Evaluating Monte-Carlo Tree Search for Property Falsification on Hybrid Systems

Rémi Delmas, Thomas Loquen (ONERA)
Josep Boada-Bauxell, Mathieu Carton (Airbus Operations SAS)

ONERA – Airbus Operations SAS

20/06/2019

Outline

Outline

The IKKY-SEFA Project (2016–2019)

Intégration cocKpit & sYstèms – Embedded Systems & Advanced Functions

- Funding: Délégation Générale de l'Aviation Civile (DGAC)
- Partners: ONERA, Airbus, Dassault, LAAS-CNRS
- Goals:
  - Improve the **design processes** for industrial embedded systems
  - **Evaluate** the SoA of **hybrid systems verification** ...
    - Model-checking, SAT-modulo-semidefinite-programming, robustness analysis, reinforcement learning, ...
  - ...**on industrial benchmarks**:
    - An aircraft pitch control law (Airbus),
    - Reference model + altered models.

This presentation

Reinforcement learning techniques applied to property falsification.

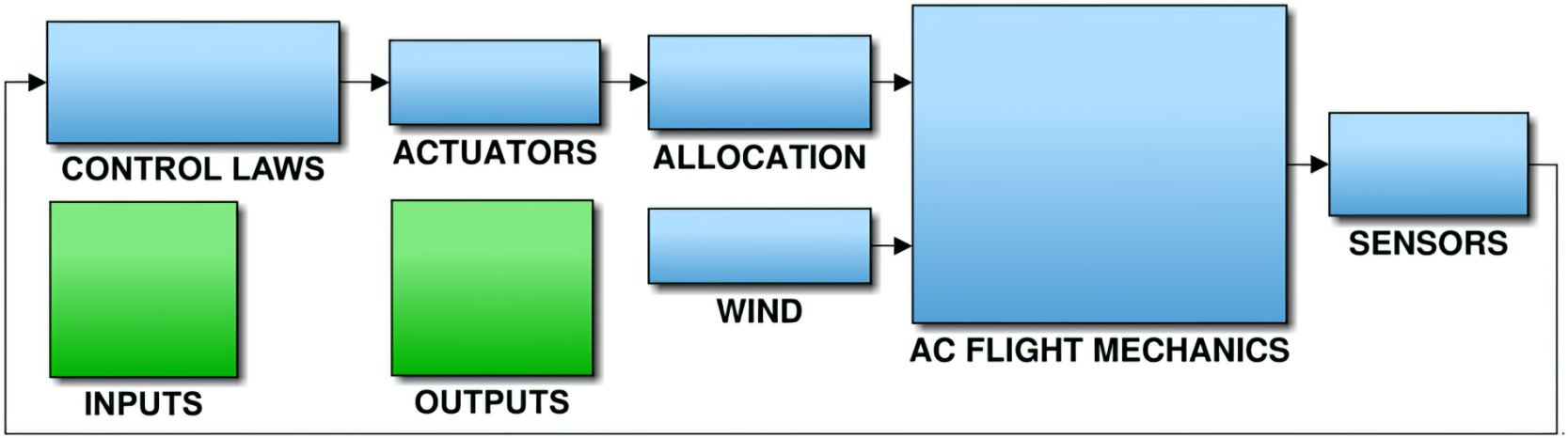
Hybrid Systems Verification Challenges

In our particular case:

- Time-Discrete/Continuous hyrbid closed-loop model,
- Modal control law: manual mode, autopilot mode, flight envelope protection,
- Non-linearity: polynomials, trig. functions, LUTs, vote, saturations, ...
- matlab/Simulink: no formal semantics, numerical issues (ODEs), ...

Outline

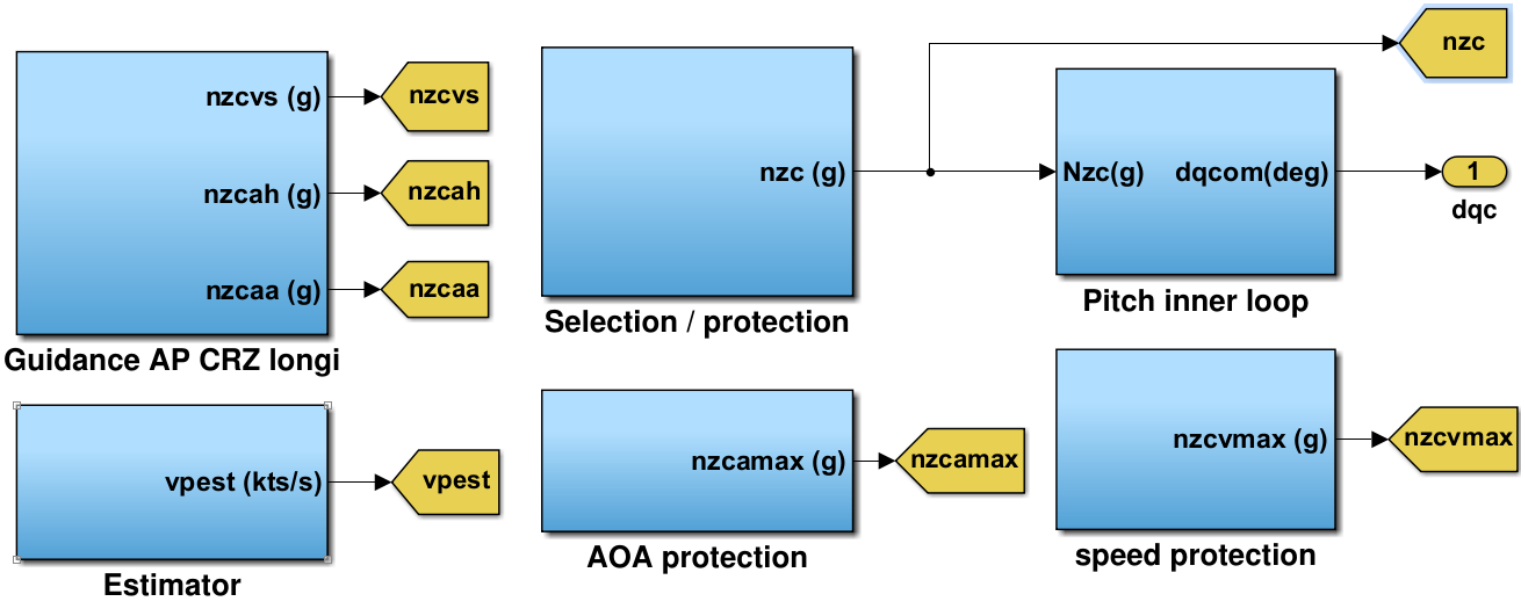
Benchmark Overview



Composants

- Continuous time (ODE),
- Aircraft model: flight dynamics + wind,
- Actuator model: order allocation, dynamics, saturations,
- Sensor model: dynamics, filtering, delay.

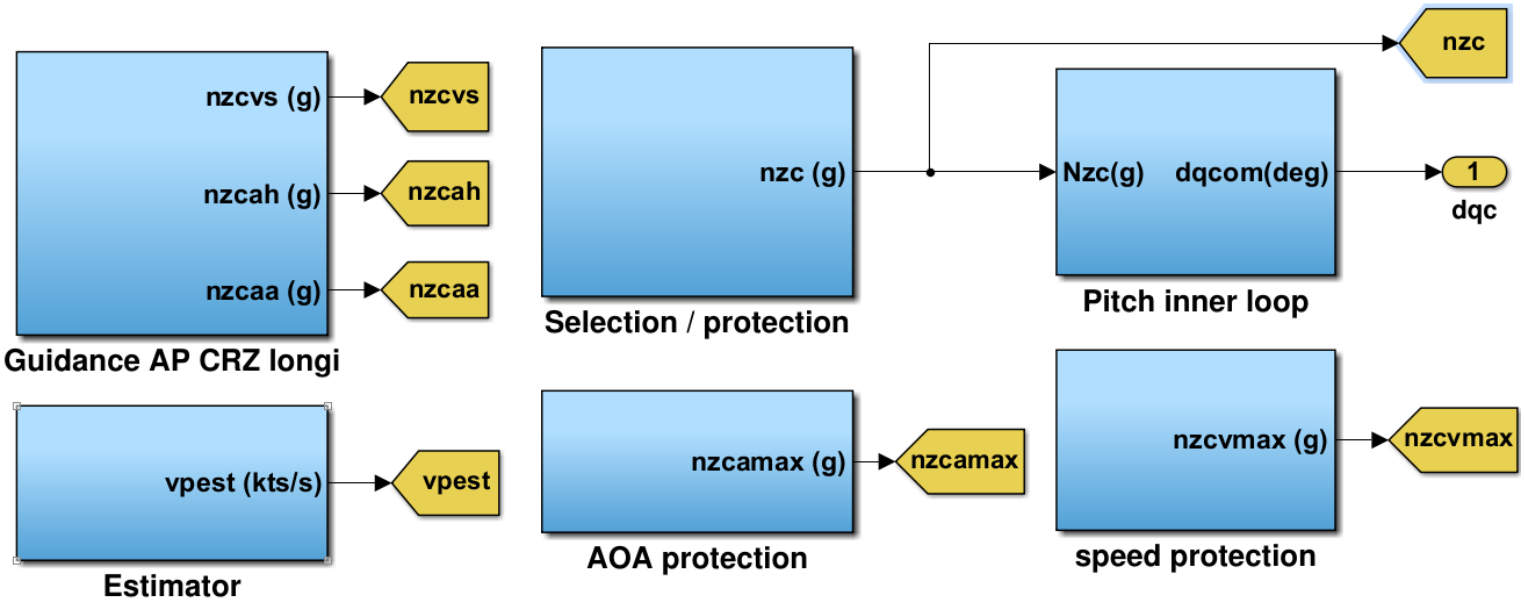
Control Law



Components

- Discrete time, multi-rate,
- AC State estimation,
- Feedback control on n_z (LPV),
- Manual and autopilot modes,
- Dynamic flight envelope protection.

