On the safety assessment of RPAS safety policy

ERTS

January 30th, 2020

Diego Couto, Kevin Delmas, Xavier Pucel

ONERA

THE FRENCH AEROSPACE LAB

retour sur innovation

Increasing demands for RPAS

Increasing number of operational concepts involving Remotely Piloted Aircraft Systems (RPASs)

- Urban logistic (CDiscount, La Poste, . . .)
- Infrastructure inspection (SNCF, RTE, . . .)
- Rescue mission (Helper drone, . . .)

Safety issues

Integrating Unmanned Aerial Vehicles in airspace raises safety issues:

Ground Risk Collision with infrastructure or on-ground population

Air Risk Air collision with inhabited aerial traffic

Plan

How are these risks managed?

Underlying assumptions

Classical aviation:

1. Aircraft is inhabited
   $\Rightarrow$ ensuring flight safety $=$ ensuring aircraft integrity
2. Pilot is on-board
   $\Rightarrow$ numerous safety actions involve the pilot

UAV:

1. UAV is uninhabited
   $\Rightarrow$ ensuring flight safety $\neq$ ensuring UAV integrity
2. Pilot is remote
   $\Rightarrow$ safety actions taken by the remote pilot and the drone

Leads to different risk management
Must be considered during the safety assessment

Underlying assumptions

Classical aviation:

1. Aircraft is inhabited
   $\Rightarrow$ ensuring flight safety $=$ ensuring aircraft integrity
2. Pilot is on-board
   $\Rightarrow$ numerous safety actions involve the pilot

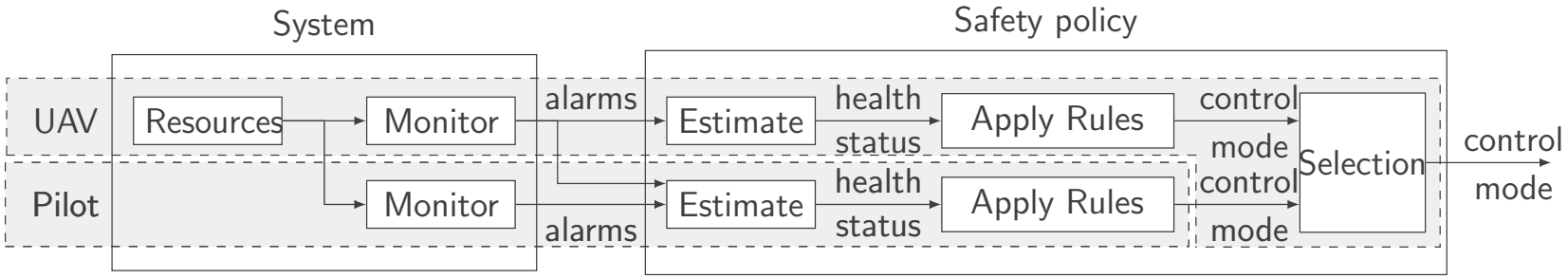
UAV:

1. UAV is uninhabited
   $\Rightarrow$ ensuring flight safety $\neq$ ensuring UAV integrity
2. Pilot is remote
   $\Rightarrow$ safety actions taken by the remote pilot and the drone

Leads to different risk management
Must be considered during the safety assessment

How hazardous situations are handled in an RPAS?

Safety policy by the example

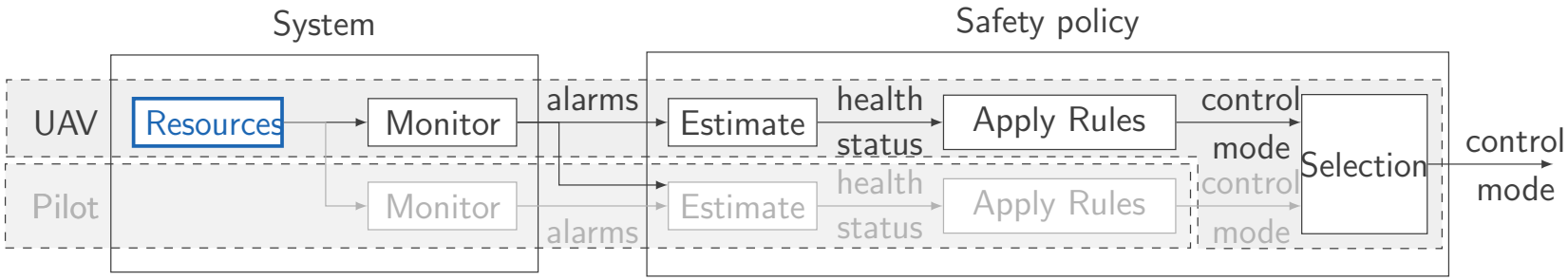


Mission Inspect infrastructures located in pre-defined and controlled evolution zone

