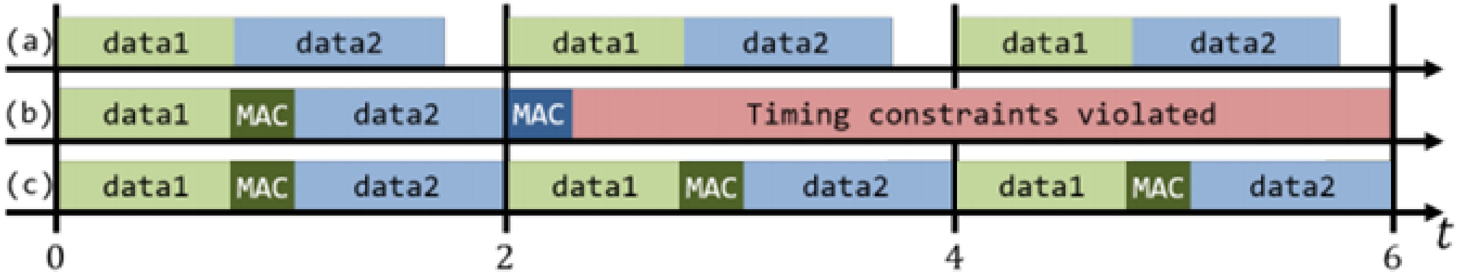
$$\mathcal{S} = \mathcal{A}_O^{-1} \circ \underline{\mathcal{D}} \circ \mathcal{A}_I^{-1}.$$

The desired model is controllable if and only if $\underline{\mathcal{D}} = \widetilde{\mathcal{D}} = \mathcal{D}$.

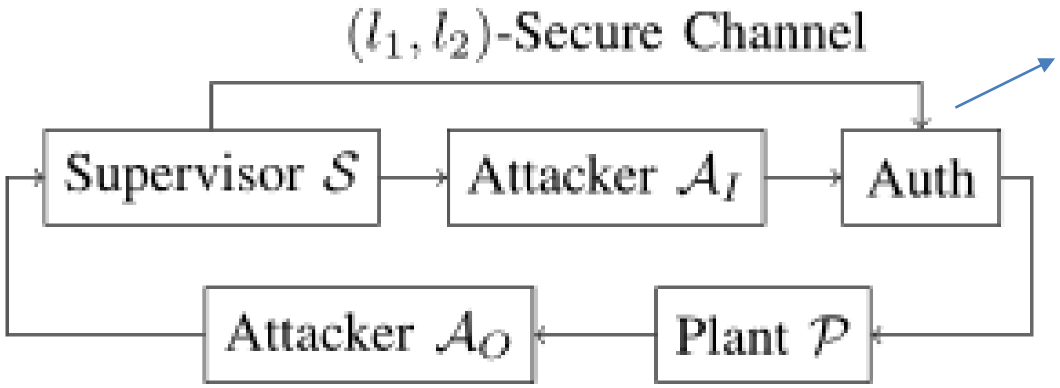
Toolbox: *ARSC* for Synthesis of Attack-Resilient Supervisory Control

Modeling Intermittent Authentication in DES

- Activated by supervisor when necessary
- Not consecutively
- **Transmit anchoring word** $\leq l_1$ **and recovering word** $\leq l_2$



$(i_5, i_2 i_3 i_4)$



Can only accept or repair symbols

Received: $i_1 i_5 i_1 i_2$ ...

Recovered: $i_1 i_2 i_3 i_4 i_1 i_2$ ...

Want: $i_1 i_2 i_3 i_4 i_1 i_2$ ...

Send attack-resilient: $i_1 i_5 i_1 i_2$ ...

[CDC19b]

Resiliency with Intermittent Authentication

(l_1, l_2)-**Accessibility:** For models $N \subseteq M$, M is (l_1, l_2)-accessible from N if

1. The graph subtraction M/N is a tree, with longest path $\leq l_2$.
2. For any such path, there is a path $\leq l_1$ with same start and end in the graph of N

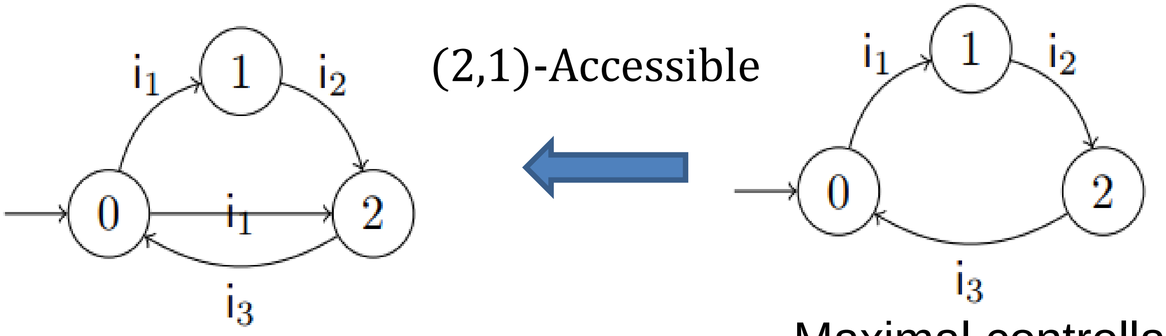
Want: $i_1 i_3 i_1 i_2 i_3$ ...

Send attack-resilient word: $i_1 i_2 i_3 i_1 i_2 i_3$ ...

$(i_1 i_2, i_1)$

Received: $i_1 i_2 i_3 i_1 i_2 i_3$ ...

Recovered: $i_1 i_3 i_1 i_2 i_3$ ...



Desired

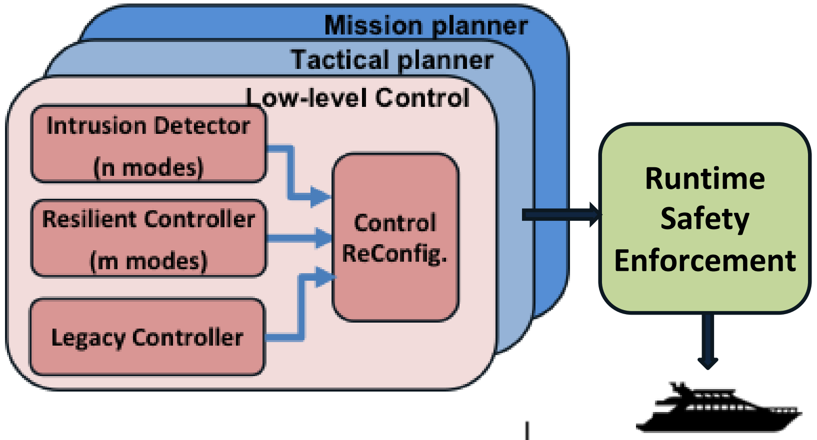
Maximal controllable sub-model

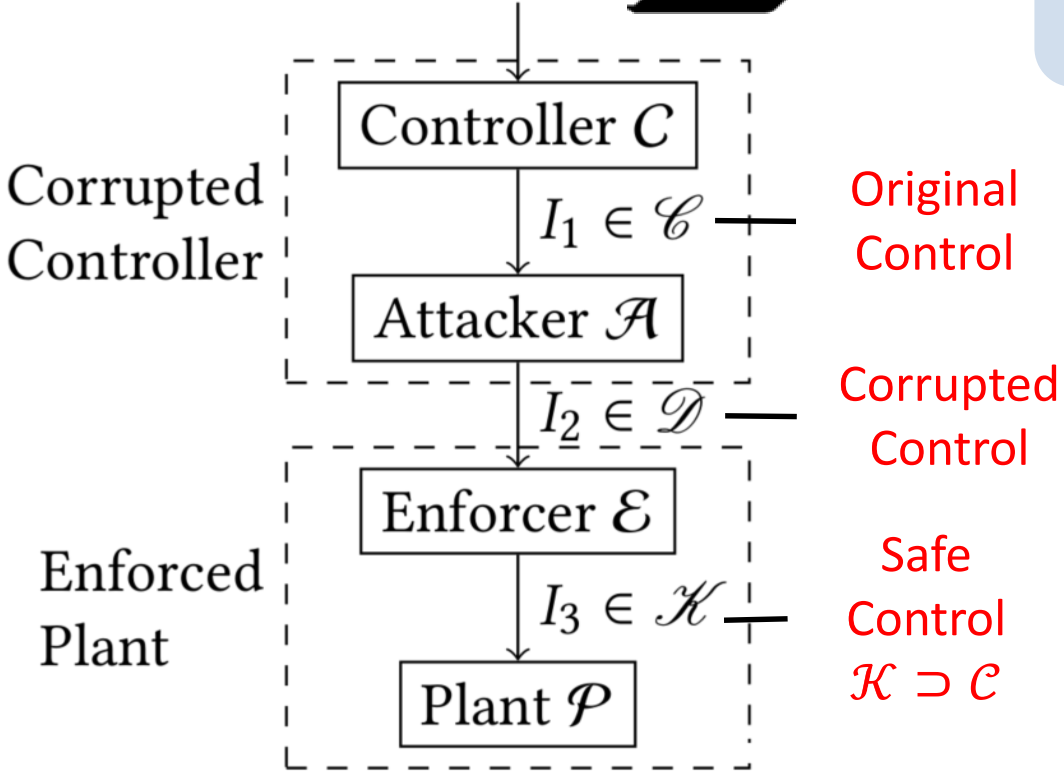
Controllability Theorem with Intermittent Authentication : The the desired model $\mathcal{D}$ is controllable if and only if it is (l_1, l_2)-accessible from $\underline{\mathcal{D}}$.