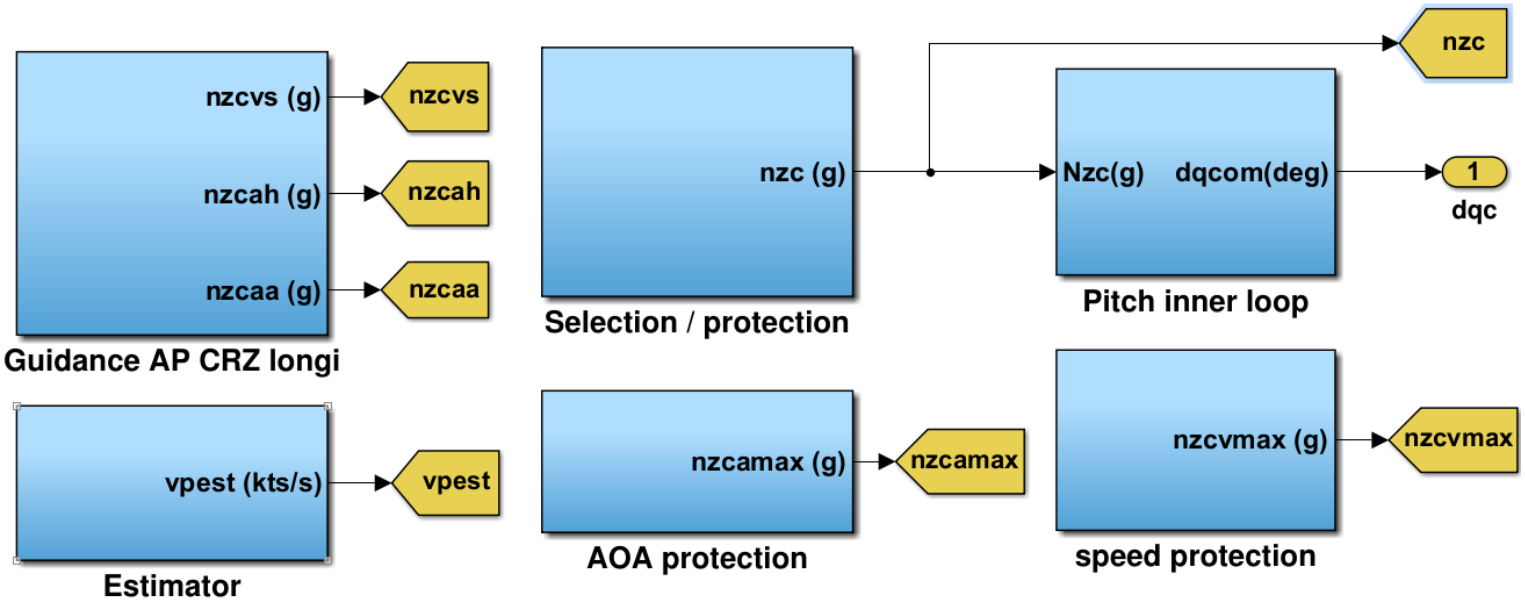
Control Law



Controllable inputs

- **bapeng** : Boolean d'*autopilot engagement*,
- **selalt** : real d'*selected altitude* for autopilot,
- **nzcmanche** : real d'*pilot stick order* for direct mode,
- **wx**,**wz** : real wind speed on horiz. and vert. axes.

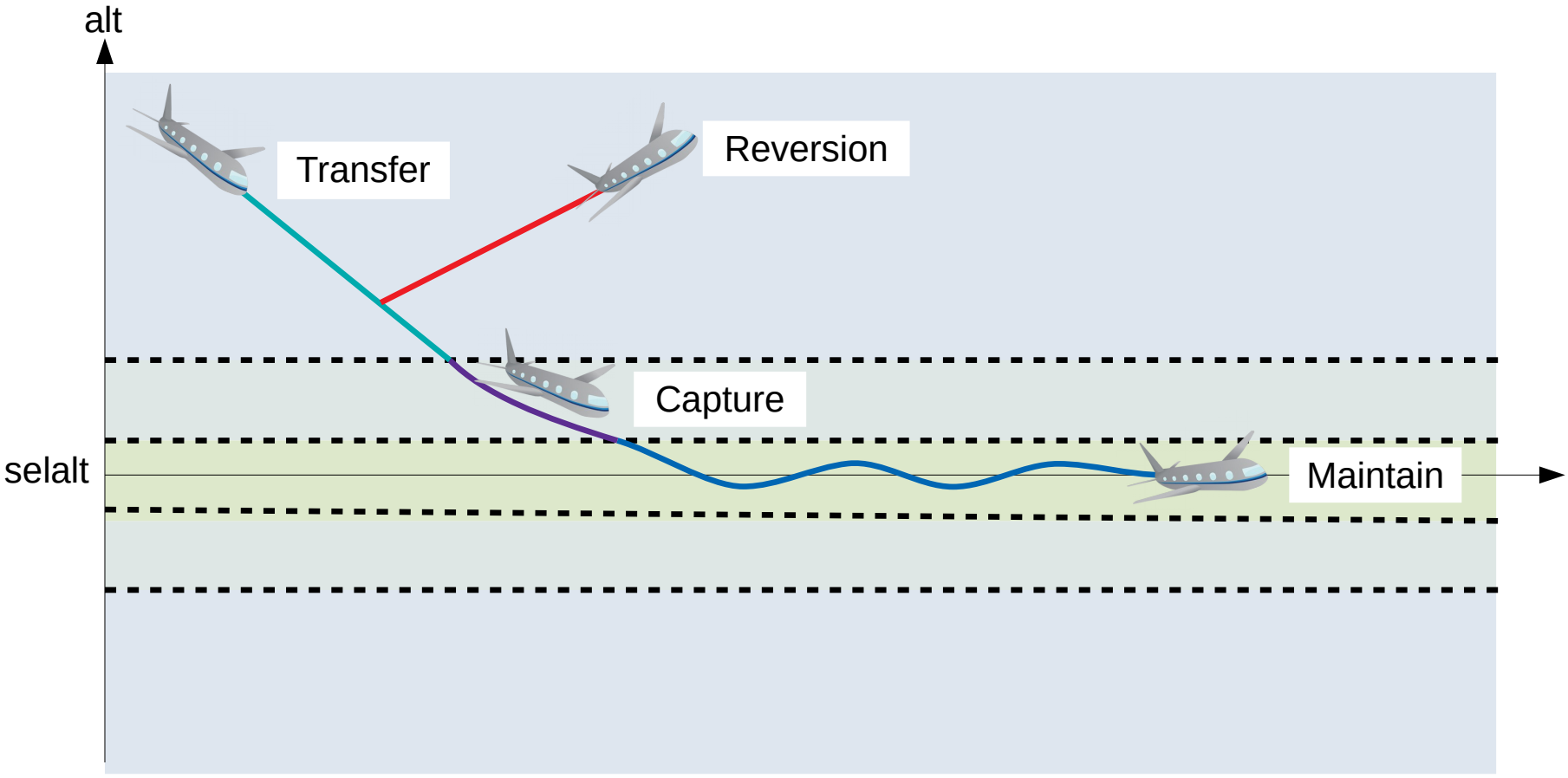
Control Law



Some Figures

- 112 continuous states, 27 switches, 4 latches, 28 2D-LUTs, 34 saturations.
- discrete multi-rate: $T1$ et $T2 = 1.5\,T1$

Autopilot Modes



Outline

Temporal Logics with Robust Semantics

The Signal Temporal Logic (STL) [4] with language, $(a, b) \in \mathbb{R}^2$:

$$\begin{array}{lcl} \phi & ::= & \text{true} \mid x_i \geq 0 \mid \neg\phi \mid \phi \wedge \phi \mid \phi \, \mathcal{U}_{[a,b]} \, \phi \\ \mathcal{F}_{[a,b]} \, \phi & \triangleq & \text{true} \, \mathcal{U}_{[a,b]} \, \phi \\ \mathcal{G}_{[a,b]} \, \phi & \triangleq & \neg(\mathcal{F}_{[a,b]} \, \neg\phi) \end{array}$$

defines, in addition to the standard Boolean interpretation $. \models .$, a *robust* interpretation $\mathcal{R}$ over timed traces w such that:

$$\mathcal{R}(\phi, w, t) \geq 0 \text{ iff } (w, t) \models \phi$$

Where:

$$\begin{array}{lcl} \mathcal{R}(\text{true}, w, t) & = & +\infty \\ \mathcal{R}(\text{x}_\text{i} \geq 0, w, t) & = & x_i^w(t) \\ \mathcal{R}(\neg\phi, w, t) & = & -\mathcal{R}(\phi, w, t) \\ \mathcal{R}(\phi_1 \wedge \phi_2) & = & \min(\mathcal{R}(\phi_1, w, t), \mathcal{R}(\phi_2, w, t)) \\ \mathcal{R}(\phi_1 \, \mathcal{U}_{[a,b]} \, \phi_2) & = & \max_{t' \in t+[a,b]}(\min(\mathcal{R}(\phi_2, w, t'), \min_{t'' \in [t,t']}(\mathcal{R}(\phi_1, w, t'')))) \end{array}$$

From Verification to Optimization

Given:

- $H = \langle S_H, A_H, T_H \rangle$ a hybrid model with:
  - S_H: state space,
  - A_H: controllable input space,
  - T_H: hybrid transition relation,
- $T \in \mathbb{R}$ a finite *horizon*,
- $d \in \mathbb{R}$ a constant *action duration* $d < T$,
- $\Phi = \mathcal{G}_{[0,T]}\ \phi$: a safety property on H with ϕ modality-free,
- $\mathrm{sim}(T_H, s, a, d)$ the trajectory of H from state s duration d with constant control input a.