Hazard Flyaway or crash outside of the evolution zone

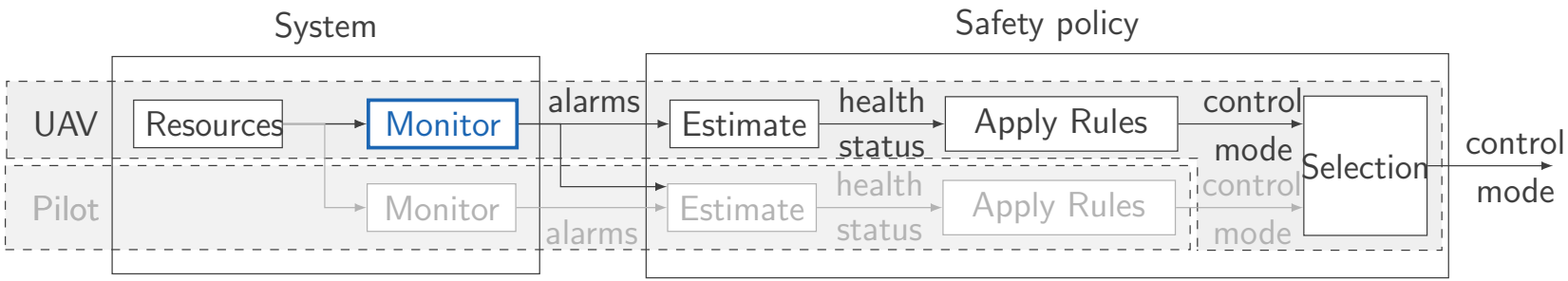
Modes Autonomous (A)
Return to home (H)
Descending spiral (S)

Safety policy by the example



Resource h_1 and h_2 needed by A, h_p needed by H

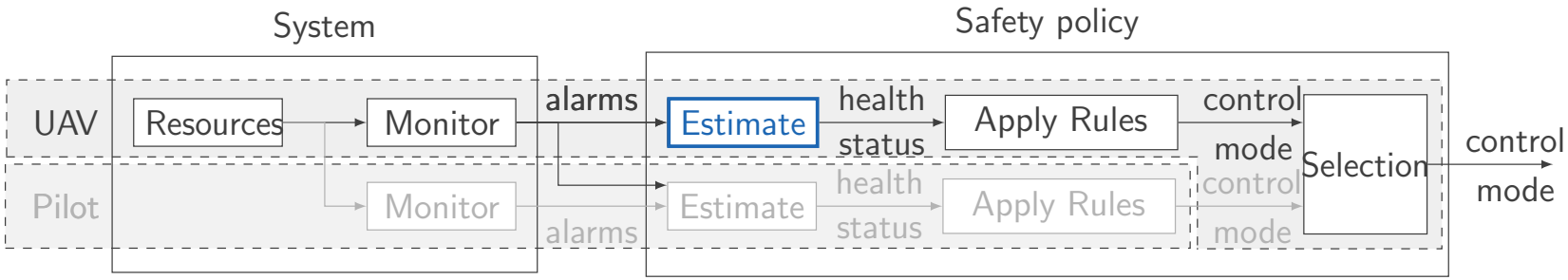
Safety policy by the example



Resource h_1 and h_2 needed by A, h_p needed by H

Monitor a_1 (resp. a_2) powered by h_p monitoring h_1 (resp. h_2)

Safety policy by the example

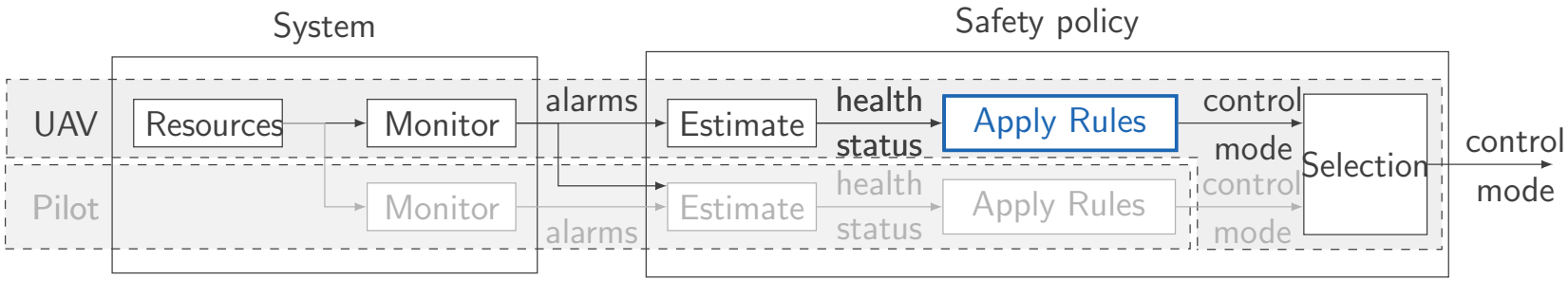


Resource h_1 and h_2 needed by A, h_p needed by H

Monitor a_1 (resp. a_2) powered by h_p monitoring h_1 (resp. h_2)

Estimate if a_1 (resp. a_2) then $\overline{h_1}$ (resp. $\overline{h_2}$)

Safety policy by the example



Resource h_1 and h_2 needed by A, h_p needed by H

Monitor a_1 (resp. a_2) powered by h_p monitoring h_1 (resp. h_2)

Estimate if a_1 (resp. a_2) then $\overline{h_1}$ (resp. $\overline{h_2}$)

Apply if $\overline{h_1}$ or $\overline{h_2}$ initiate H
if $\overline{h_p}$ initiate S

Challenges of safety assessment

Dynamism Policy is performed according to the successive estimation of the health status.