[CDC19b]

Real-Time Enforcement of Regular Specifications

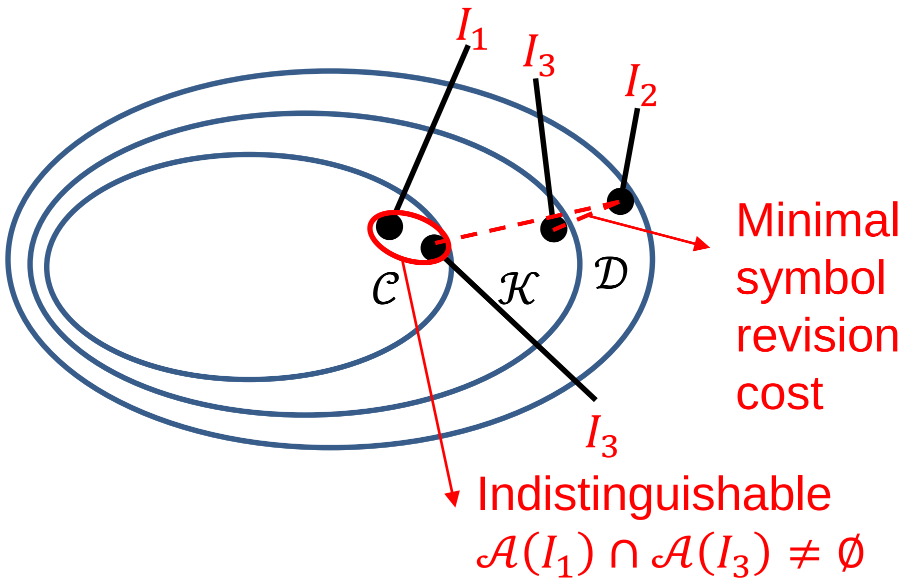
Assuring safe control execution in the age of AI





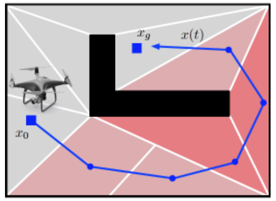
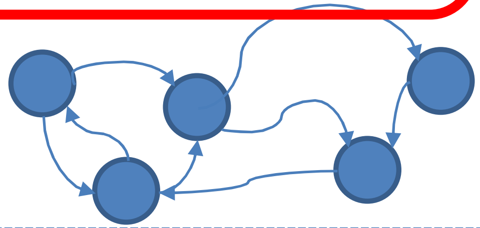
Challenge 1: Given the set of possible corrupted controls $\mathcal{D}$, how to revise any corrupted control $I_2 \in \mathcal{D}$ with minimal cost to some safe control $I_3 \in \mathcal{K}$

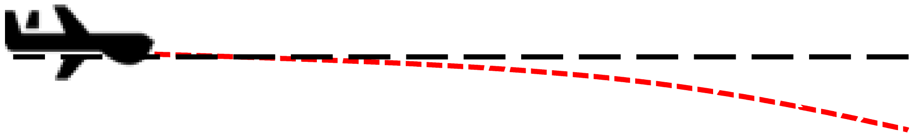
Challenge 2: Given the attack model $\mathcal{A}$, how to repair any corrupted control $I_2 \in \mathcal{D}$ with minimal cost to some control $I_3 \in \mathcal{K}$ that is indistinguishable from I_1

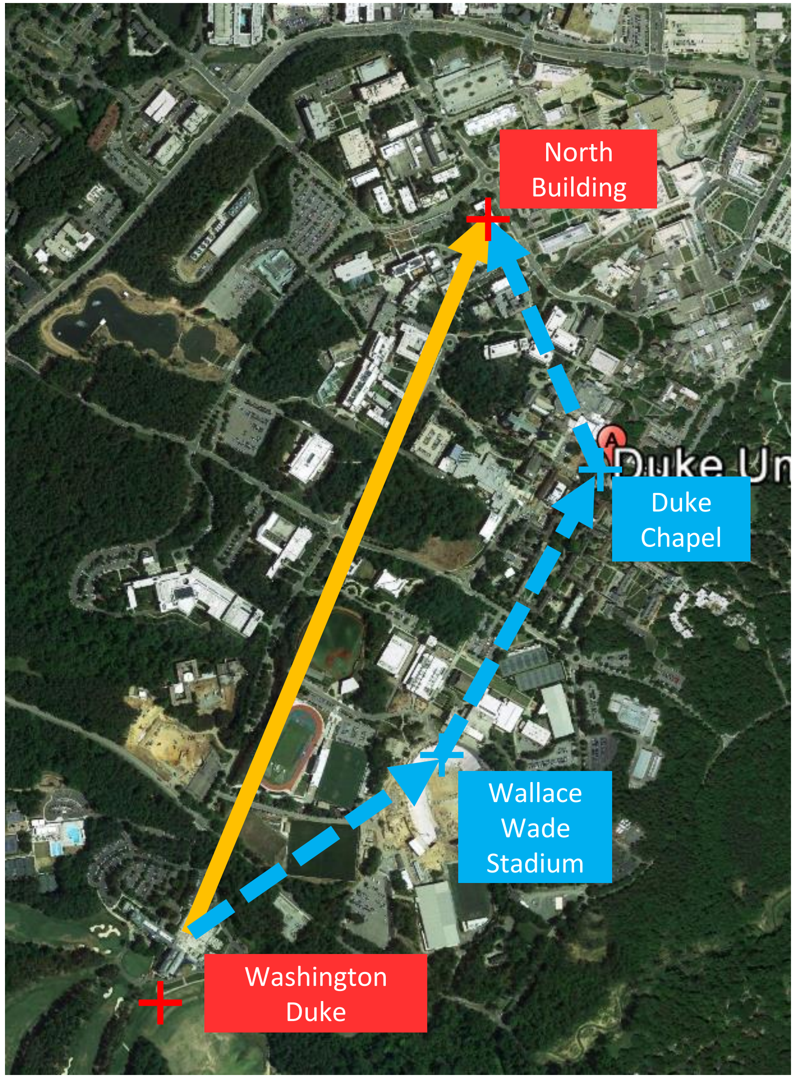


Security-Aware Control for Autonomous Systems

Duke PRATT SCHOOL of ENGINEERING

Control Stack	Control view	Modeling view
Mission Planner	Long-horizon views	
Tactical Planner	Short-horizon views	
Low-level Control	Continuous/discrete control with constraints	$f_r(x(t)) = \int_0^T g(x(t), h(t))dt + \int_0^T \|x(t)\|^2 dt,$ $\min \; f_r(x_r(t)) + f_h(x_h(t))$ $\text{s.t.} \; x_r(t) = x_h(t), \; u_r(t) = u_h(t),$
Vehicle		

