A Method for Accelerated Simulations of Reinforcement Learning Tasks of UAVs in AirSim

Alberto Musa, Luca Zanatta, Francesco Barchi, Andrea Bartolini, Andrea Acquaviva

Università di Bologna
Viale del Risorgimento, 2,
Bologna, Italy
alberto.musa@unibo.it, luca.zanatta3@unibo.it, francesco.barchi@unibo.it,
a.bartolini@unibo.it, andrea.acquaviva@unibo.it

About me

Alberto Musa

Researcher assistant in the Department of Electrical, Electronic, and Information of the Alma Mater Università di Bologna.

Research topic:

- Optimization of autonomous Cyber Physical Systems (CPS), whit more focus on Unmanned Aerial Vehicles (UAVs), applied to Reinforcement Learning (RL) algorithms in real and simulated environments.

Introduction – UAV and Autonomous Navigation

Mavic 2 PRO from DJI

Micro UAV from BitCraze

Unmanned Aerial Vehicles (**UAVs**) are complex robotic platforms (or Cyber-Physical Systems - **CPS**) designed for *flight without a human pilot*.

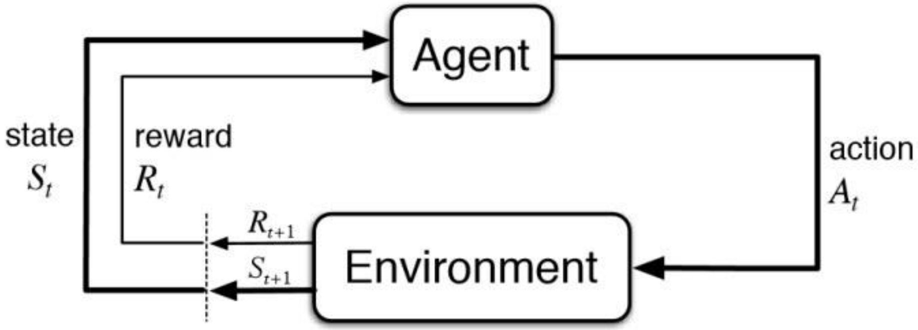
Introduction – UAV and Autonomous Navigation

Mavic 2 PRO from DJI

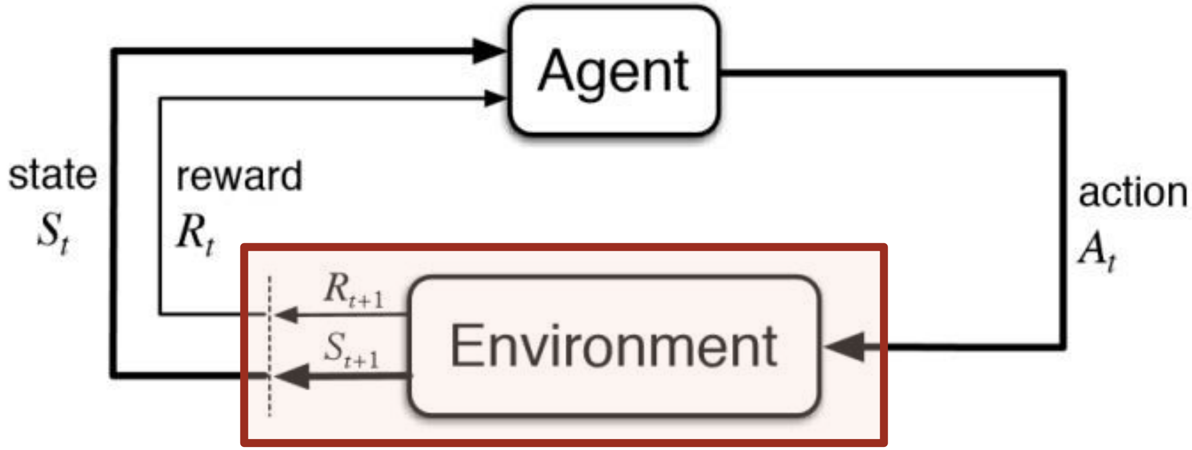
Micro UAV from BitCraze

Unmanned Aerial Vehicles (**UAVs**) are complex robotic platforms (or Cyber-Physical Systems - **CPS**) designed for *flight without a human pilot*.