We define a finite-action MDP $M = \langle S, A, T, R, \alpha \rangle$ where:

- $S \subseteq S_H$,
- $A \subseteq A_H$ is finite and user-specified,
- $(s, a, s') \in T$ *iff* s' is the final state of $\mathrm{sim}(T_H, s, a, d)$
- $R(s, a, s') = -\mathcal{R}(\phi, \mathrm{sim}(T_H, s, a, d), d)$

The falsification problem of Φ on H from state s_0 is under-approximated as an optimal planning problem on M from s_0 over finite horizon T, where the goal is to generate a finite action sequence producing a trace w such that $\mathbb{E}[\mathcal{R}(\Phi, w, t)]$ is the minimal for all t.

Outline

Monte-Carlo Tree Search (MCTS)

Monte-Carlo Tree Search (MCTS)[3], [6] is a generic algorithm for finite-horizon planning of discrete-action MDPs which builds a search tree over A from the initial state, and estimates $\mathbb{E}[R_{t+i}|a_t = a_0, \ldots, a_{t+i} = a_i]$ in each node of depth i using:

Rollout for fringe nodes: the cumulative reward is sampled to the horizon using a stochastic policy,

Backup for internal nodes: backpropagates estimates from subtrees up to the root,

Multi-Armed Bandit policies to select which branch to grow during search.

Multi-Armed Bandits

A K-Bandit problem is defined by:

- a set of sequences of random variables $R_{i,n}$ pour $i \in [1, K]$, $n \in \mathbb{N}^+$, i.i.d with unknow mean μ_i and finite variance σ_i^2,
- $R_{i,*}$ and $R_{j,*}$ are independant for all i, j.

A each game step n, the learner choses an arm i and gets a reward $r_{i,n} \sim R_{i,n}$.

Multi-Armed Bandits: Exemple

Exemple:

i	$r_{i,1}$	$r_{i,2}$	$r_{i,2}$	$r_{i,4}$	$r_{i,5}$	$r_{i,6}$	$r_{i,7}$	$r_{i,8}$	$\bar{R}_{i,[1-8]}$
1	15.84	11.92	15.09	14.40	14.78	12.66	13.08	17.87	14.46
2	12.27	14.12	13.93	13.25	14.28	13.79	12.69	15.79	13.77
3	11.60	10.79	9.11	10.94	11.29	11.89	11.58	9.67	10.86

Questions:

- Is it possible to design a policy maximizing the cumulative reward expectation ?

Multi-Armed Bandits: Exemple

Exemple:

i	$r_{i,1}$	$r_{i,2}$	$r_{i,2}$	$r_{i,4}$	$r_{i,5}$	$r_{i,6}$	$r_{i,7}$	$r_{i,8}$	...	$\bar{R}_{i,[1-10000]}$	Dis.
1	15.84	11.92	15.09	14.40	14.78	12.66	13.08	17.87	...	13.98	$\mathcal{N}(10, 8)$
2	12.27	14.12	13.93	13.25	14.28	13.79	12.69	15.79	...	14.01	$\mathcal{N}(12, 4)$
3	11.60	10.79	9.11	10.94	11.29	11.89	11.58	9.67	...	10.50	$\mathcal{N}(9, 4)$

Questions:

- Is it possible to design a policy maximizing the cumulative reward expectation ?

Exploration vs. Exploitation

The problem:

- One must use an arm to estimate its mean & variance,
- **Exploring non-optimal** arms instead of **exploiting** the **optimal** arm generates a **regret**,
- The goal is to build a *regret-minimizing policy* π using only past information:
  - $\bar{R}_i(n)$: empirical mean reward of arm i at step n,
  - $\mathbb{V}(\bar{R}_i)(n)$: emprical variance of the empirical mean reward for machine i at step n.

Cumulative Regret

$$regret(n) = n\mu^\star - \mu_j \sum_{1 \leq j \leq K} \mathbb{E}(t_j(n))$$

With:

- $\mu^\star = \max_i(\mu_i)$: optimal average reward,
- $t_j(n)$: numer of times arm j was played over the n first game steps.

The Upper Confidence Bound Policy (UCB1)[2]

At each game step n select arm i maximizing an over-approximation of the mean reward:

$$UCB1_i(n) = \bar{R}_i(n) + c \times \sqrt{\frac{\ln(n)}{t_i(n)}}$$

Where:

- $c > 0$: exploration/exploitation tradeoff parameter,

This policy is such that:

- $regret(n) \simeq \mathcal{O}(\ln(n))$ when $n \to \infty$,
- the probability of using a sub optimal arm goes 0 when $n \to \infty$.

UCB1: Intuition

- The *exploration term* term $c \times \sqrt{\frac{\ln(n)}{t_i(n)}}$:
  - decreases when i is played,
  - increases if an arm $j \neq i$ is played,
  - approaches 0 for all arms when $n \to \infty$,
- initially, fair exploration of arms i, j if $\bar{R}_i \simeq \bar{R}_j$,
- long term, exploitation of the arm with best mean reward.

Monte-Carlo Tree Search (MCTS)

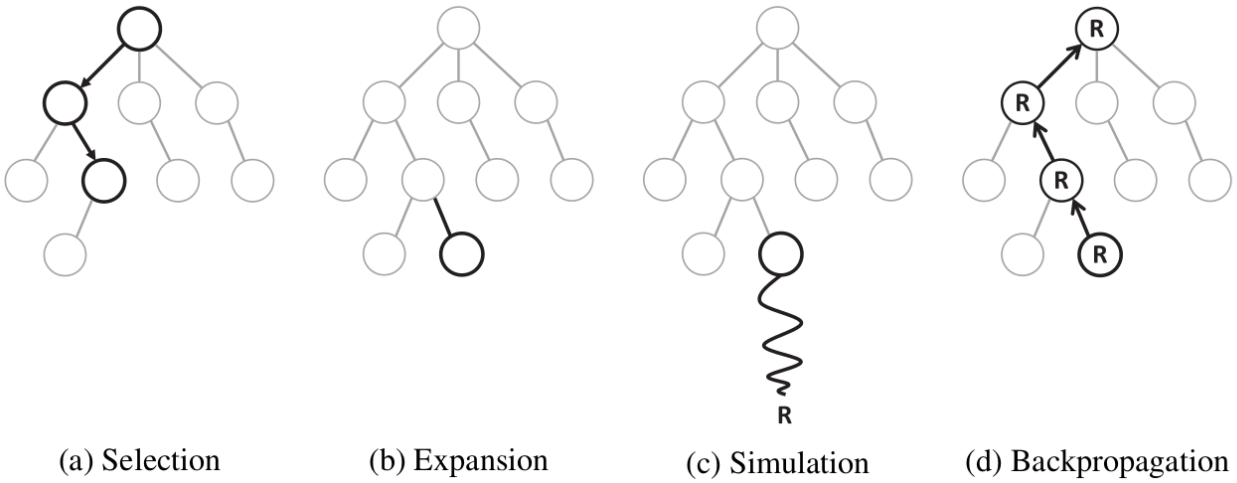


Figure: Algo. MCTS – extrait de [5]

The Upper Confidence bound applied to Trees (UCT) algorithm:

- Each node stores *UCB1* statistics $(\bar{R}_t, n)$,
- Only requires a black box simulator for R_t.

Monte-Carlo Tree Search (MCTS)

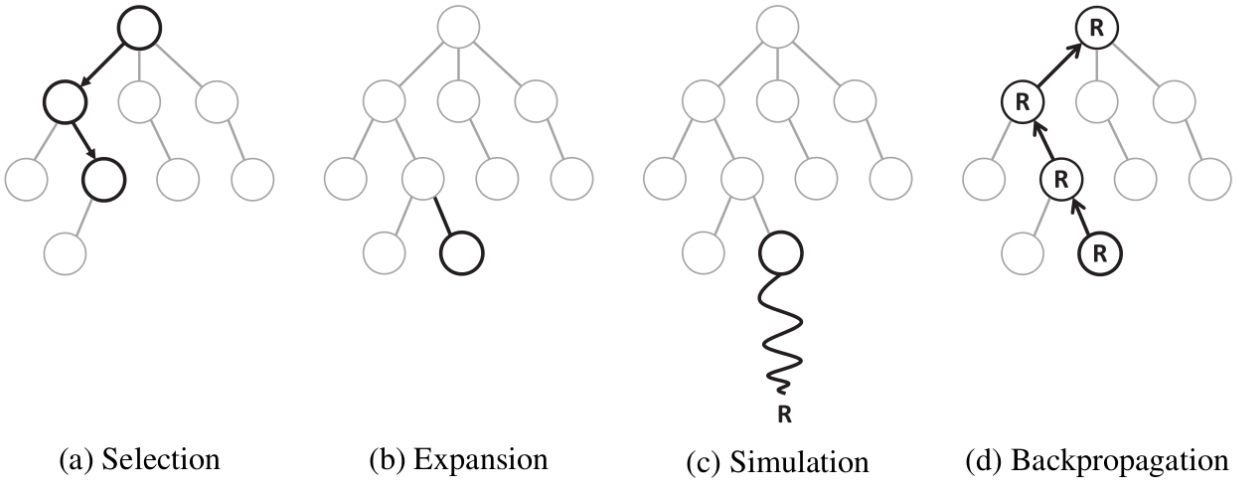


Figure: Algo. MCTS – extrait de [5]

(a) Selection: From the root, travers down following best UCB1 nodes, stop on first incomplete leaf node.