- addressed using modelling language for dynamic systems (ALTARICA[APGR99], [PPR16], ...)

Decision UAV on-board monitoring provides partial obervability $\Rightarrow$ possible health status estimation issues

1. selection of unsuitable mode
2. hazardous situations (flyaway, uncontrolled crash, ...)

Problem reformulation

- knowing the alarms (*i.e.* observations) received by the UAV and the pilot
- knowing the possible failures of on-board components (*i.e.* system model)
- a safety policy:
  1. selects a preferred health status among the possible ones
  2. provides a control mode out of this health status

Safety assessment identify when the policy is not able to select a safe mode

$\Downarrow$

Estimation problem identify mis-estimations (policy) leading to an unsafe mode selection

Problem reformulation

- knowing the alarms (*i.e.* observations) received by the UAV and the pilot
- knowing the possible failures of on-board components (*i.e.* system model)
- a safety policy:
  1. selects a preferred health status among the possible ones
  2. provides a control mode out of this health status

Safety assessment identify when the policy is not able to select a safe mode

$\Downarrow$

Estimation problem identify mis-estimations (policy) leading to an unsafe mode selection

Contribution

1. formal framework to model the safety policy as a preference-based estimator
   - **Modular** split system model, estimation preferences and mode selection
   - **Generic** no assumptions over the kind of UAV (fixed wing, quad-copter, ...)
2. formal encoding of hazardous events
   $\Rightarrow$ use existing solver to identify hazardous failure combinations

Why considering a preference-based estimation problem?

Estimation problem by the example

Modes Autonomous(A),
Return to home (H)
Descending spiral (S)

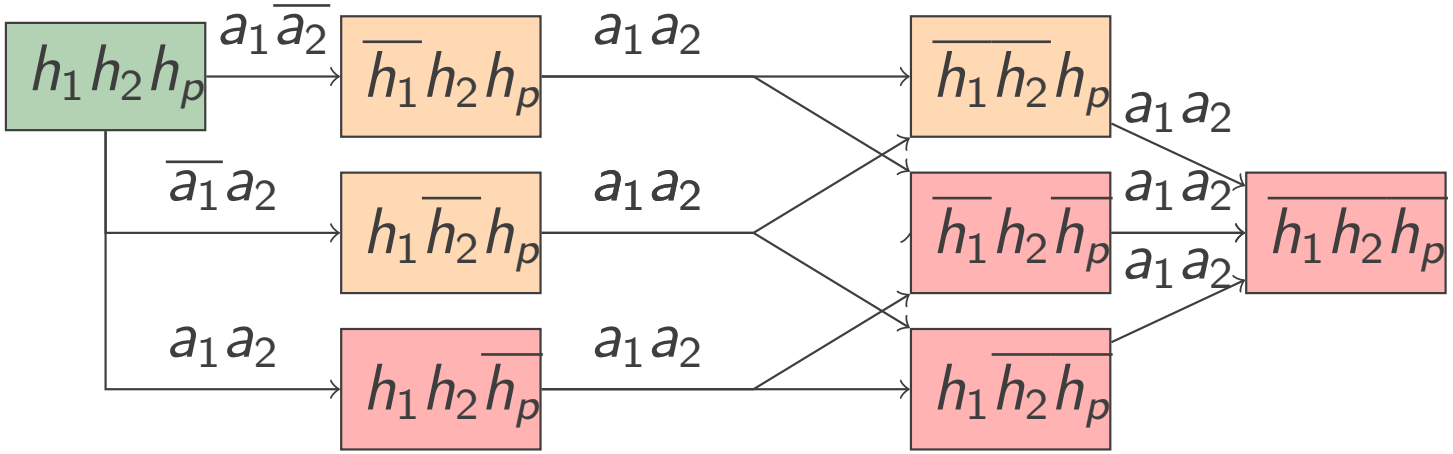
Resource h_1 and h_2 needed by A, h_p needed by H

Monitor a_1 (resp. a_2) powered by h_p monitoring h_1 (resp. h_2)