Reinforcement Learning (***RL***) algorithms enable ***UAVs*** to perform *autonomous control tasks* such as obstacle avoidance.

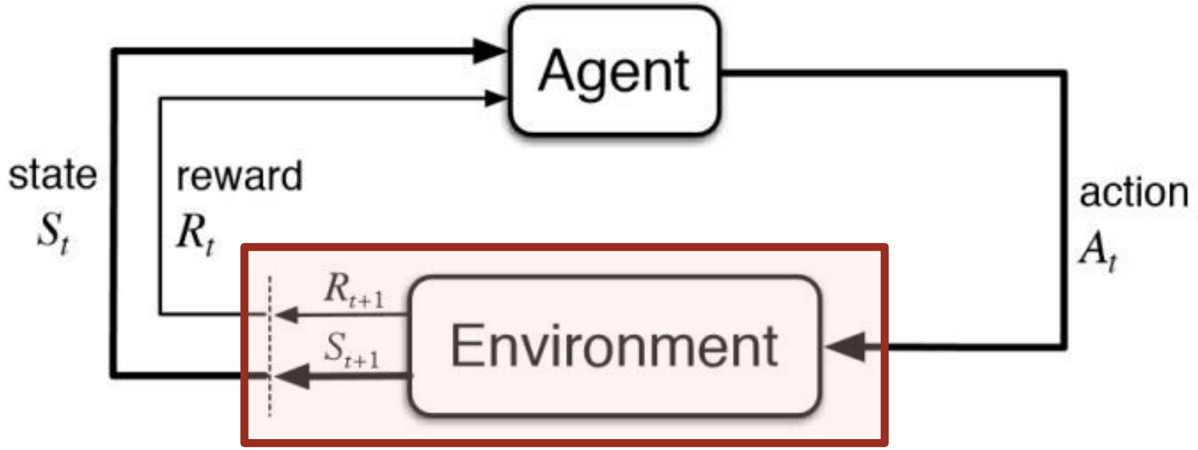


Introduction – UAV and Autonomous Navigation



The RL loop requires to *simulate both the environment and the CPS agent*. RL *extracts knowledge from the interaction* between the environment and the agent.

Introduction – UAV and Autonomous Navigation



The RL loop requires to *simulate both the environment and the CPS agent*. RL *extracts knowledge from the interaction* between the environment and the agent.

RL training requires *performing the task* (called game) *repeatedly until the task is learned*. This can easily require *thousands or millions of games*.

Introduction – UAV and Autonomous Navigation

Simulators *replicate the agent capabilities* (e.g., the **UAV** flight) *and real environments*

- real-world physics rules, and perception capabilities.
- forces that act in the simulated scenario (gravity, rotors actuation, collisions, etc.)

Introduction – UAV and Autonomous Navigation

Simulators *replicate the agent capabilities* (e.g., the **UAV** flight) *and real environments*

- real-world physics rules, and perception capabilities.
- forces that act in the simulated scenario (gravity, rotors actuation, collisions, etc.)

Most suitable simulators for UAV applications				
	AirSim	*Flightmare*	*Gazebo*	*Webots*
Photorealism	**x**	**x**		
Co-Simulation	**x**		**x**	**x**

Relevant simulator features are:

- **Co-Simulation:** rapid flight control and RL-trained solution rapid prototyping.
- **Photorealism:** if the task to be performed by the agent leverages camera sensors, accurate and photorealistic rendering of the scene becomes mandatory.