Monte-Carlo Tree Search (MCTS)

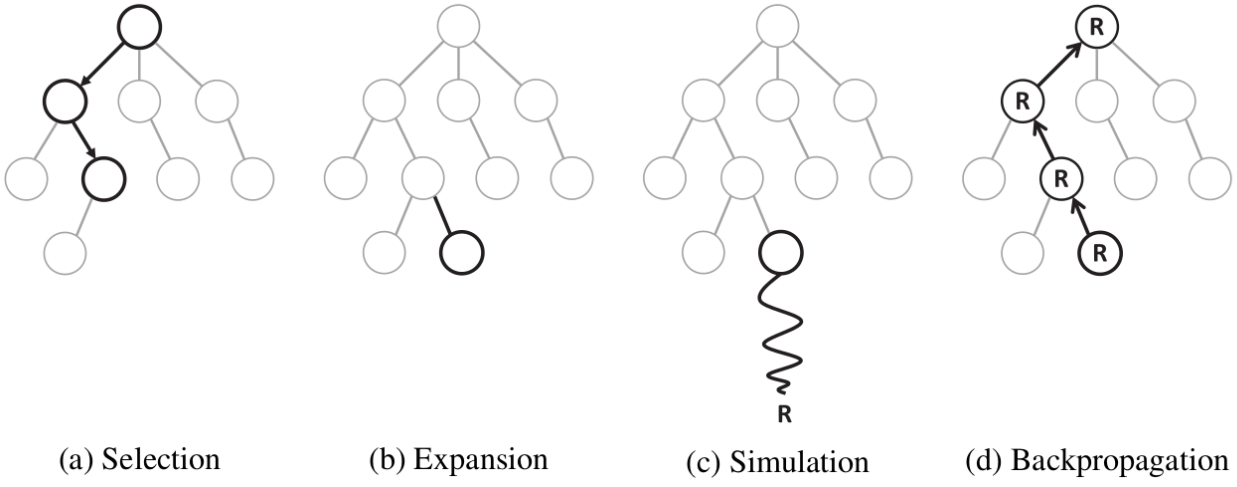


Figure: Algo. MCTS – extrait de [5]

(b) Expansion Randomly select a not-yet-explored action and add child node.

Monte-Carlo Tree Search (MCTS)

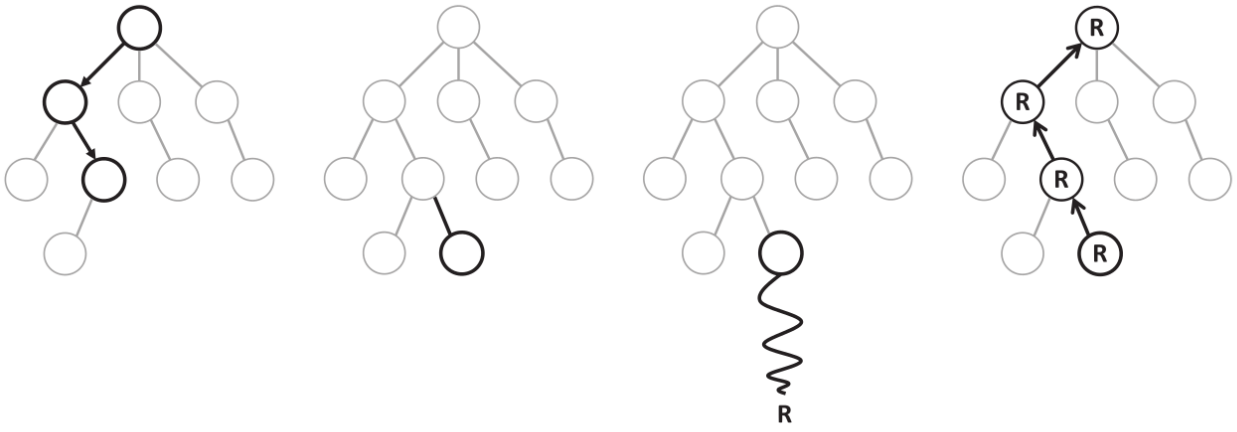


Figure: Algo. MCTS – extrait de [5]

(c) Simulation Simulate R_t to the finite horizon using a uniform random policy over A.

Monte-Carlo Tree Search (MCTS)

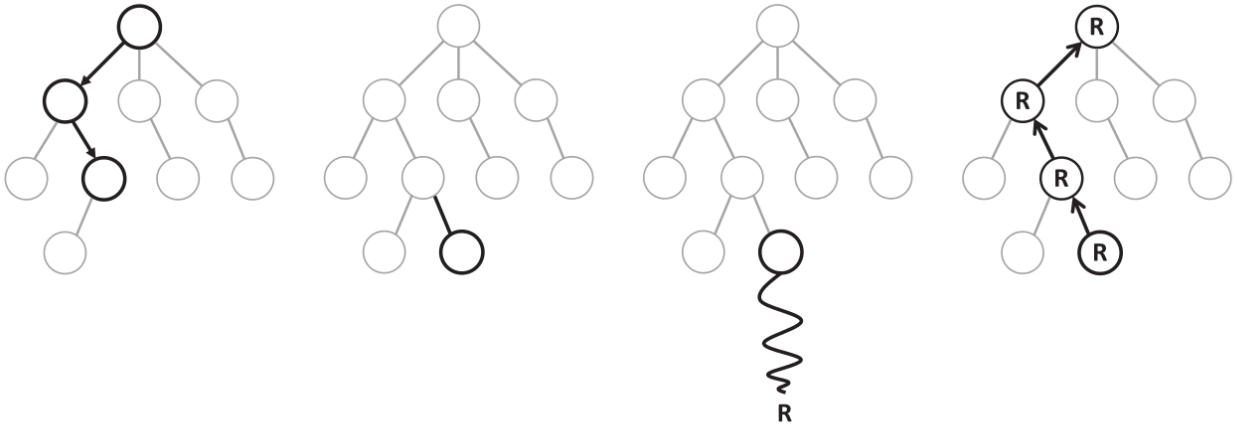


Figure: Algo. MCTS – extrait de [5]

(d) Backpropagation Update UCB1 statistics ($\bar{R}_t$, n) for each node of the current branch, then goto selection **(a) Selection**.

Outline

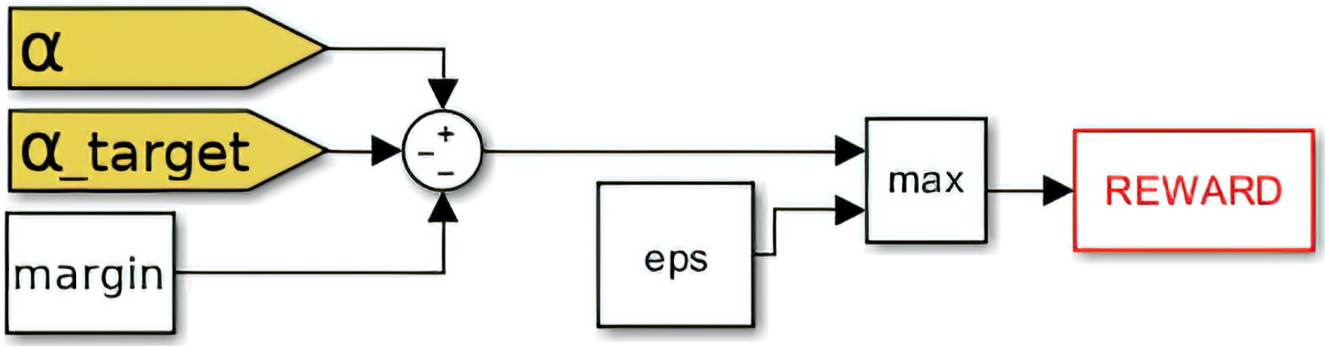
Experimentation Approach

Implementation

- A matlab implementation of UCT,
- Simulink for cumulative reward sampling,
- Properties:
  - threshold overshoots, frequential behaviour, event-based properties,
  - modeled as synchronous Simulink observers.
- UCT is run on the reference benchmark and altered benchmarks,
- A human analysis of maximum reward traces is conducted: do they activate the expected defects ?