Assumptions

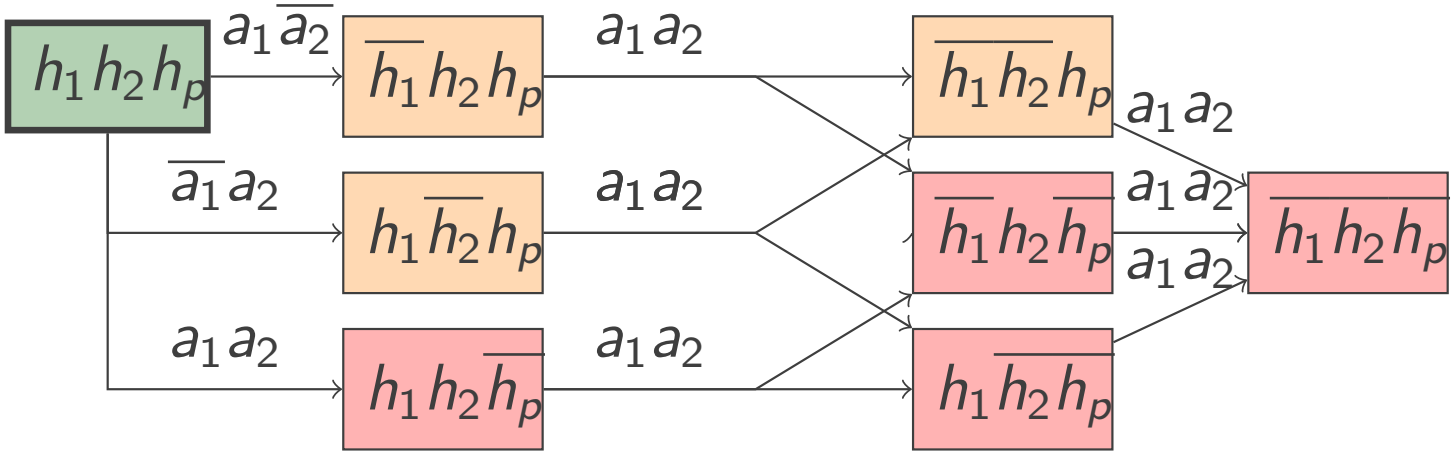
1. permanent failures
2. interleaving
3. only loss failure mode for resources