Introduction – UAV and Autonomous Navigation

Simulators *replicate the agent capabilities* (e.g., the **UAV** flight) *and real environments*

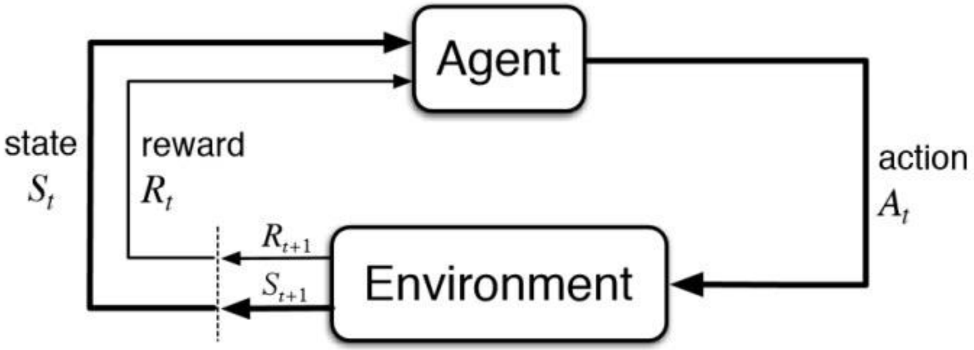
- real-world physics rules, and perception capabilities.
- forces that act in the simulated scenario (gravity, rotors actuation, collisions, etc.)

Most suitable simulators for UAV applications				
	AirSim	*Flightmare*	*Gazebo*	*Webots*
Photorealism	x	x		
Co-Simulation	x		x	x

Relevant simulator features are:

- **Co-Simulation:** rapid flight control and RL-trained solution rapid prototyping.
- **Photorealism:** if the task to be performed by the agent leverages camera sensors, accurate and photorealistic rendering of the scene becomes mandatory.

Challenges



RQ1: Is it *possible* to accelerate RL training for UAVs in AirSim?

RQ2: What is the implication in terms of performance of the simulated drone flight, interaction with the environment, and trained RL algorithm?

AirSim

Client APIs

Used to interact with the built-in autopilot.

- Impose the desired waypoint through an **asynchronous command**
  - Enforcing *synchronous control*
- Gather the **UAV agent state**

Client API

AirSim

Client APIs
Used to interact with the built-in autopilot.

- Impose the desired waypoint through an **asynchronous command**
  - Enforcing *synchronous control*
- Gather the **UAV agent state**

Built-in Flight Control Systems (**Simple Flight** Controller)

- *stoppable clock* to pause the simulation at any point.

AirSim

Client APIs

Used to interact with the built-in autopilot.

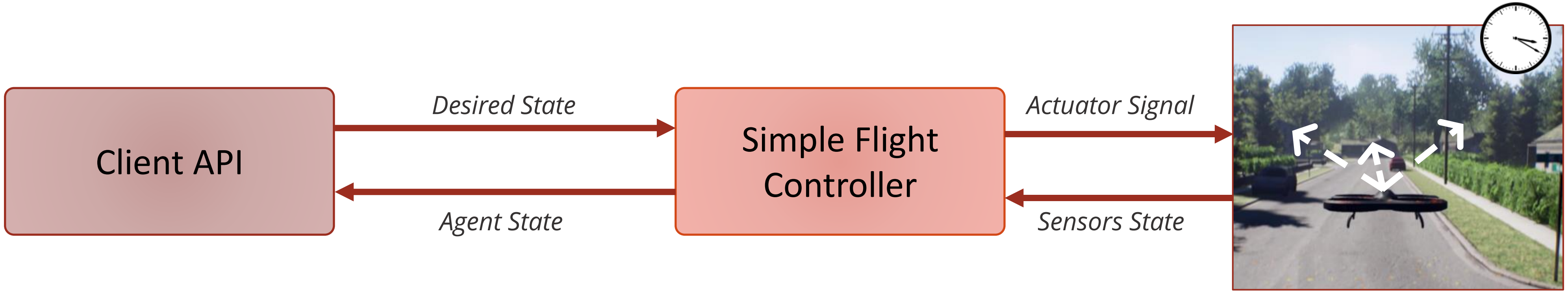
- Impose the desired waypoint through an **asynchronous command**
  - Enforcing <u>*synchronous control*</u>
- Gather the **UAV agent state**

Built-in Flight Control Systems (**Simple Flight** Controller)