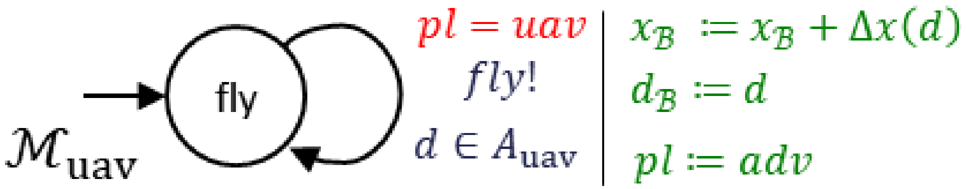




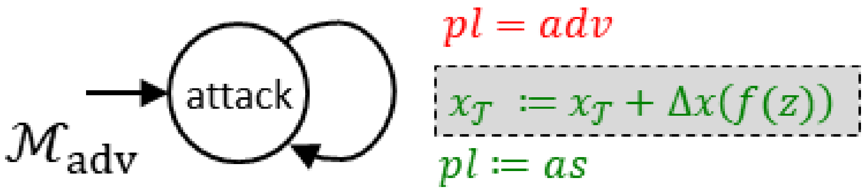
DAG | Hidden-Information Semantics

Duke PRATT SCHOOL of ENGINEERING

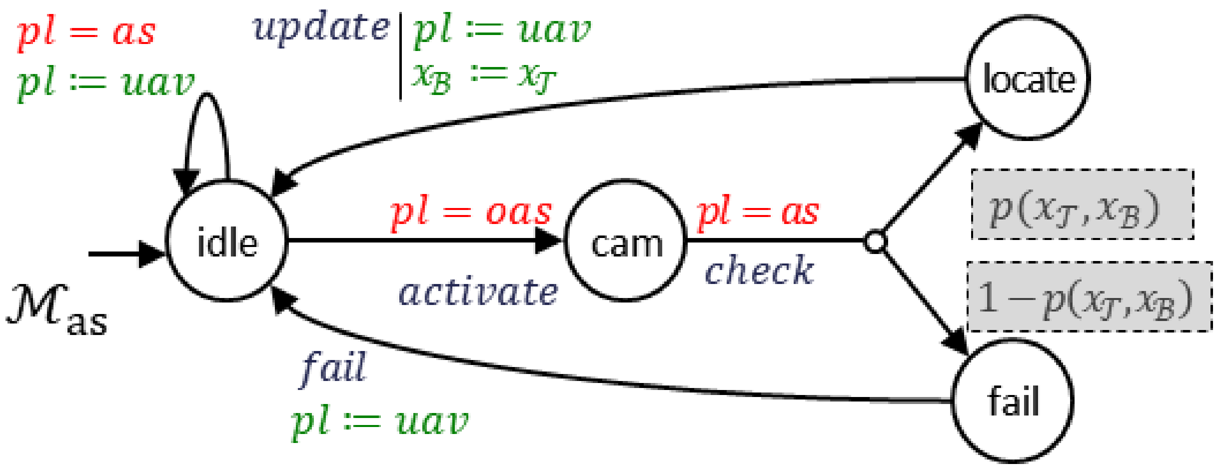
UAV Model



Adversary Model



Advisory System Model

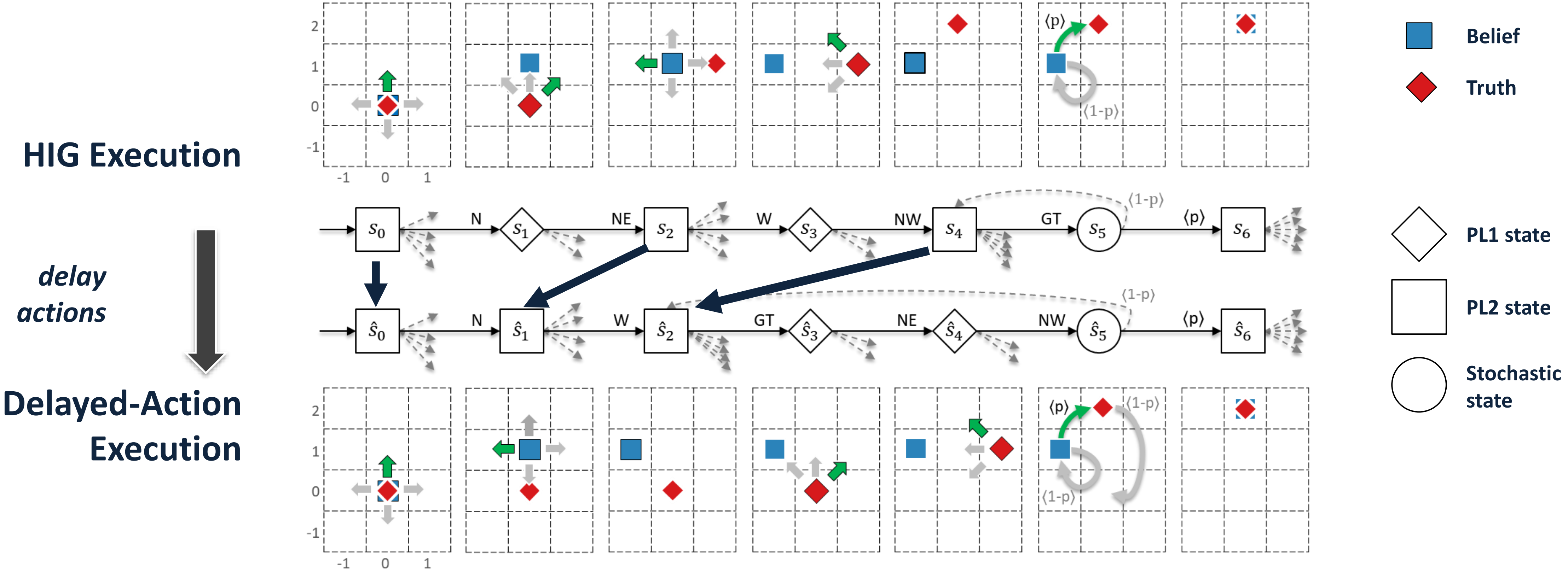


Information inside this box is oftentimes unknown, i.e., **hidden**

Off-the-shelf model checkers do NOT support hidden variables

Strategies CANNOT be synthesized based on hidden information

Approach: Delaying Actions



→ **Information is hidden from one player (H-UAV) by delaying the actions of the other player (ADV)**

Delayed-Action Games (DAGs)

Definition (Delayed-Action Game).

A DAG of an HIG $\mathcal{G}_{\mathsf{H}} = \langle S, (S_{\mathrm{I}}, S_{\mathrm{II}}, S_{\bigcirc}), A, s_0, \beta, \delta \rangle$, — Is based on an HIG

with players $\Gamma = \{\mathrm{I}, \mathrm{II}, \bigcirc\}$

over a set of variables $V = \{v_{\mathcal{T}}, v_{\mathcal{B}}\}$ — Truth and Belief

is a tuple $\mathcal{G}_{\mathsf{D}} = \langle \hat{S}, (\hat{S}_{\mathrm{I}}, \hat{S}_{\mathrm{II}}, \hat{S}_{\bigcirc}), A, \hat{s}_0, \beta, \hat{\delta} \rangle$ *where*

- $\hat{S} \subseteq Ev(v_{\mathcal{T}}) \times Ev(v_{\mathcal{B}}) \times A^*_{\mathrm{II}} \times \mathbb{N}_0 \times \Gamma$
  partitioned into $\hat{S}_{\mathrm{I}}, \hat{S}_{\mathrm{II}}$ *and* $\hat{S}_{\bigcirc}$;
- $\hat{s}_0 \in \hat{S}_{\mathrm{II}}$ *is the initial state;* — Always starts with PL2
- $\hat{\delta} \colon \hat{S} \times A \times \hat{S} \to [0,1]$ *is a transition function s.t.*
  $\hat{\delta}(\hat{s}_{\mathrm{II}}, a, \hat{s}_{\bigcirc}) = \hat{\delta}(\hat{s}_{\mathrm{I}}, a, \hat{s}_{\mathrm{II}}) = \hat{\delta}(\hat{s}_{\bigcirc}, a, \hat{s}_{\mathrm{I}}) = 0,$ *and*
  $\hat{\delta}(\hat{s}_{\mathrm{II}}, a, \hat{s}_{\mathrm{II}}), \hat{\delta}(\hat{s}_{\mathrm{I}}, a, \hat{s}_{\mathrm{I}}), \hat{\delta}(\hat{s}_{\mathrm{I}}, a, \hat{s}_{\bigcirc}) \in \{0,1\},$ — Specific order for players
  $\hat{\delta}(\hat{s}_{\mathrm{II}}, \theta, \hat{s}_{\mathrm{I}}) \in \{0,1\},$ — PL2 to PL1 through special action θ
  for all $\hat{s}_{\mathrm{I}} \in \hat{S}_{\mathrm{I}}$, $\hat{s}_{\mathrm{II}} \in \hat{S}_{\mathrm{II}}$, $\hat{s}_{\bigcirc} \in \hat{S}_{\bigcirc}$ *and* $a \in A$,
  where $\sum_{\hat{s}' \in \hat{S}_{\mathrm{II}}} \delta(\hat{s}_{\bigcirc}, a, \hat{s}') = 1.$

DAG Properties

- DAG-HIG simulation relation

Definition 9 (Game Proper Simulation). *A game $\mathcal{G}_{\mathsf{D}}$ properly simulates $\mathcal{G}_{\mathsf{H}}$, denoted by $\mathcal{G}_{\mathsf{D}} \rightsquigarrow \mathcal{G}_{\mathsf{H}}$, iff $\forall \varrho \in \mathrm{Prop}(\mathcal{G}_{\mathsf{H}})$, $\exists \hat{\varrho} \in \mathrm{Prop}(\mathcal{G}_{\mathsf{D}})$ such that $\varrho \sim \hat{\varrho}$.*

Theorem 1 (Probabilistic Simulation). *For any $s_0 \simeq \hat{s}_0$ and $\varrho \in \mathrm{Prop}(\mathcal{G}_{\mathsf{H}})$ where $first(\varrho) = s_0$, it holds that*