Estimation problem by the example



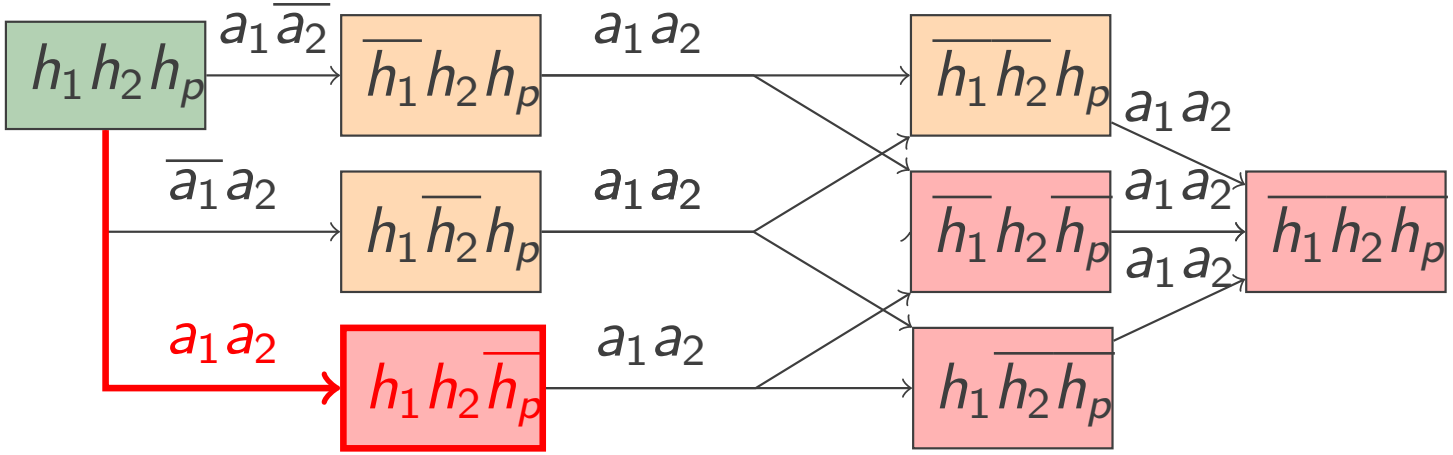
Observation	Real	Estimated

Estimation problem by the example



Observation	Real	Estimated
$\overline{a_1 a_2}$		$h_1 h_2 h_p$

Estimation problem by the example

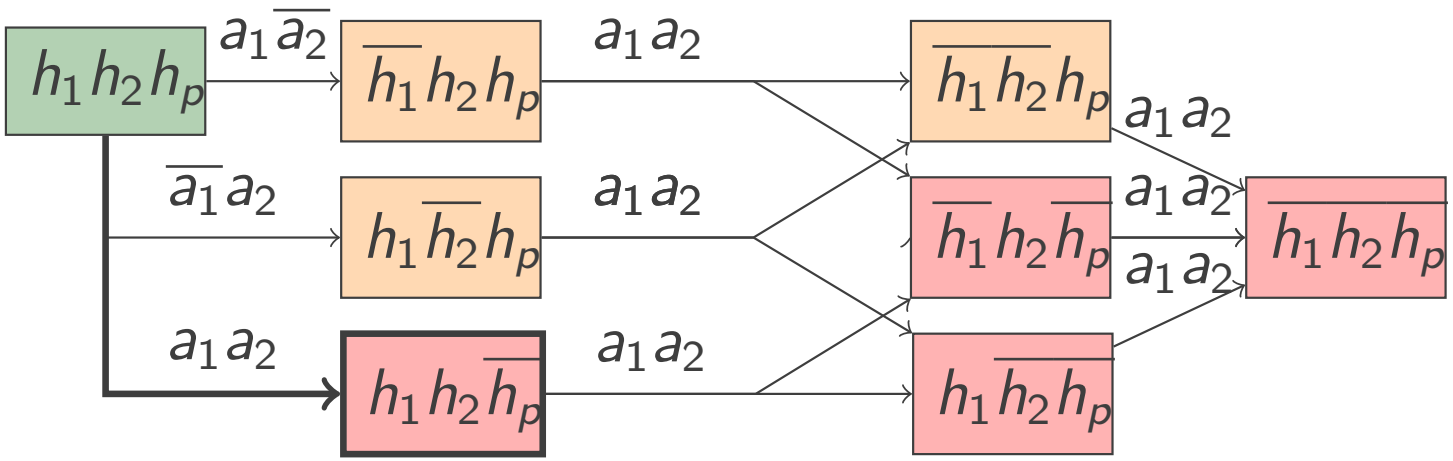


Observation	Real	Estimated
$\overline{a_1 a_2}$		$h_1 h_2 h_p$
$a_1 a_2$		$h_1 h_2 \overline{h_p}$?

if a_1 (resp. a_2) then h_1 (resp. h_2) failed

Cannot select mode

Estimation problem by the example



Observation	Real	Estimated
$\overline{a_1 a_2}$	$h_1 h_2 h_p$	
$a_1 a_2$	$h_1 h_2 \overline{h_p}$	$h_1 h_2 \overline{h_p}$

if a_1 (resp. a_2) prefers $\overline{h_1}$ (resp. $\overline{h_2}$)
if a_1, a_2 both triggered now and not previously prefers $\overline{h_p}$

Preference-based estimation

Modelling of estimation problem with preference provided in [PPR16]:

System model (Δ) Possible behaviours (state transitions) of the system, encoded as a set of PTLTL constraints

Example (Hard constraint)

An alarm is set either when the monitored resource fails or the power supply of the alarm fails. $a_1 \Leftrightarrow \overline{h_1} \vee \overline{h_p}$

Preference (Γ) Ordered conditional preferences (when several possible values)

Example (Preference)

$\overline{h_p}$ is preferred when a_1, a_2 both triggered now and not previously $\overline{h_p} \leftrightsquigarrow \neg Y(a_1) \wedge \neg Y(a_2) \wedge a_1 \wedge a_2$

How do we encode a safety policy using this formalism?

Encoding the safety policy: Main idea

Resource model(Δ_R) Failure model of on-board components

- possible failures of the on-board components
- requested resources for each mode
- assumptions over failure occurrence

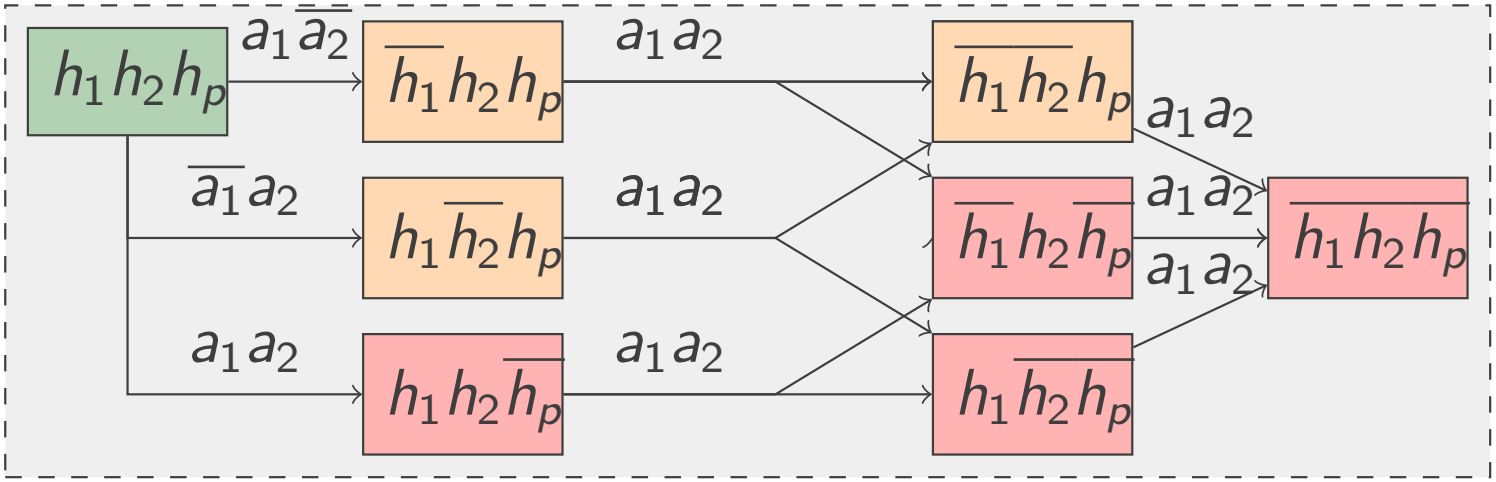
Alarm model(Δ_A) Failure model of alarms