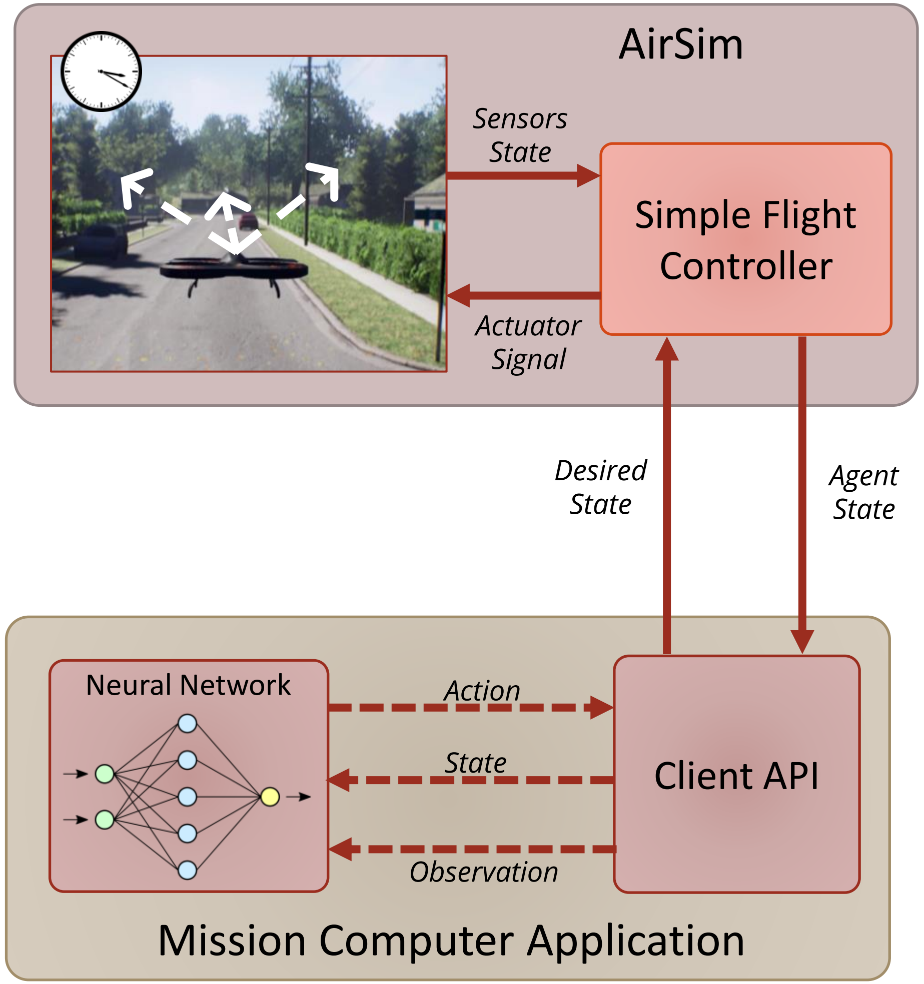
- <u>*stoppable clock*</u> to pause the simulation at any point.

ClockSpeed setting

changes the ratio between the simulation and the wall clock time



AirSim Default Synchronous Command



Client *gathers multiple UAV states during the command execution* by issuing a concurrent *thread that pauses the simulation at specific time intervals.*

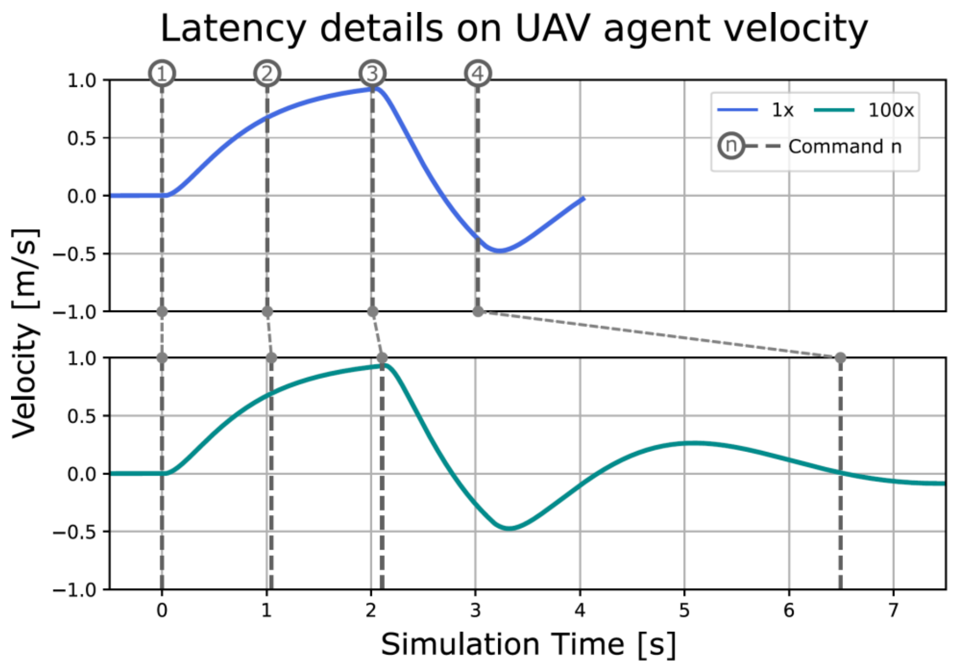
AirSim Default Synchronous Command (ADSC).