Threshold Overshoot: Spec + Reward

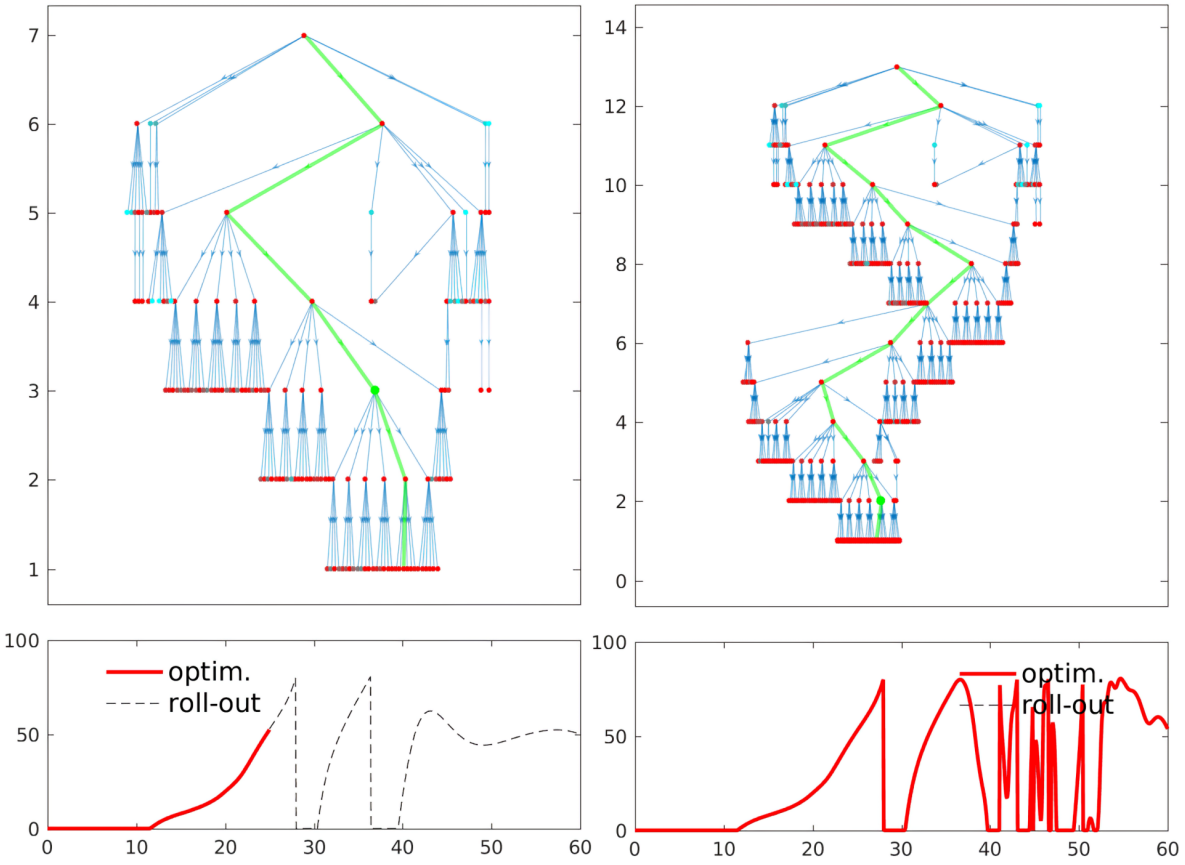
Spec: flight parameter X exceeds its target value by some given *margin*, expressed as $X \geq X_{target} + margin$.
Reward Function:



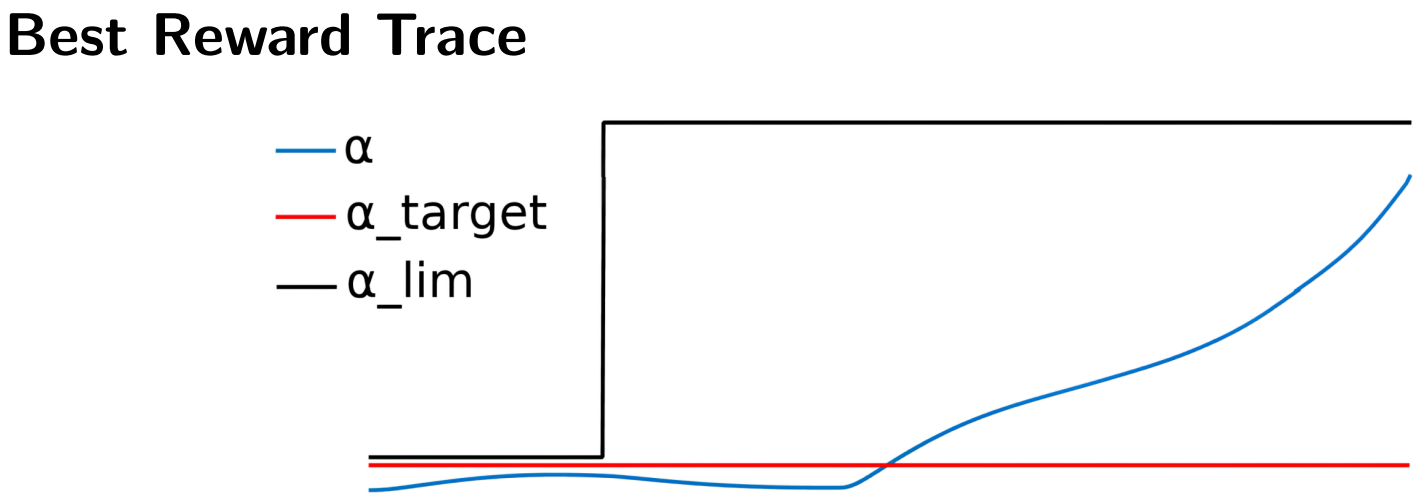
Action Space: $A = bapeng \in \{T, F\}$, $selalt = 10000 fts$, $nzcmanche \in \{neutral, half\_up, full\_up\}$, $wx = wz = 0.0$ and an action duration 5s.
MCTS Parameters: $\alpha = 0.9999$ and $p = 5$, Plan size to 30 with 30 MCTS iterations per plan step.

Threshold overshoot: Spec + Reward

Tree Search

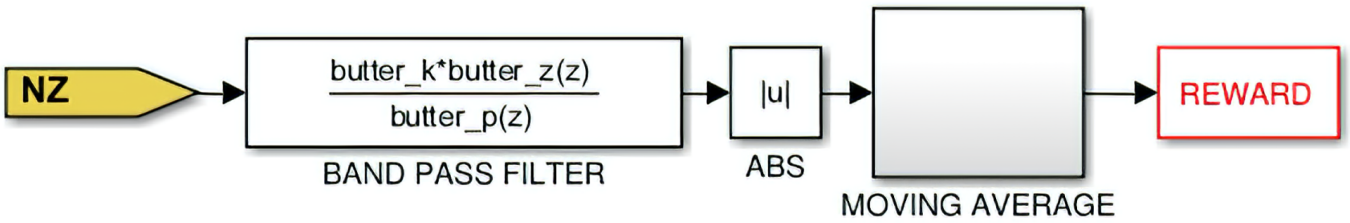


Threshold overshoot: Spec + Reward



Frequential Prop. 1

Spec: the average amplitude of n_z oscillations in a specific frequency band corresponding should be minimal.
Reward Function:



Low frequencies, corresponding to the expected response to low frequency pilot orders, and high frequencies corresponding to noise, are cut using an 11^{th} order Butterworth filter. The edge frequencies are defined according to flight control engineers knowledge. The absolute value of the filtered signal is then fed into an exponentially decaying moving average operator to obtain the final reward signal.

Frequential Prop. 1

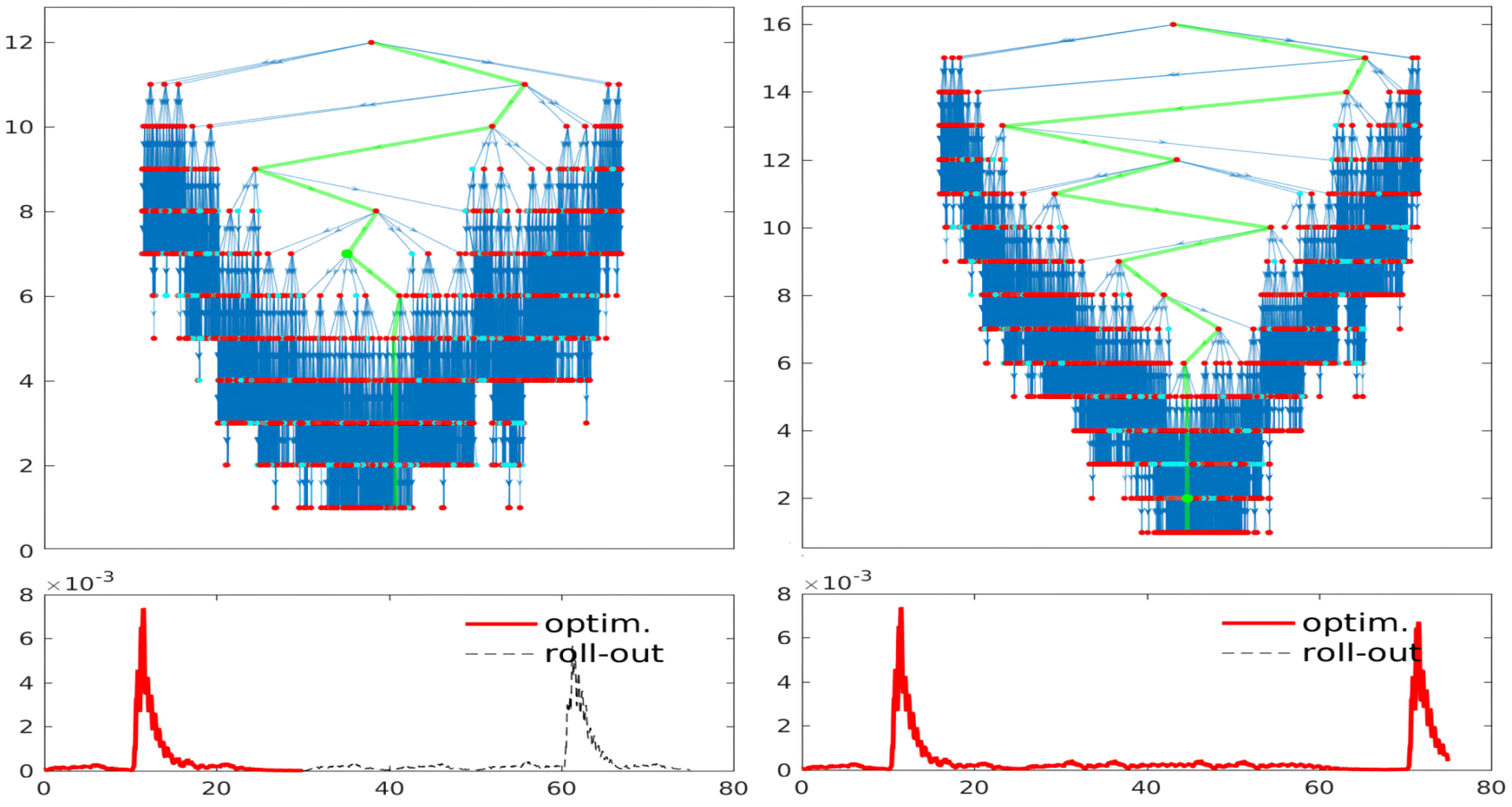
Action Space:

- $bapeng \in \{T, F\}$,
- $selalt \in \{25000fts, 28000fts\}$,
- $nzcmanche \in \{half\_down, neutral, half\_up\}$,
- $wx = wz = 0.0$
- action duration 5s.

MCTS Parameters: $\alpha = 0.9999$ and $p = 10$, plan size $= 25$, iterations $= 2880$.

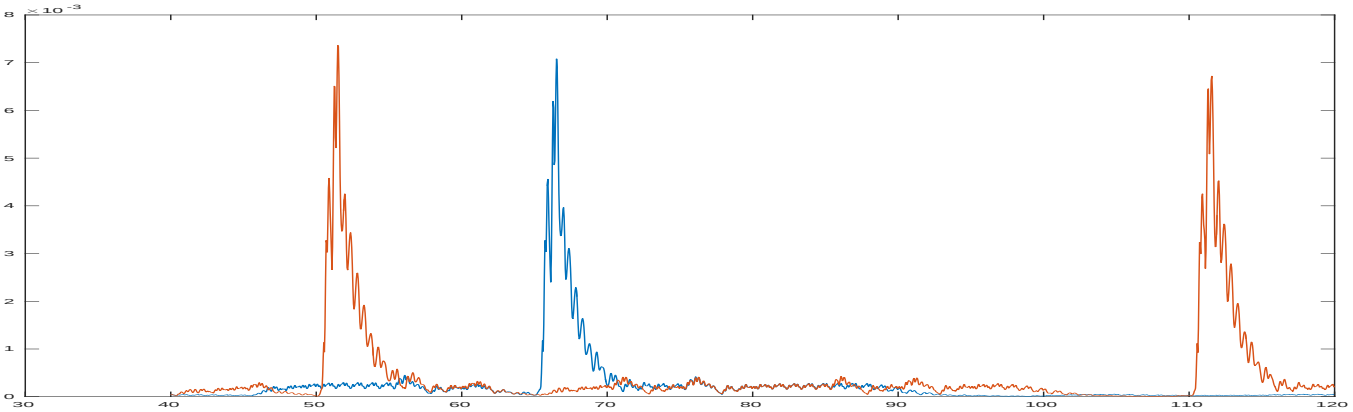
Frequential Prop. 1

Tree Search



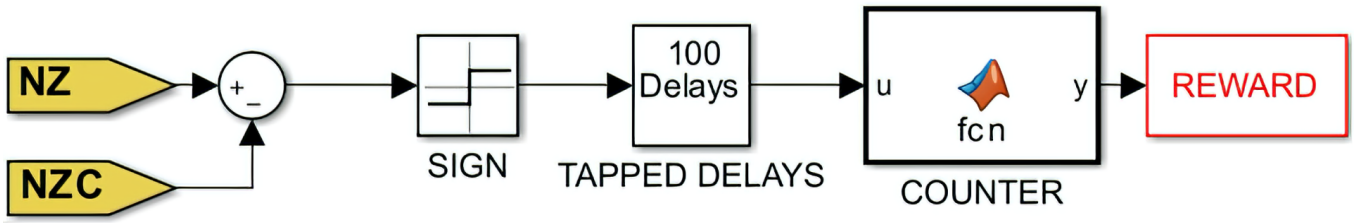
Frequential Prop. 1

Best Reward Trace



Frequential Prop. 2

Spec: n_z oscillations around the commanded n_{zc} should be minimal. Such oscillations are tolerated when they are low frequency, but can become problematic when they are high frequency and sustained over time, regardless of amplitude.
Reward Function:



Counts the number of sign inversions of $n_z - n_{zc}$ in a sliding window of a few seconds,

Frequential Prop. 2

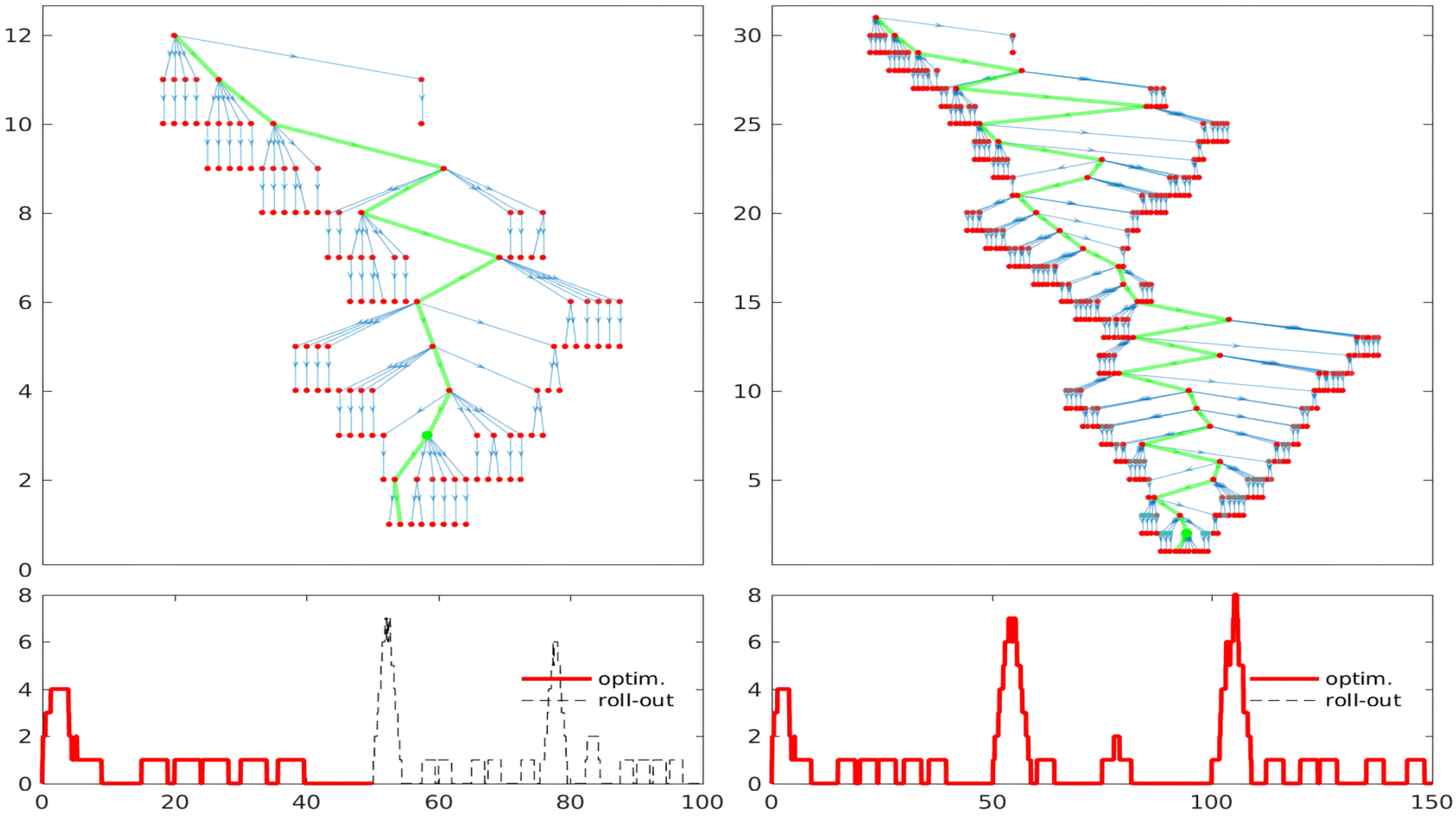
Action Space :

- $bapeng \in \{T, F\}$,
- $selalt \in \{25000fts, 28000fts\}$,
- $nzcmanche \in \{half\_down, neutral, half\_up\}$,
- $wx = wz = 0.0$
- action duration 5s.