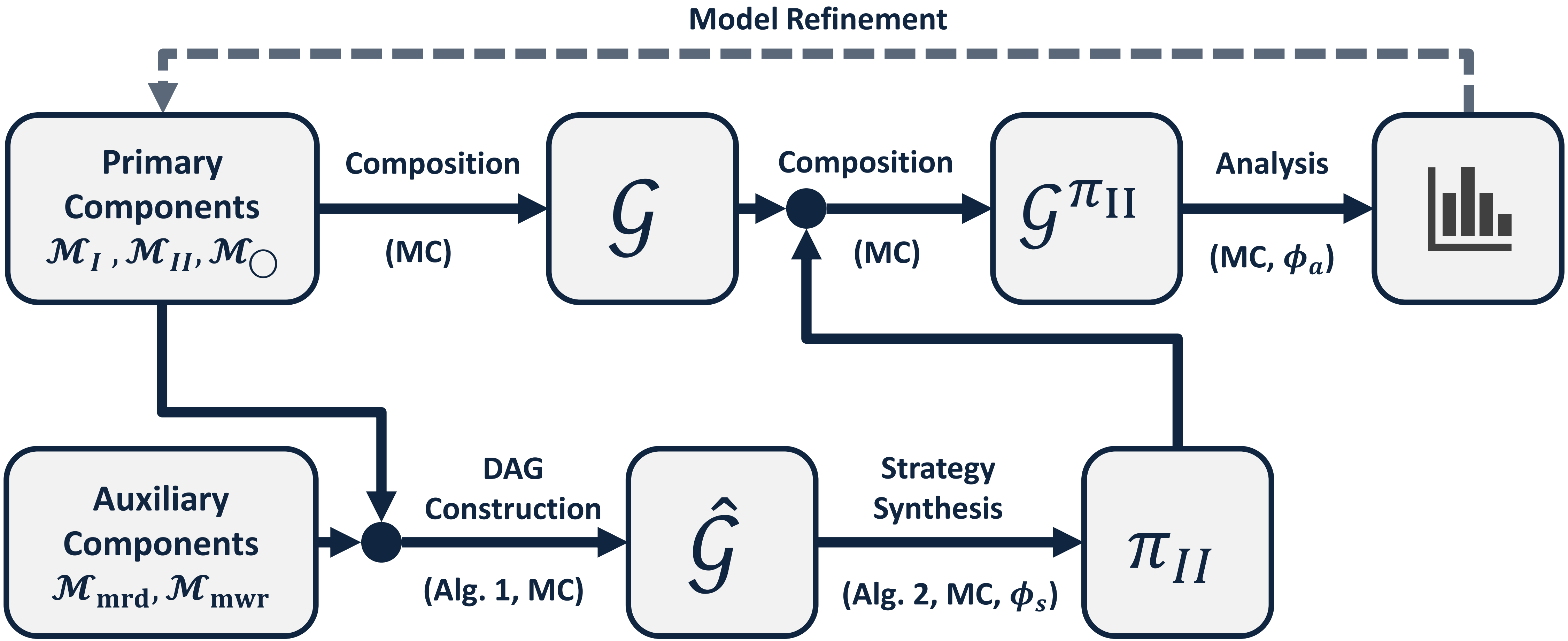
$$\Pr\left[last(\varrho) = s'\right] = \Pr\left[\left(\overline{move(\varrho)}\right)(\hat{s}_0) = \hat{s}'\right] \quad \forall s', \hat{s}' \ \ s.t. \ \ s' \simeq \hat{s}'.$$

Theorem 2 (DAG-HIG Simulation). *For any HIG $\mathcal{G}_{\mathsf{H}}$ there exists a DAG $\mathcal{G}_{\mathsf{D}} = \mathfrak{D}[\mathcal{G}_{\mathsf{H}}]$ such that $\mathcal{G}_{\mathsf{D}} \rightsquigarrow \mathcal{G}_{\mathsf{H}}$ (as defined in Def. 9).*

- DAG decomposition

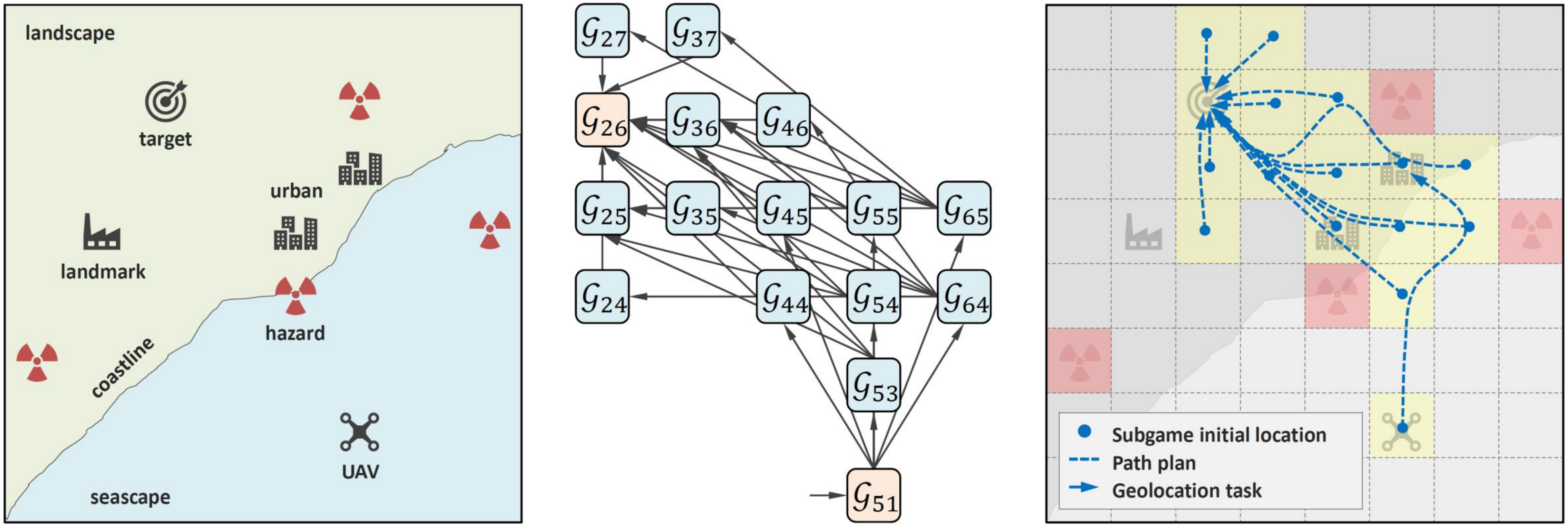
Definition 10 (DAG Subgames). *The* subgames *of a $\mathcal{G}_{\mathsf{D}}$ are defined by the set $\left\{\hat{\mathcal{G}}_i \,\middle|\, \hat{\mathcal{G}}_i = \left\langle \hat{S}^{(i)}, (\hat{S}_{\mathrm{I}}^{(i)}, \hat{S}_{\mathrm{II}}^{(i)}, \hat{S}_{\bigcirc}^{(i)}), A, \hat{s}_0^{(i)}, \hat{\delta}^{(i)} \right\rangle, i \in \mathbb{N}_0\right\}$, where $\hat{S} = \bigcup_i \hat{S}^{(i)}$; $\hat{S}_\gamma = \bigcup_i \hat{S}_\gamma^{(i)} \ \forall \gamma \in \Gamma$; and $\hat{s}_0^{(i)} = \hat{s}_{\mathrm{II}}^{(i)}$ s.t. $\hat{s}_{\mathrm{II}}^{(i)} \in \mathrm{Prop}(\mathcal{G}_{\mathsf{D}}^{(i)})$, $\hat{s}_{\mathrm{II}}^{(i)} \neq \hat{s}_{\mathrm{II}}^{(j)} \ \forall i, j \in \mathbb{N}_0$.*