Mission Computer Application (MCA)

- **Deep RL** aims at *solving complex robotic tasks by mimicking human training behavior* with the use of **Neural Networks.**
  - ✓ Uses **Client APIs** to issue commands and observe the **UAV** state

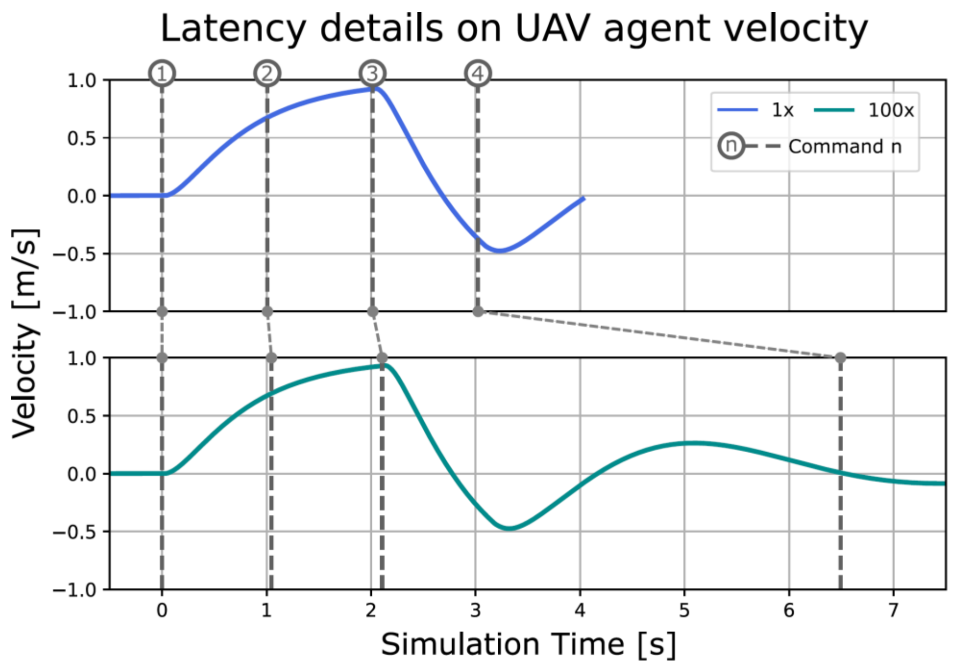
AirSim Setup Profiling

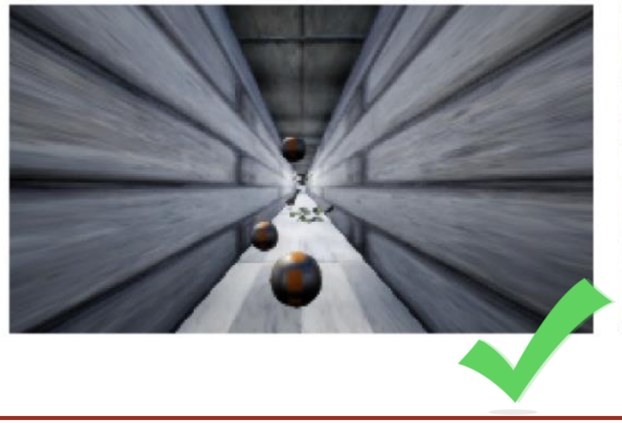
- *Callback timing of three consecutive commands with **ADSC***.
- *Accelerated simulation is subject to command delays* due to communication between simulation, Simple flight, and Client API.
- *Delay in the move command increases with the accelerator factor* ⇒ during the command delay, *the simulator continues to evaluate the **UAV***.



