RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

SCIENCES SORBONNE UNIVERSITÉ

ONERA
THE FRENCH AEROSPACE LAB

Model-Based Deep Reinforcement Learning for Fixed-Wing UAV Attitude Control with Prior Physics Knowledge

Titouan Guerin[1,2], Pierre Fournier[1], Julien Marzat[1], Olivier Sigaud[2]
[1] ONERA, [2] Sorbonne Université

Context

- Fixed-wing UAV, **attitude control**
- Away from level flight : **nonlinear, coupled**
- Autopilots **not robust** to those conditions

Our positioning

Equations : **known**

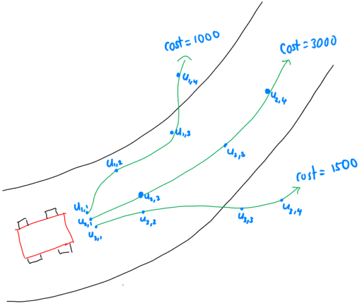
Coefficients : **approximate**

1. Train a controller **inside** a physics-informed model
2. **Correct** the physics when inaccurate
3. **Plan** on top of the controller

Background

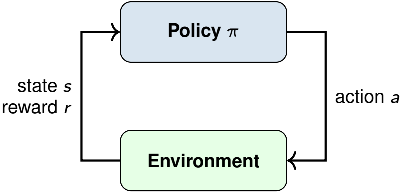
Two ways to build a controller

Planning *(MPC)*



Needs a **model**

Data-driven *(RL)*



Policy π + critic Q

$\rightarrow$ *We combine the two.*

From Model-Free to Model-Based RL

Model-Free

No model

Many interactions

Model-Based

Learns $\hat{f}(s_t, a_t) \rightarrow s_{t+1}$

Sample-efficient

Our approach

Train the controller **entirely inside** the learned model.

Physics-informed models

APHYNITY (2021)

$$\dot{X} = \underbrace{F_p(X)}_{\text{prior}} + \underbrace{F_a(X)}_{\text{residual}}$$

no action

PhIHP (2024)

$$\dot{s} = F_p(s, {\color{red}a}) + F_a(s, {\color{red}a})$$

controlled

Validated on pendulum, cartpole, acrobot

$\rightarrow$ *simple, and physics **exactly known**. An aircraft is neither.*

Our work

Our dynamics model

$$\dot{s} = \underbrace{F_p^{\theta_p}(s, a)}_{\text{physics prior}} + \underbrace{F_a^{\theta_a}(s, a)}_{\text{learned residual}} \qquad s_{t+1} = s_t + \int_t^{t+\Delta t} \dot{s} \, d\tau$$

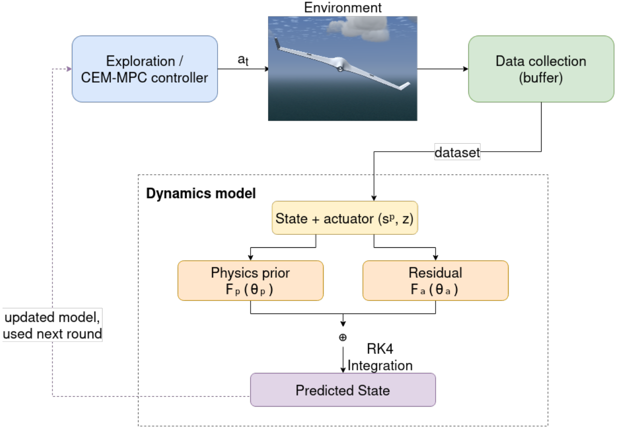
The prior is **approximate by design**

and its coefficients θ_p are **never exact**

$\rightarrow$ *so we let the optimizer* ***correct*** *them.*

Learning the dynamics model

- Explore, collect, **retrain**, repeat
- Then the model **replaces** the environment



Results

Experimental setup

Training and planning : **our model** | Evaluation : **the simulator**

Conditions

- JSBSim, **full physics**
- No wind, no turbulence

Metrics

- **Settled** error, in degrees
- $< 1^\circ$ near-perfect
- Inference : **10 ms** budget