DAG-Based Synthesis



MC: Model Checker
ϕ_s : Synthesis query
ϕ_a : Analysis query

Case Study

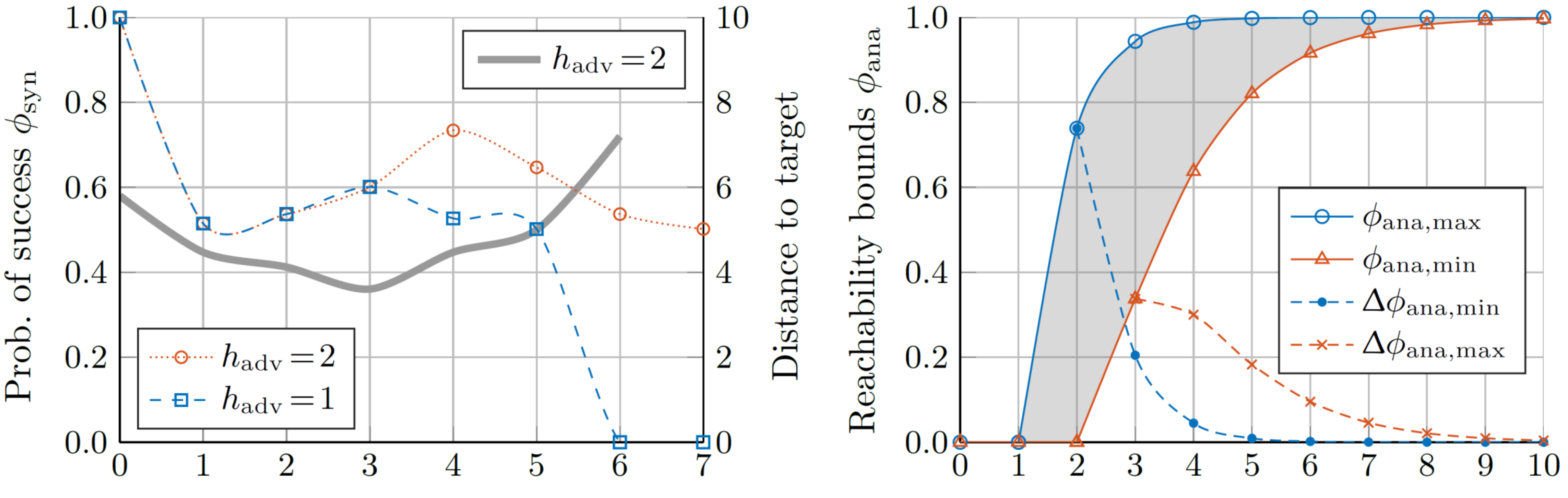


(a) Environment setup. (b) Supergame $\mathcal{G}_D$. (c) Protocols.

- Model Checker: PRISM-games
  - Kwiatkowska, M., Parker, D. and Wiltsche, C., 2018. PRISM-games: verification and strategy synthesis for stochastic multi-player games with multiple objectives. *International Journal on Software Tools for Technology Transfer*, *20*(2), pp.195-210.

Case Study

- Analysis



(a) Geolocation task at stage h

(b) Max. no. of geolocation tasks n

Security-Aware Human-on-the-Loop Protocols

How can we use human context awareness (in real-time) for security?

Duke
PRATT SCHOOL of ENGINEERING

Operator

- Set goals
- Supervise mission
- Imagery tasks

Autonomy/automation

- Target assignment
- Trajectory planning
- Attack detection

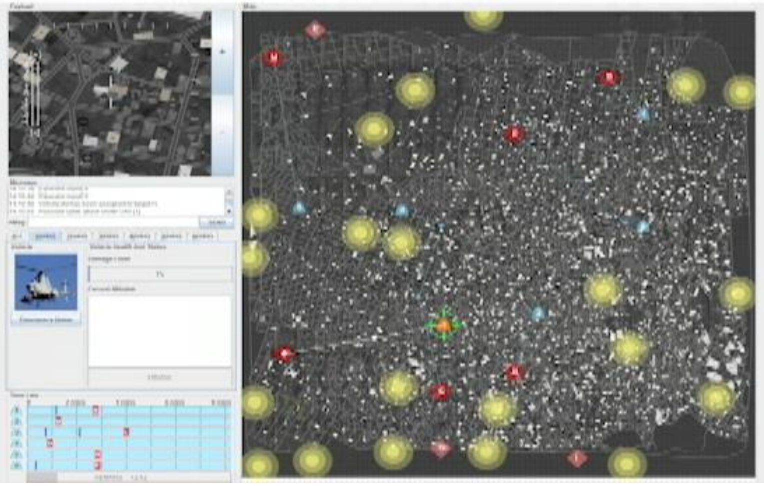
Adversary

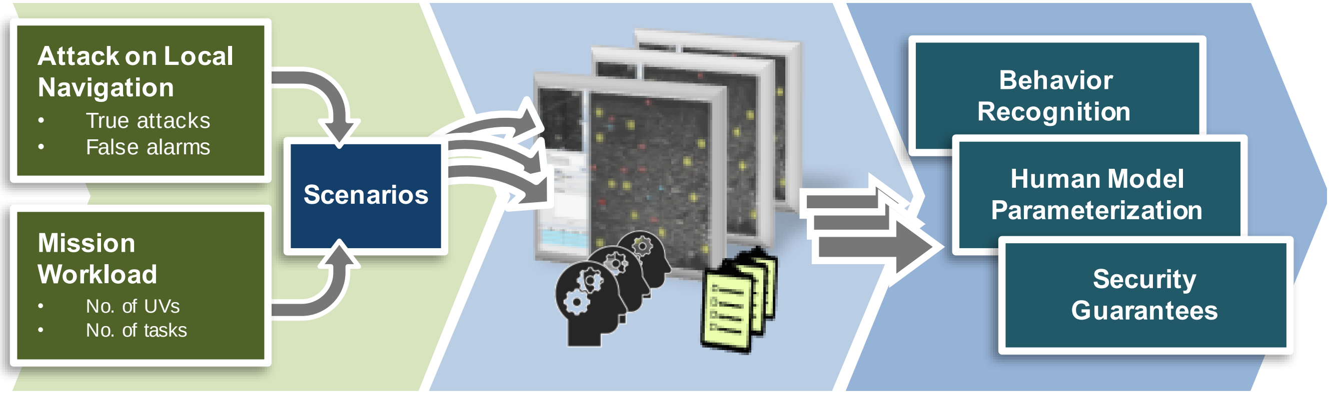
- Effects low-level control

Security-aware protocols

- Exploit human context-awareness for security

RESCHU-SA



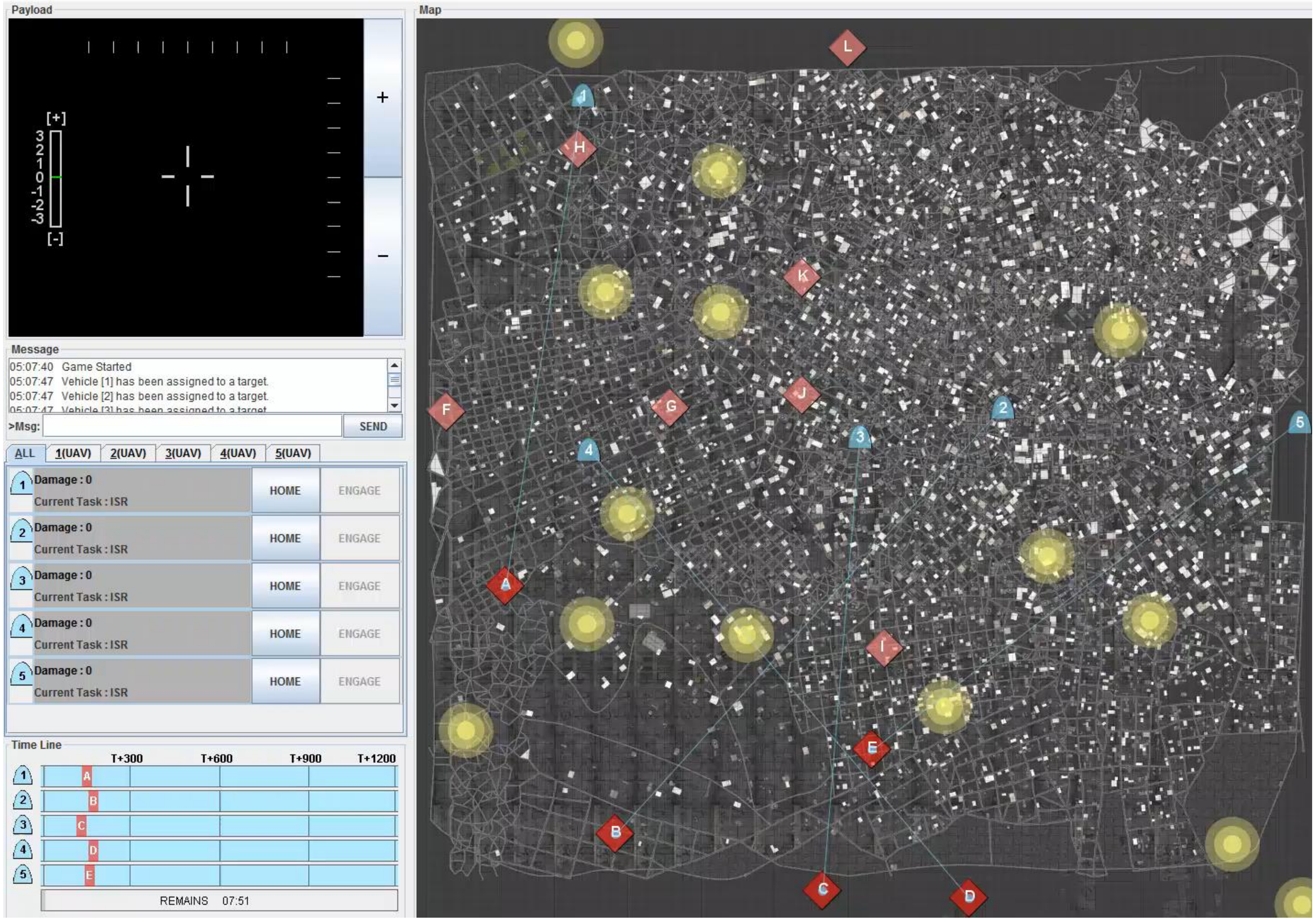


Scenarios
- Design of experimental variables
- Generate RESCHU-SA configuration files