- possible failures of the alarms
- monitoring capabilities of alarms

Resource preferences (Γ_R) preferred failures considering alarms

Mode preferences (Γ_M) preferred modes considering estimated available resources

Encoding the safety policy: Example



1. if a_1 (resp. a_2) prefers $\overline{h_1}$ (resp. $\overline{h_2}$)
2. if a_1, a_2 both triggered now and not previously prefers $\overline{h_p}$

Γ

Framework features

- Structure to encode failure modes, resources, alarms and mode dependencies
- Library of generic constraints to encode:
  - failure assumptions (permanent failures, exclusive failures, interleaving, ...)
  - alarm behaviours (active low/high alarms,...)
  - failure preference (common cause, non monitored components, ...)
  - mode selection (exclusivity, pilot/UAV priority,...)

An example of framework usage

Active low alarm a with:

- monitoring r with a set of detectable failure modes F
- a false negative failure mode fn,
- requesting a set N of resources is modelled by:

$$\overline{a} \Leftrightarrow \left(fn \vee \left(\bigwedge_{f \in F} \overline{f} \wedge \bigwedge_{\substack{r \in N, \\ f' \in r.fm}} \overline{f'} \right) \right)$$

Example (Active low alarm)

An active low alarm *alpi* (powered by *pow*) over a component *pi* is modelled by

$$\overline{alpi} \Leftrightarrow \left(alpi.fn \vee \left(\overline{pi.LS} \wedge \overline{pow.LS} \right) \right)$$

How to identify hazardous failure combinations?

Safety assessment as bounded reachability

Hazardous situations

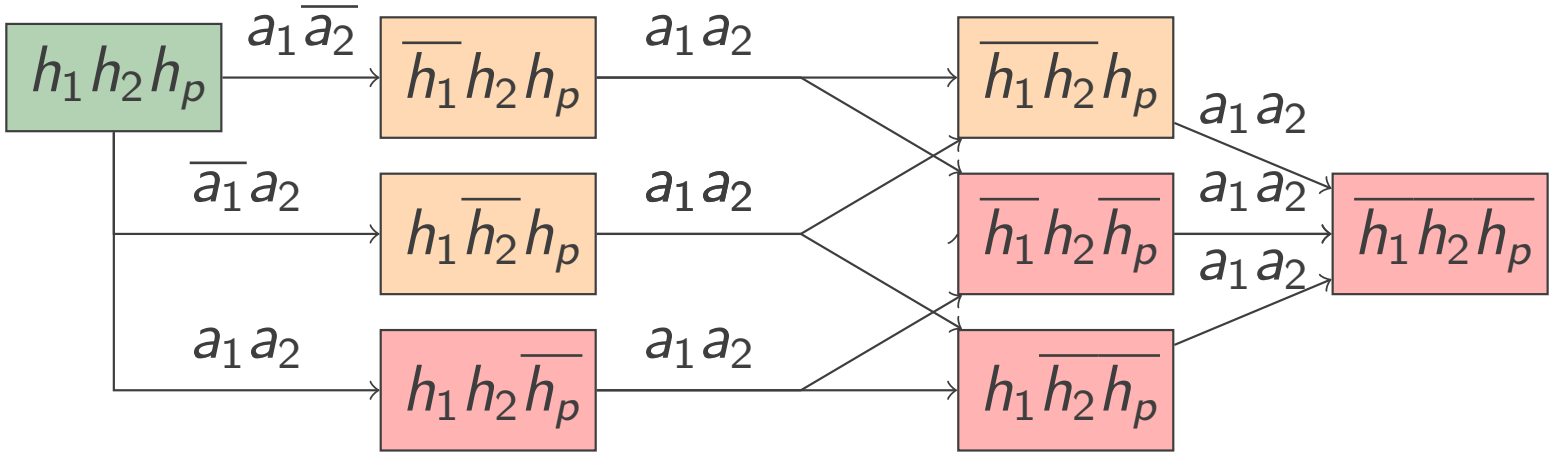
1. combination of failures (of bounded size) leading to unsafe mode selection
2. mis-estimation of the health status
3. addressable through automated bounded reachability analysis

Definition (Reachability analysis)

Safety assessment performed with $\text{REACHABLE}_{\Delta,\Gamma}(\phi_R,\phi_E,n)$ that enumerates pairs (S_R, S_E) and $(e_i)_{[1,n]}$ where:

- S_R satisfies Δ and S_E satisfies both Δ and Γ;
- at the last time step, S_R satisfies ϕ_R and S_E satisfies ϕ_E
- e_i the failure event(s) on the transition $S_{R_{i-1}} \rightarrow S_{R_i}$

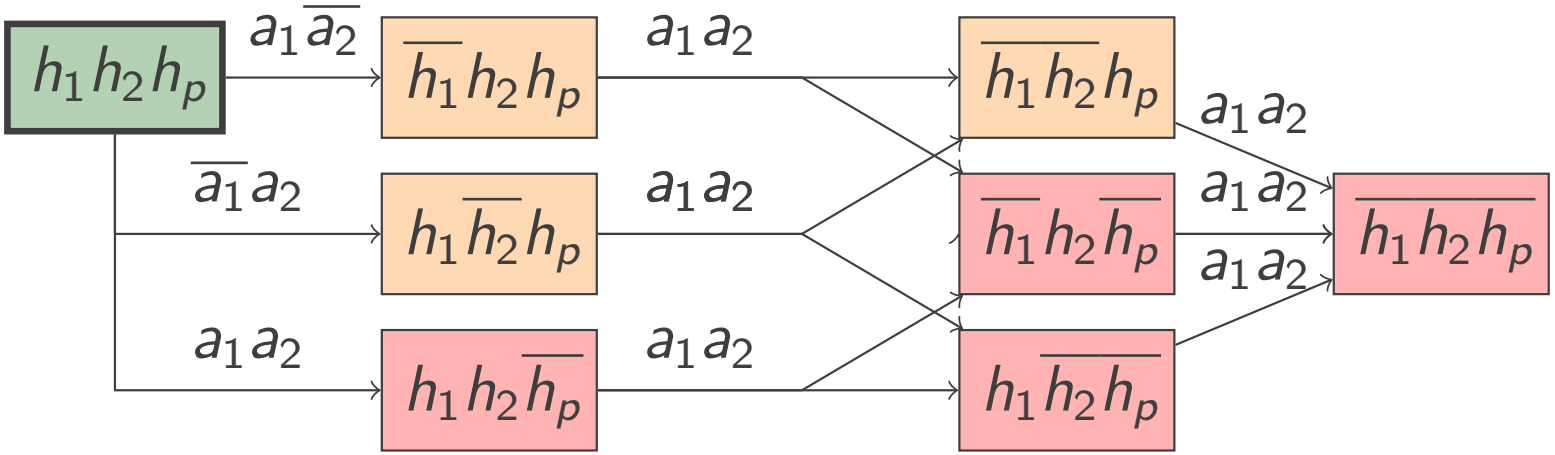
An example of reachability assessment



Observation	Event	Real	Estimated

Is there a failure sequence leading to a flyaway?

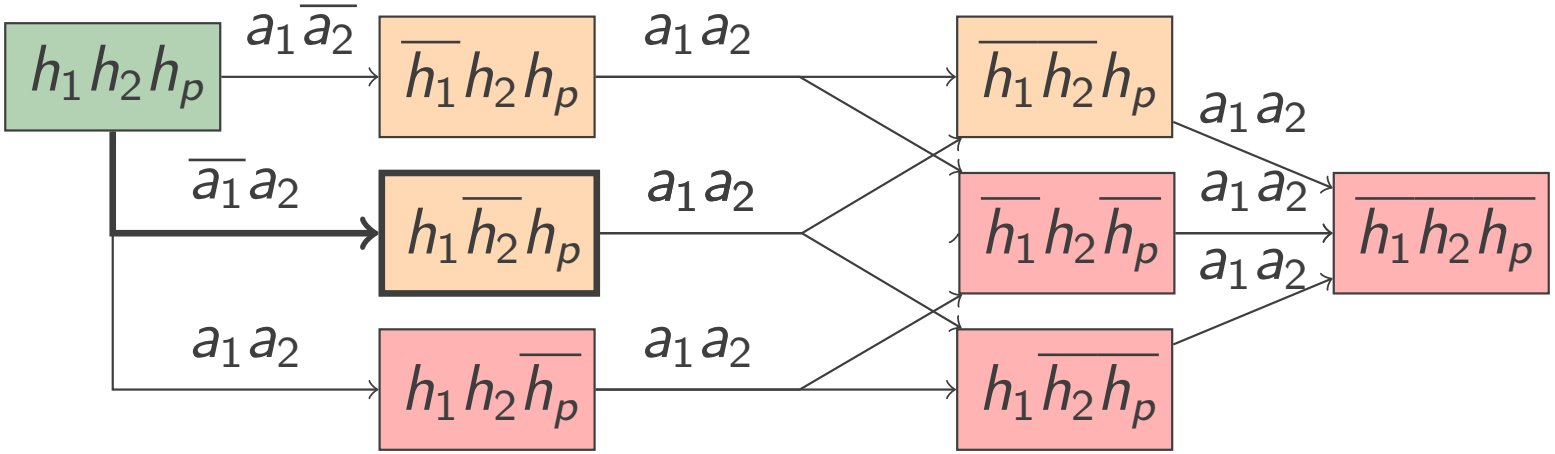
An example of reachability assessment



Observation	Event	Real	Estimated
$\overline{a_1 a_2}$	$\emptyset$	$h_1 h_2 h_p$	

Is there a failure sequence leading to a flyaway?