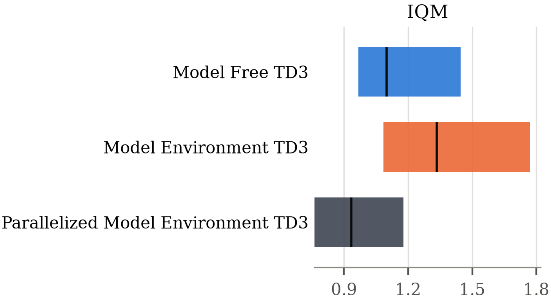
Protocol

- **10 seeds** per condition
- Easy and hard targets split
- Median / IQM / mean, 95% CI

$\rightarrow$ *A PID already handles easy targets.* ***Hard targets is what we are interested in.***

Q1 : training inside the model

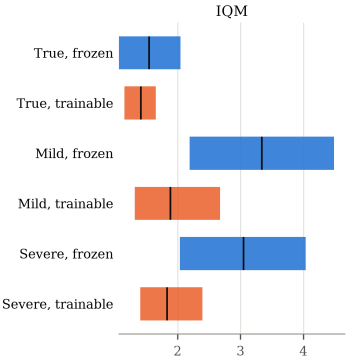
- Rollouts of **hundreds of steps**, single model
- No simulator interaction needed
- **Parallelized model** $\rightarrow$ best overall



Settled attitude error (deg), 10 seeds, 95% CI. Lower is better.

Q2 : correcting the physics prior

- Accurate prior → tuning changes little
- Wrong prior → tuning recovers **most** of the loss
- **Takeaway :** Approximate physics is **enough** to learn a good controller.



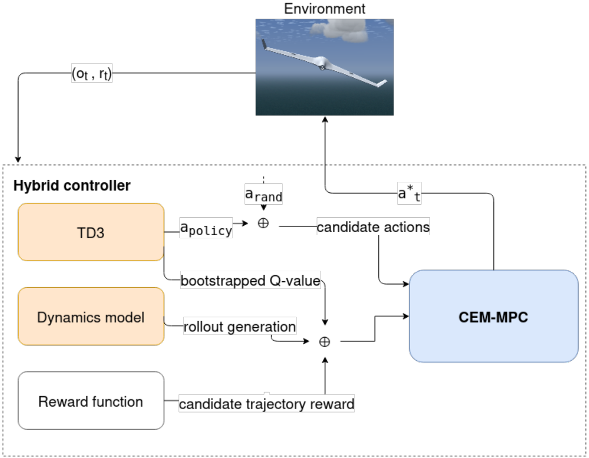
Settled attitude error hard targets (deg), 10 seeds per model, 95% CI. Lower is better.

Q3 : planning on top of the controller

MPC $+$

- policy π
  seeds the search
- critic Q
  sees past the horizon

Which one matters most ?



Which component drives the gain ?

All six prior conditions

Controller	All	Easy	**Hard**
Policy alone	1.17	0.59	1.91
MPC + π	19.57	3.74	41.44
MPC + Q	0.82	0.48	1.30
Hybrid	**0.77**	**0.44**	**1.21**

-37% on hard targets

True prior only

Controller	All	Easy	**Hard**
Policy alone	0.97	0.55	1.55
MPC + π	19.76	4.79	40.49
MPC + Q	0.67	0.50	**0.90**
Hybrid	**0.64**	**0.43**	0.92

-41% on hard targets

IQM settled attitude error (deg)

Takeaway

The **critic** makes short-horizon planning work, the policy proposals do not.
Cost : **21 ms/step** against a 10 ms budget.

Conclusion

- Possible to train a RL controller **entirely inside** a dynamics model
- **Correcting the physics prior** recovers most of what an inaccurate one costs
- Planning helps for attitude control, and the **critic** is what drives the improvement

*Our method only requires a **physics prior**. Not specific to fixed-wing flight.*

Next : PhD at ONERA and ISIR

- Waypoint tracking
- Reducing inference cost
- Joint learning of model and controller
- Hierarchical control

Thank you for your attention！

Questions？

Extra Info

TD3

$$y = r + \gamma \min_{j \in \{1,2\}} Q_{\theta'_j}\left(o', \pi_{\phi'}(o') + \epsilon\right)$$

- Actor π $+$ **twin** critics Q
- Minimum of the two $\rightarrow$ helps with value over-estimation

Prior degradation

- 10 coefficients perturbed : **mild** 25–50%, **severe** 75–100%
- Chosen : the ones that are **hard to measure**