Experiments
- Capture HOL behaviors with varying levels of workload and fatigue

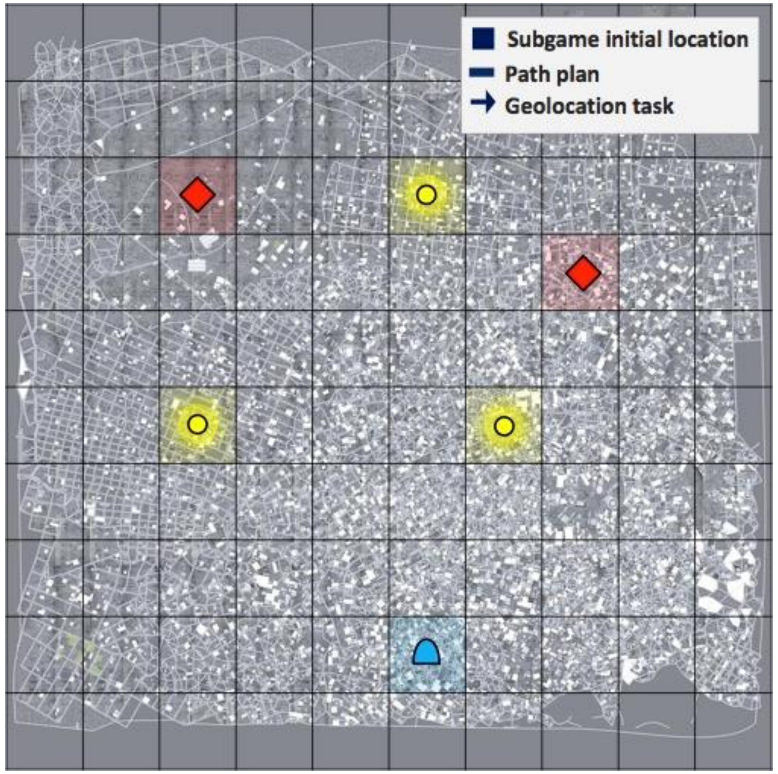
Data Mining
- HOL context awareness
- Impacts of workload and fatigue on system performance

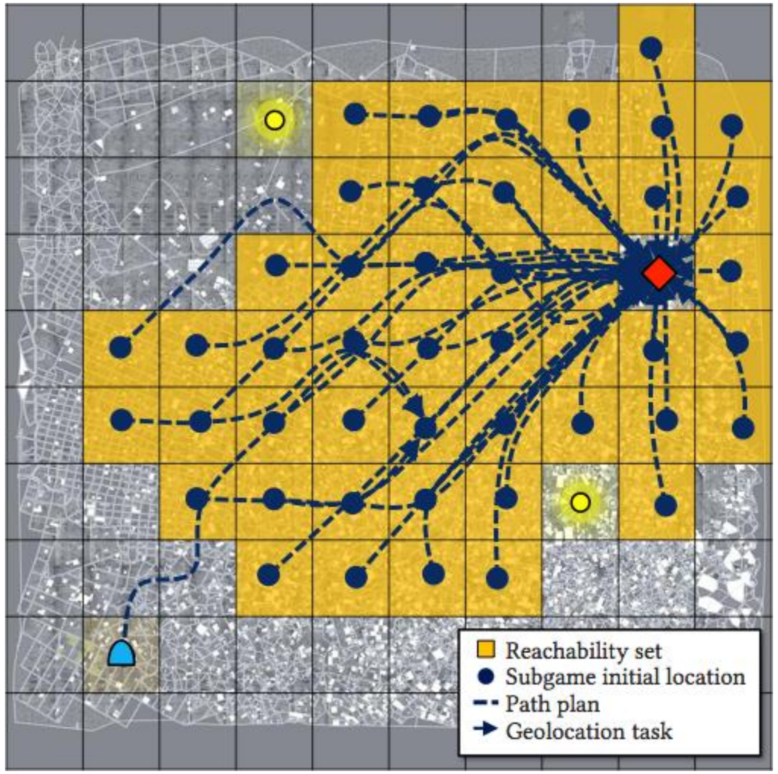
Security-aware Human-on-the-Loop Planning

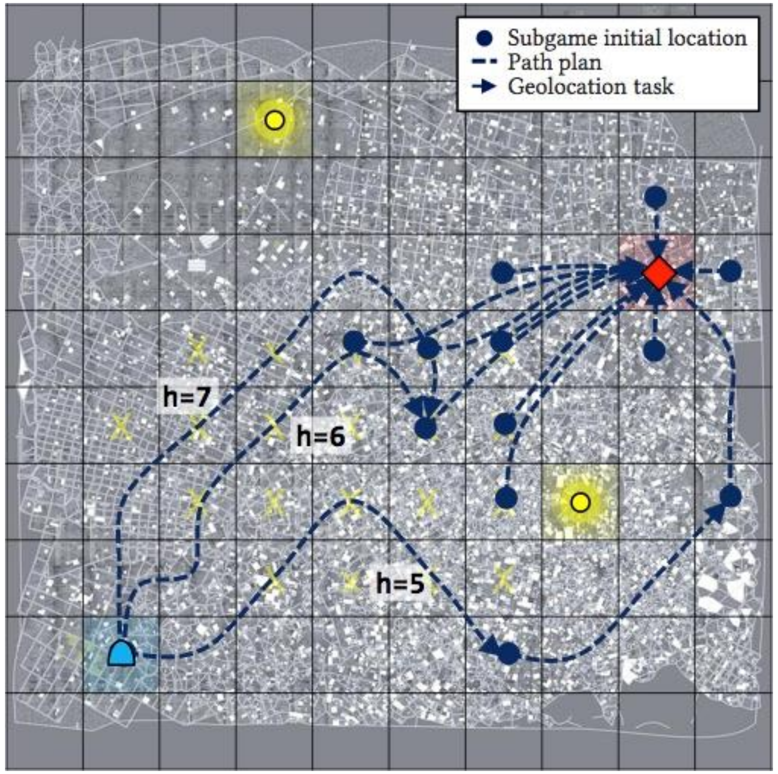


[ICRA'19, iEEE THMS'19]

Security-aware Human-on-the-Loop Planning [ICRA'19]







Attack-Resilient Mission Design

Duke PRATT SCHOOL *of* ENGINEERING

- Develop planning methods that will improve attack-detection guarantees by allowing the deployed intrusion detection system to interact with the controller and the rest of the system

- How to model such interactions? – MDPs, PTAs, SHAs

- Optimization based on solving stochastic games
  - How to incorporate learning?
  - How to incorporate formal guarantees?

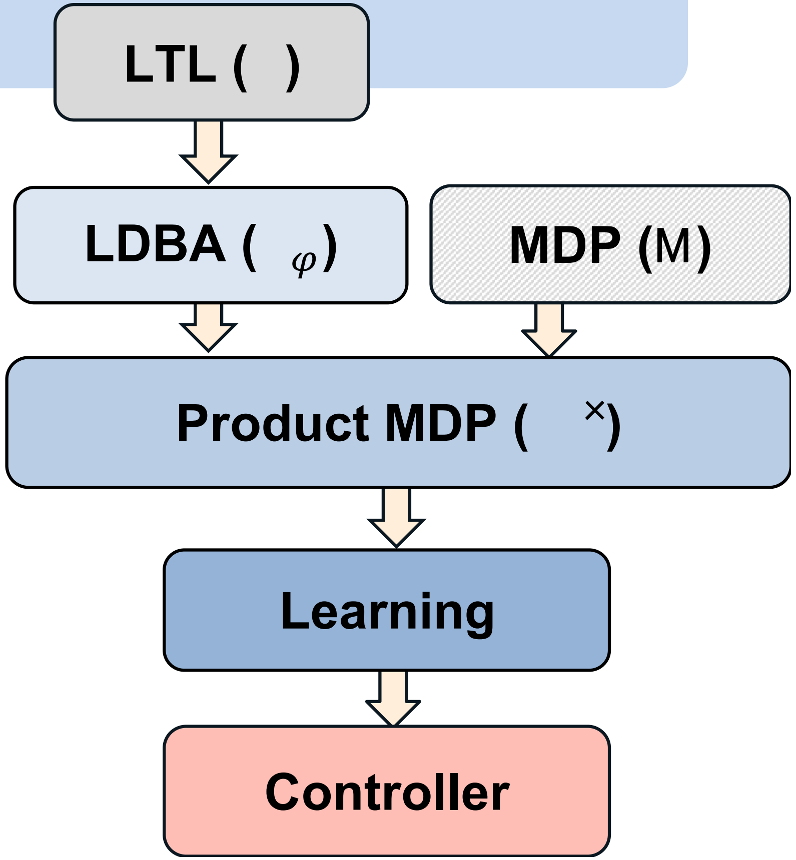
Model-free Control Synthesis from LTLs [ICRA20a*]