MCTS Parameters: $\alpha = 0.9999$ and $p = 5$, plan size to 25 with 120 MCTS iterations per plan step.

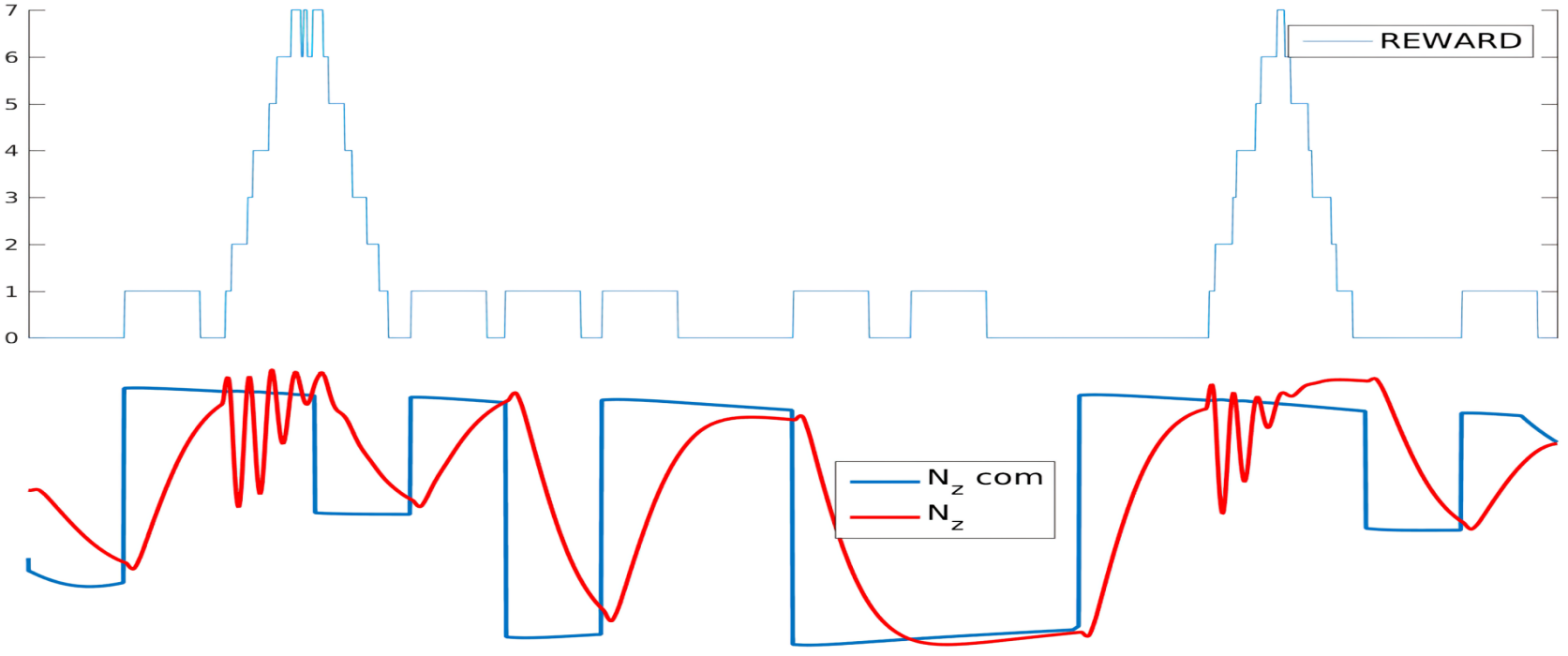
Frequential Prop. 2

Tree Search



Frequential Prop. 2

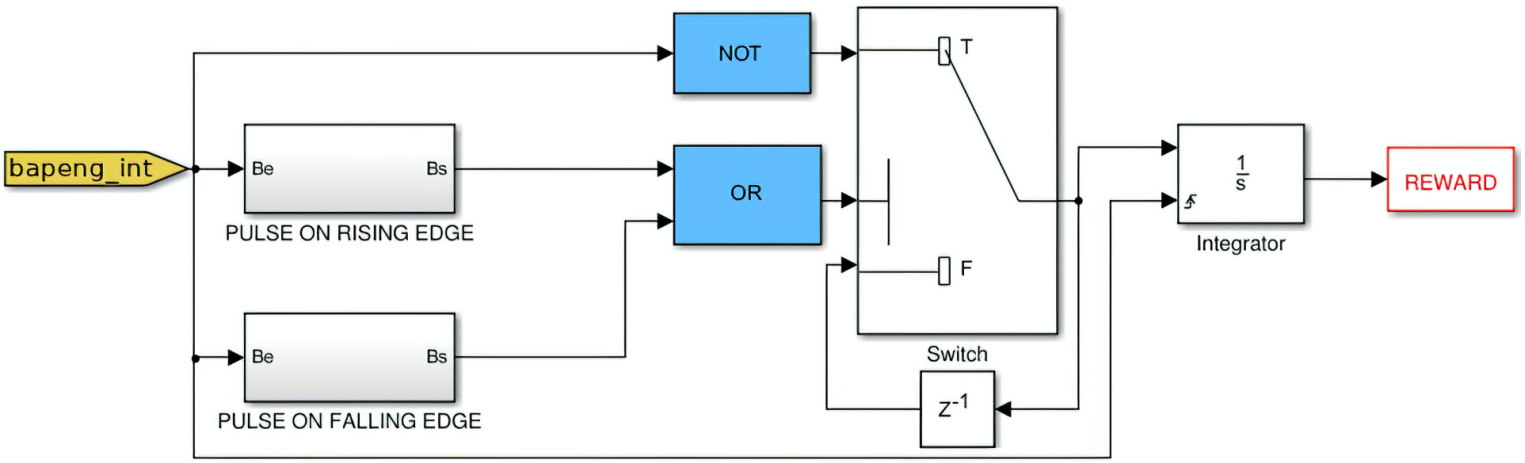
Best Reward Trace



Spurious AP Disconnection

Spec: no spurious auto-pilot disconnection in presence of wind perturbations.

Reward Function:



More precisely, we are searching for wind scenarios which cause the internal auto-pilot engagement signal *bapeng_int* to become false on a stabilized altitude in the absence of pilot intervention. We use the disconnection time of the auto-pilot as reward function.