AirSim Setup Profiling

- *Callback timing of three consecutive commands with **ADSC***.
- *Accelerated simulation is subject to command delays* due to communication between simulation, Simple flight, and Client API.
- *Delay in the move command increases with the accelerator factor* ⇒ during the command delay, *the simulator continues to evaluate the **UAV***.





- In an accelerated game *the simulator rendering time is lowered*, making *hard to complete the images.*
- *The result images may be perturbed.*

Speed-up Effects Mitigation – Latencies

Time-Controlled Simulation Command (TCSC).

- uses the *stoppable clock* of the Simple Flight to control the simulation until the command expires.
- is implemented by *periodic stimulation interrupts* interleaved with command expired-time checks.

Speed-up Effects Mitigation – Latencies

Time-Controlled Simulation Command (TCSC).

- uses the *stoppable clock* of the Simple Flight to control the simulation until the command expires.
- is implemented by *periodic stimulation interrupts* interleaved with command expired-time checks.

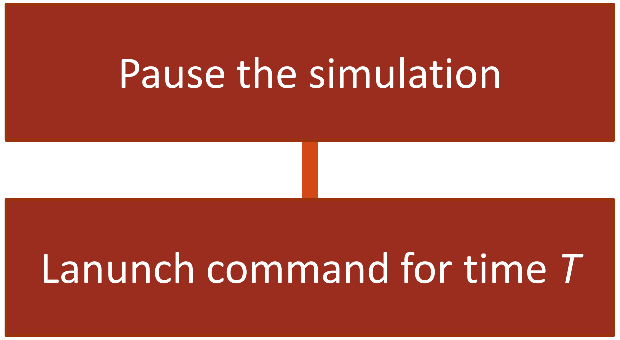
Pause the simulation

Simulation pause

Speed-up Effects Mitigation – Latencies

Time-Controlled Simulation Command (TCSC).

- uses the *stoppable clock* of the Simple Flight to control the simulation until the command expires.
- is implemented by *periodic stimulation interrupts* interleaved with command expired-time checks.



T = imposed time;

Simulation pause

1. Launch asynchronous command for time *T*

Speed-up Effects Mitigation – Latencies

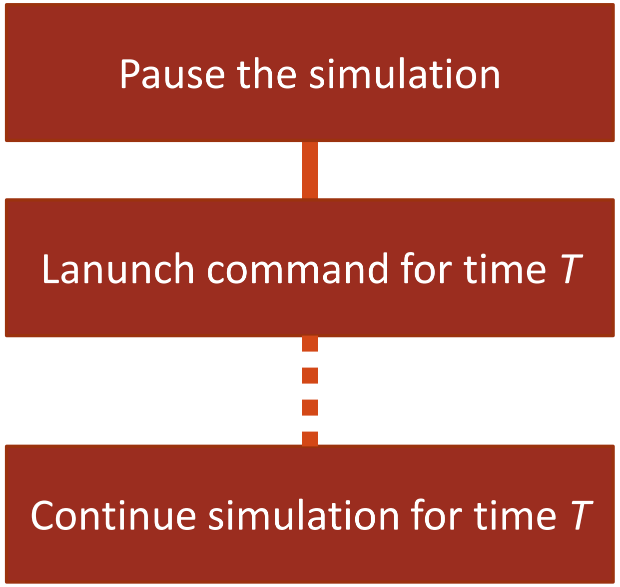
Time-Controlled Simulation Command (TCSC).

- uses the *stoppable clock* of the Simple Flight to control the simulation until the command expires.
- is implemented by *periodic stimulation interrupts* interleaved with command expired-time checks.