Problem Statement

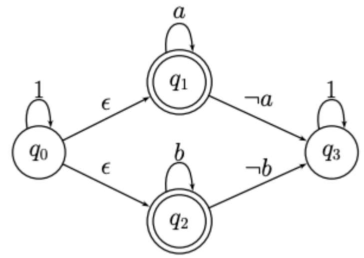
Given an MDP $\mathrm{M} = (S, A, P, s_0, AP, L)$ where P is ***fully*** unknown and an LTL specification φ, design a model-free RL algorithm that finds a finite-memory objective policy π^{φ} that satisfies

$$Pr_{\pi^{\varphi}}(s \vDash \varphi) = Pr_{max}(s \vDash \varphi),$$

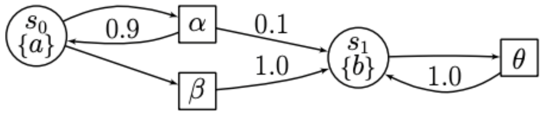
where $Pr_{max}(s \vDash \varphi) = max_{\pi} Pr_{\pi}(s \vDash \varphi)$ for all $s \in S$.



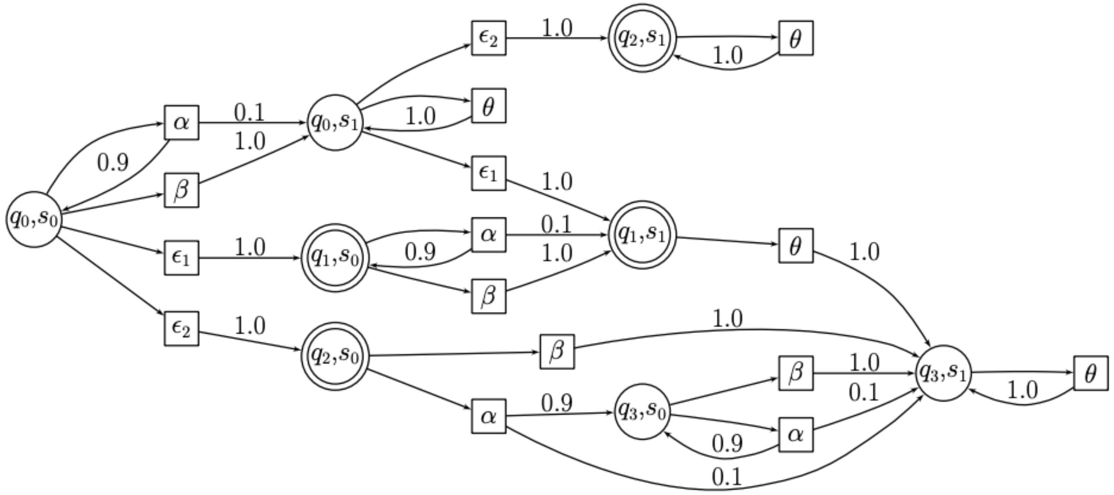
Limit-Deterministic Büchi Automata (LDBA) – consist of two deterministic components the ***initial*** and ***accepting***. The only nonde-terministic transitions are the ϵ-moves from the initial component to the accepting components.



(a) A derived LDBA $\mathcal{A}$ for the LTL formula $\varphi = \Diamond\Box a \vee \Diamond\Box b$



(b) An example MDP $\mathcal{M}$; the circles denote MDP states, rectangles denote actions, and numbers transition probabilities



(c) The obtained product MDP

Model-free Control Synthesis from LTLs

Problem Statement

Given an MDP $\mathrm{M} = (S, A, P, s_0, AP, L)$ where P is ***fully*** unknown and an LTL specification φ, design a model-free RL algorithm that finds a finite-memory objective policy π^{φ} that satisfies

$$Pr_{\pi^{\varphi}}(s \vDash \varphi) = Pr_{max}(s \vDash \varphi),$$

where $Pr_{max}(s \vDash \varphi) = max_{\pi} Pr_{\pi}(s \vDash \varphi)$ for all $s \in S$.

Learning for Büchi conditions

For a given MDP M with $B \subseteq S$, the value function v_{π}^{γ} for the policy π and the discount factor γ satisfies

$$\lim_{\gamma \to 1^-} v_{\pi}^{\gamma}(s) = Pr_{\pi^{\varphi}}(s \vDash \varphi)$$

for all states for all $s \in S$ if the return of a path is defined as

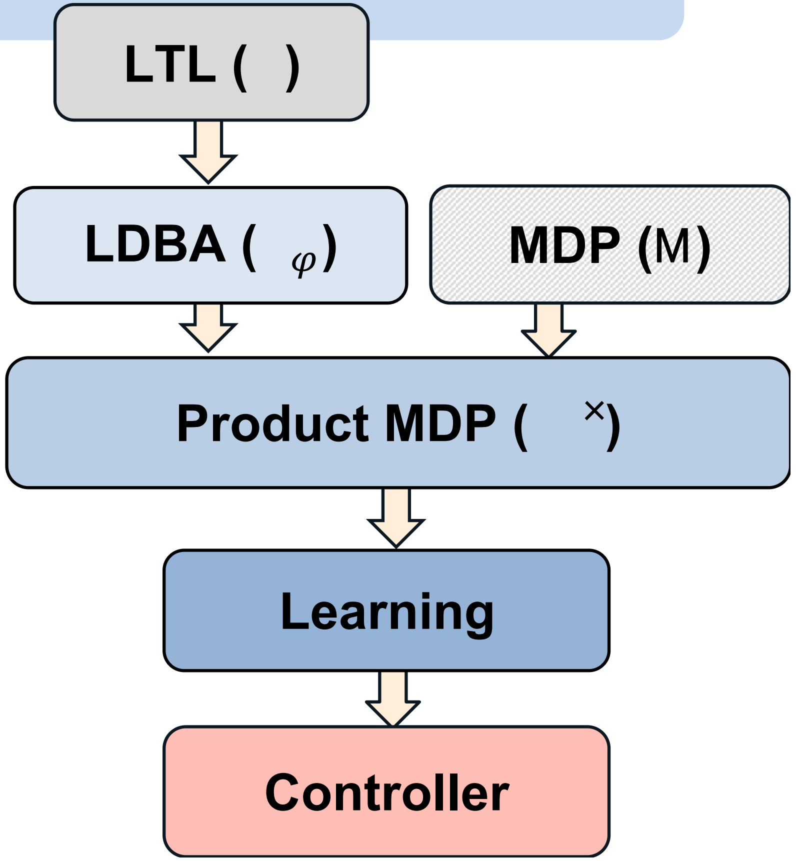
$$G_t(\sigma) \coloneqq \sum_{i=0}^{\infty} R_B(\sigma[t+i]) \prod_{j=0}^{i-1} \Gamma_B(\sigma[t+j])$$

where $\prod_{j=0}^{-1} \coloneqq 1$, $R_B: S \to [0,1)$ and $\Gamma_B: S \to (0,1)$ are the reward and the discount functions defined as

$$R_B(s) \coloneqq \begin{cases} 1-\gamma_B, & s \in B \\ 0, & s \notin B \end{cases}, \qquad \Gamma_B(s) \coloneqq \begin{cases} \gamma_B, & s \in B \\ \gamma, & s \notin B \end{cases}.$$

Here, we set $\gamma_B = \gamma_B(\gamma)$ as a function of γ such that

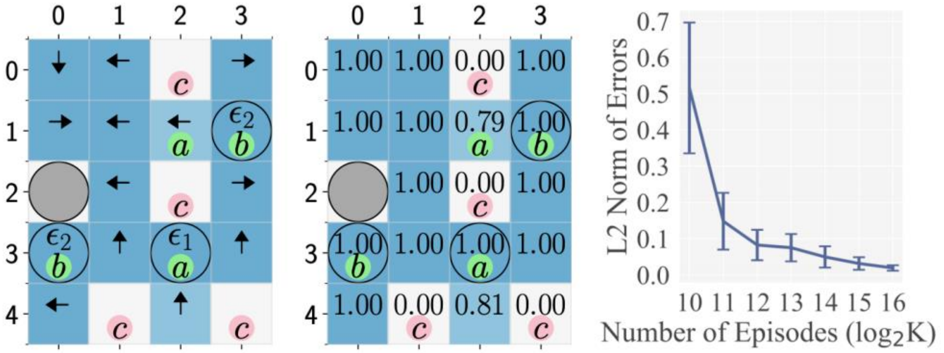
$$\lim_{\gamma \to 1^-} \frac{1-\gamma}{1-\gamma_B(\gamma)} = 0.$$