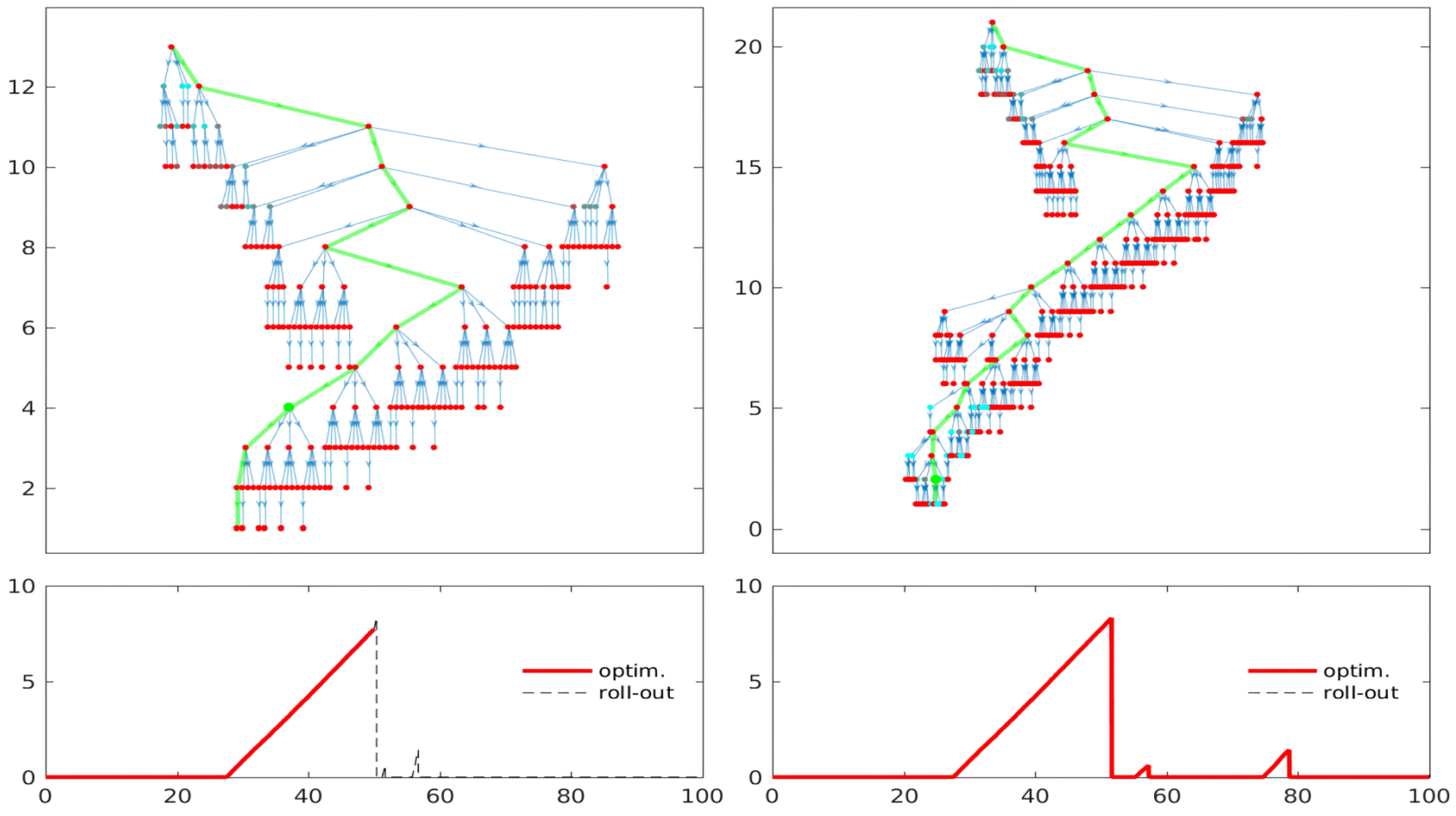
Spurious AP Disconnection

MCTS Parameters

- **Weak Wind**:
  - $bapeng = T$,
  - $selalt = 25000fts$,
  - $nzcmanche = neutral$,
  - $(wx, wz) \in \{zero, low\}^2$,
  - action duration 5s,
  - $\alpha = 0.9999$, $p = 5$,
  - plan size 15, 20 MCTS iterations per plan step.
- **Strong Wind**:
  - $bapeng = T$,
  - $selalt = 25000fts$,
  - $nzcmanche = neutral$,
  - $(wx, wz) \in \{zero, low, medium, high, very\_high\}^2$,
  - action duration 5s,
  - $\alpha = 0.9999$, $p = 5$, plan size 15, 25 MCTS iterations per plan step.

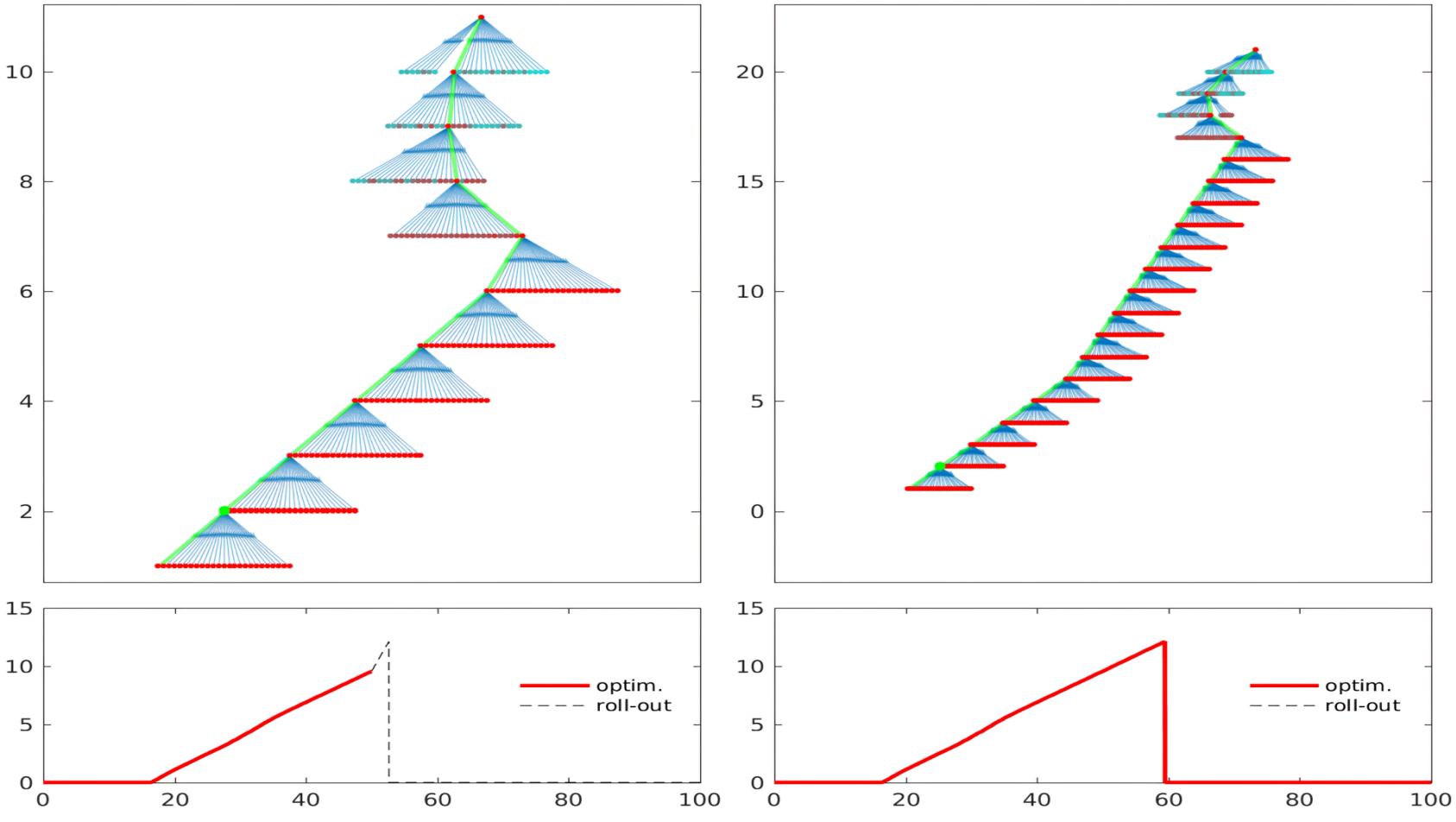
Spurious AP Disconnection

Tree Search Weak Wind



Spurious AP Disconnection

Tree Seach Strong Wind



Outline

State of the Art 2017–2018

Our work is directly inspired from the following papers:

- "Time-Staging Enhancement of Hybrid System Falsification" [8] Continuous action space, eager time-staging + hill-climbing local search, closed-loop simulink models, automotive domain.
- "Two-Layered Falsification of Hybrid Systems Guided by Monte Carlo Tree Search" [9] Same as above with an MCTS harness on top.

State of the Art 2017–2018

- "Falsification of Cyber-Physical Systems Using Deep Reinforcement Learning" [1] Metric interval Temporal Logic or PSL + Deep-RL (Asynchronous Advantage Actor-Critic (A3C), Double Deep Q Network (DDQN)) to optimise for a robust temporal logic property.
- "Adaptive Stress Testing: Finding Failure Events with Reinforcement Learning" [7] MCTS pour falsification, differential stress testing: MCTS used to maximize the divergence between two concurrent system revisions.

Outline

Conclusion

Results

- Surprisingly good results,
- All benchmarks were successfully analyzed,
- Generated traces allowed to pinpoint unsafe behaviours.

Limitations

- Reward engineering,
- Discrete action space and duration selection.

Perspectives

- Algorithmic evolutions:
  - Stochastic tree search policy instead of UCB1,
  - Introduce Kernel-Regression estimators to:
    - handle continuous action spaces,
    - use as a similarity measure of reached states to share subtrees,
    - reduce the rollout budget by predicting rollout values using KR.
- Study compilation of hybrid dataflow models to HW accelerators (GPU, FPGA) to speed-up rollout.


Takumi Akazaki, Shuang Liu, Yoriyuki Yamagata, Yihai Duan, and Jianye Hao.
Falsification of cyber-physical systems using deep reinforcement learning.
In Klaus Havelund, Jan Peleska, Bill Roscoe, and Erik P. de Vink, editors, *Formal Methods - 22nd International Symposium, FM 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 15-17, 2018, Proceedings*, volume 10951 of *Lecture Notes in Computer Science*, pages 456–465. Springer, 2018.


Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer.
Finite-time analysis of the multiarmed bandit problem.
Machine Learning, 47(2-3):235–256, 2002.


Rémi Coulom.
Efficient selectivity and backup operators in monte-carlo tree search.

In H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. Donkers, editors, *Computers and Games, 5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. Revised Papers*, volume 4630 of *Lecture Notes in Computer Science*, pages 72–83. Springer, 2006.

Alexandre Donzé and Oded Maler.
Robust satisfaction of temporal logic over real-valued signals.
In *Formal Modeling and Analysis of Timed Systems - 8th International Conference, FORMATS 2010, Klosterneuburg, Austria, September 8-10, 2010. Proceedings*, pages 92–106, 2010.

Steven James, George Konidaris, and Benjamin Rosman.
An analysis of monte carlo tree search.
In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial*

Intelligence, February 4-9, 2017, San Francisco, California, USA., pages 3576–3582. AAAI Press, 2017.


Levente Kocsis and Csaba Szepesvári.
Bandit-based monte-carlo planning.
In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2006.


Ritchie Lee, Ole J. Mengshoel, Anshu Saksena, Ryan Gardner, Daniel Genin, Joshua Silbermann, Michael P. Owen, and Mykel J. Kochenderfer.
Adaptive stress testing: Finding failure events with reinforcement learning.
CoRR, abs/1811.02188, 2018.

Zhenya Zhang, Gidon Ernst, Ichiro Hasuo, and Sean Sedwards.

Time-staging enhancement of hybrid system falsification.
In *3rd Workshop on Monitoring and Testing of Cyber-Physical Systems, MT@CPSWeek 2018, Porto, Portugal, April 10, 2018*, pages 3–4. IEEE, 2018.

Zhenya Zhang, Gidon Ernst, Sean Sedwards, Paolo Arcaini, and Ichiro Hasuo.
Two-layered falsification of hybrid systems guided by monte carlo tree search.
IEEE Trans. on CAD of Integrated Circuits and Systems, 37(11):2894–2905, 2018.