T = imposed time;

Simulation pause

1. Launch asynchronous command for time *T*
2. Continue simulation for time *T*



Speed-up Effects Mitigation – Latencies

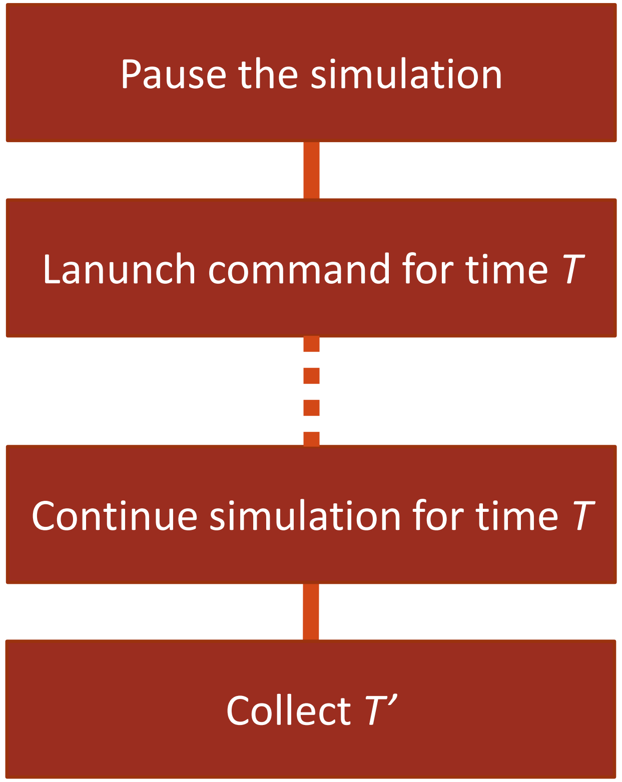
Time-Controlled Simulation Command (TCSC).

- uses the *stoppable clock* of the Simple Flight to control the simulation until the command expires.
- is implemented by *periodic stimulation interrupts* interleaved with command expired-time checks.

T = imposed time; T' = effective simulation time

Simulation pause

1. Launch asynchronous command for time *T*
2. Continue simulation for time *T*
3. Collect simulation time *T'*



Speed-up Effects Mitigation – Latencies

Time-Controlled Simulation Command (TCSC).