Case Studies

Robot tries to reach a safe absorbing state (states a or b in circle), while avoiding unsafe states (states c).

$$\varphi_1 = (\Diamond\Box a \vee \Diamond\Box b) \wedge \Box\neg c$$



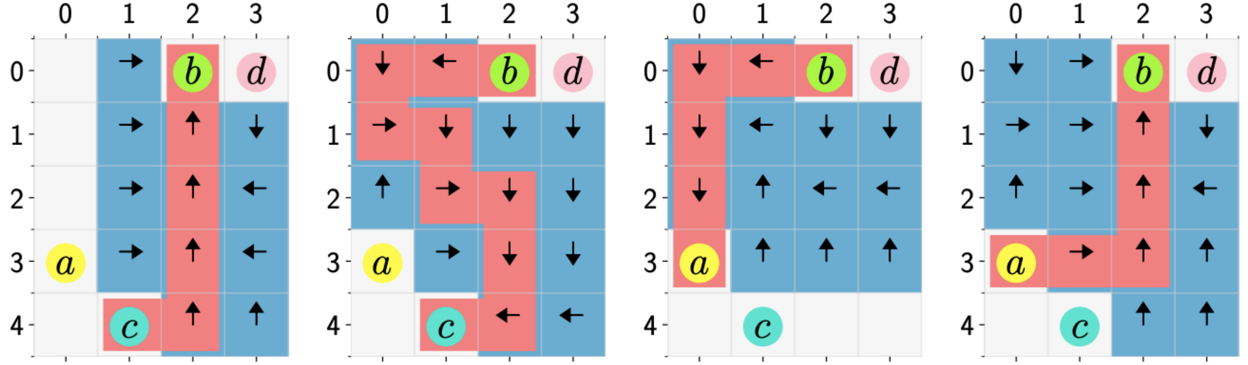
(a) Policy (b) Value Estimates (c) Convergence

Nursery Scenario

The robot's objective is to repeatedly check a baby (at state b) and go back to its charger (at state c), while avoiding the danger zone (at state d).

Near the baby b, the only allowed action is left and when taken the following situations can happen

- the robot hits the wall with probability 0.1, waking up the baby
- the robot moves left with probability 0.8 or moves down with probability 0.1.
- If the baby has been woken up, which means the robot could not leave in a single time step (represented by LTL as b ∧ Ob), the robot should notify the adult (at state a);
- otherwise, the robot should directly go back to the charger (at state c).

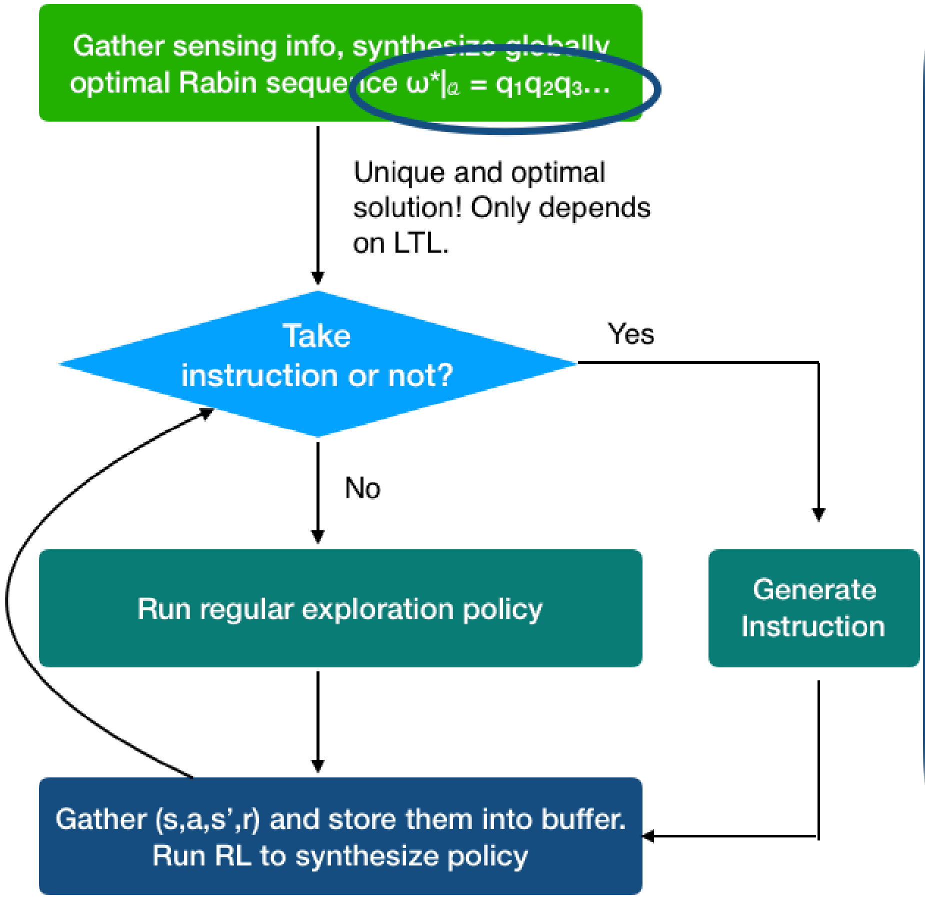


(a) Policy c to b (b) Policy b to c (c) Policy b to a (d) Policy a to b

$$\varphi_2 = \Box\Big(\underbrace{\neg d}_{(1)} \wedge \underbrace{(b \wedge \neg \bigcirc b) \to \bigcirc(\neg b \ \mathsf{U}\ (a \vee c))}_{(2)} \wedge \underbrace{a \to \bigcirc(\neg a \ \mathsf{U}\ b)}_{(3)} \wedge \underbrace{(\neg b \wedge \bigcirc b \wedge \neg \bigcirc \bigcirc b) \to (\neg a \ \mathsf{U}\ c)}_{(4)} \wedge \underbrace{c \to (\neg a \ \mathsf{U}\ b)}_{(5)} \wedge \underbrace{(b \wedge \bigcirc b) \to \Diamond a}_{(6)}\Big)$$

Synthesis from LTL via Deep Imitative Q-Learning [ICRA20b*]

Duke PRATT SCHOOL of ENGINEERING

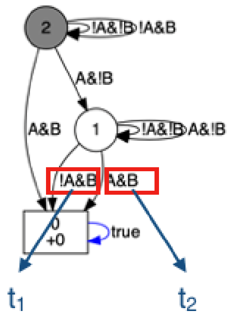


Current State: [s_3, q_2]

Locate q_2 in $\omega^*|_{\alpha}$:

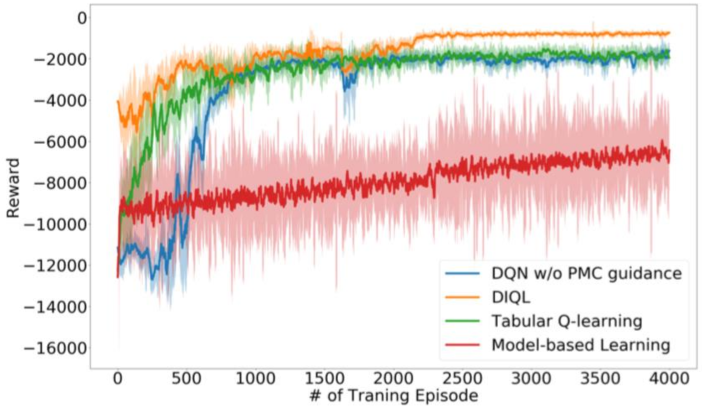
$\omega^*|_{\alpha} = q_1 q_2 q_3 \ldots$

Locate transition t1,t2… from q_2 to q3 in the DRA:



Set local LTL as ϕ'=t1 ∨ t2

Convert ϕ' to DRA' and take product w/ FTS to give instruction



MDP with 1600 states

Deep Imitative Reinforcement Learning for Temporal Logic Robot Motion Planning with Noisy Semantic Observations

Qitong Gao, Miroslav Pajic and Michael M. Zavlanos
Duke University
ICRA 20'

Attack-Resilient Mission Design

- Develop planning methods that will improve attack-detection guarantees by allowing the deployed intrusion detection system to interact with the controller and the rest of the system

- How to model such interactions? – MDPs, PTAs, SHAs

- Optimization based on solving stochastic games
  - How to incorporate learning ***in 2-player hidden information stochastic games***?
    - *with formal guarantees…*

Thank you