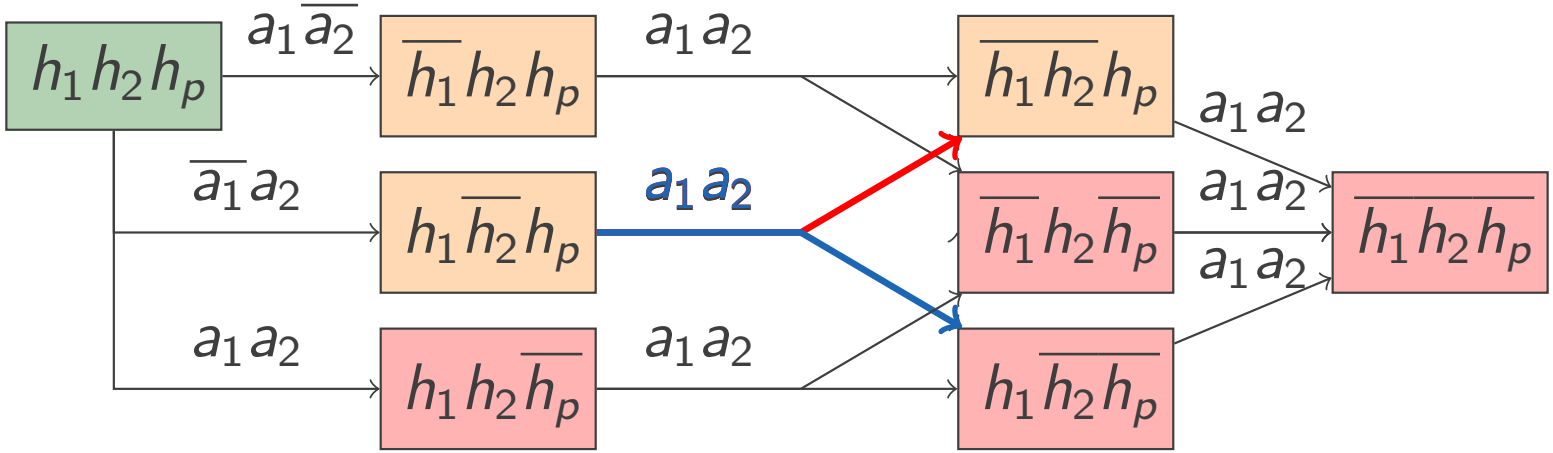
An example of reachability assessment



Observation	Event	Real	Estimated
$\overline{a_1 a_2}$	$\emptyset$	$h_1 h_2 h_p$	
$\overline{a_1} a_2$	$h_2.f$	$h_1 \overline{h_2} h_p$	

Is there a failure sequence leading to a flyaway?

An example of reachability assessment



Observation	Event	Real	Estimated
$\overline{a_1 a_2}$	$\emptyset$	$h_1 h_2 h_p$	
$\overline{a_1} a_2$	$h_2.f$	$h_1 \overline{h_2} h_p$	
$a_1 a_2$	$h_p.f$	$h_1 \overline{h_2 h_p}$	$\overline{h_1 h_2 h_p}$

Is there a failure sequence leading to a flyaway? Yes

Evaluation of the framework

Evaluation of SCALA implementation on a toy example:

Order	Failures		Comments
1	$piLaw.LS$ $piLaw.ES$		Undetectable steering control failure
	$guLaw.LS$ $guLaw.ES$		Undectable guidance control failure
2	$a_{pi}.FN$	$pi.LS$	Steering sensors failure and
⋮	⋮	⋮	⋮

Table: Excerpt of safety assessment of the RPAS for the Fly-Away

Conclusion

Proposed a generic framework providing:

- Formal way to encode safety policy
- Library of generic constraints to encode classical assumptions
- Tailorable to various UAV architectures, control modes and monitoring capabilities
- Automatic safety assessment through reachability analysis

Limitations & Future works

Experimental validation

- Performed on a toy example
  $\Rightarrow$ need to be assessed on realistic use case to assess scalability
- Limited modelling of the pilot
  $\Rightarrow$ extend the library

Assessment performance

- Reduce computation time with restriction of the computation to minimal scenarios
- Consider other assessment methods *i.e.* deadends assessment.

Thank you

Any question?

Bibliography I

André Arnold, Gérald Point, Alain Griffault, and Antoine Rauzy.

The altarica formalism for describing concurrent systems.
Fundamanta Informaticae, 40(2-3):109–124, 1999.

Cedric Pralet, Xavier Pucel, and Stéphanie Roussel.

Diagnosis of intermittent faults with conditional preferences.
In *Proceedings of the 27th International Workshop on Principles of Diagnosis (DX'16)*, 2016.