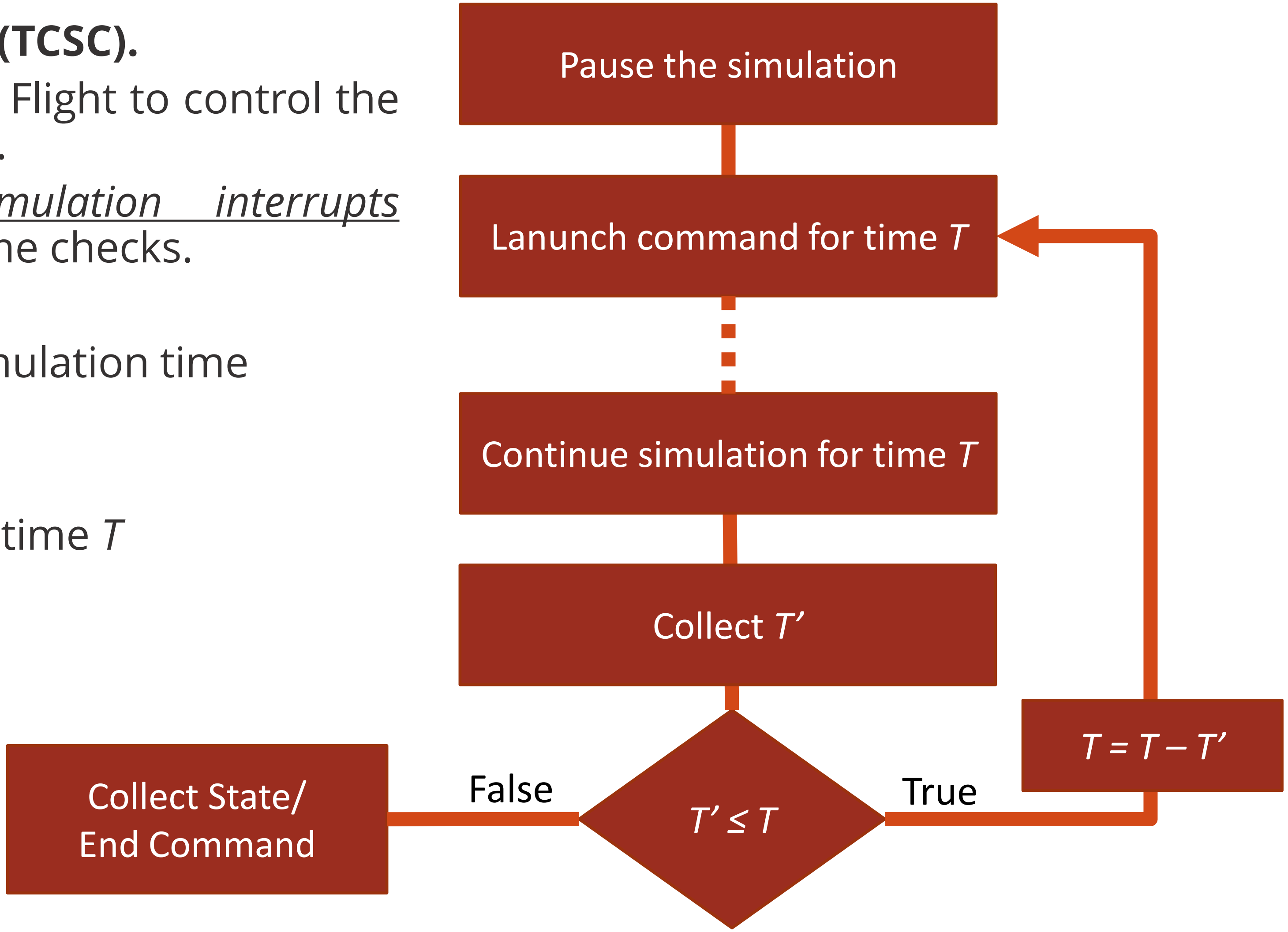
- uses the <u>*stoppable clock*</u> of the Simple Flight to control the simulation until the command expires.
- is implemented by <u>*periodic stimulation interrupts*</u> interleaved with command expired-time checks.

T = imposed time; T' = effective simulation time

Simulation pause

1. Launch asynchronous command for time *T*
2. Continue simulation for time *T*
3. Collect simulation time *T'*
4. If *T'≤T*: *T=T-T'*
   return to 1.

   Otherwise: Collect State *or* End Command



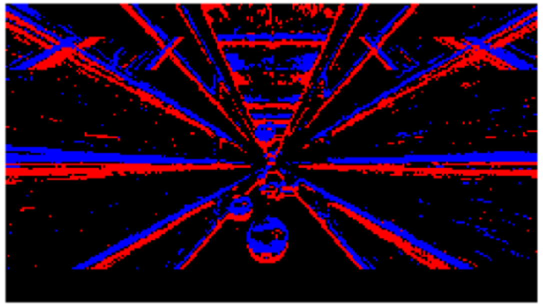
Characterization Methodology

1. Accuracy of trajectories (**Deterministic Path 1**).
   - **Spatiotemporal Linear Combined (STLC)** distance**:** score between 0 and 2, where *2 is the maximum value indicating that the trajectories are entirely overlapped*.
   - *reference is the **ADSC** real-time trajectory*

Characterization Methodology

1. Accuracy of trajectories (**Deterministic Path 1**).
   - **Spatiotemporal Linear Combined (STLC)** distance: score between 0 and 2, where *2 is the maximum value indicating that the trajectories are entirely overlapped*.
   - *reference is the **ADSC** real-time trajectory*

2. Visual perception accuracy (**Deterministic Path 2**).
   - *Is the quality of the image related to the error in **UAV** gathering coordinates?*
     - *Euclidean distance* between the coordinates (reference and accelerated simulation);
     - The *difference in percentage of image pixels* between the camera image taken in the accelerated and real-time simulation ⇒ **Event Dissimilarity (ED).**



Characterization Methodology

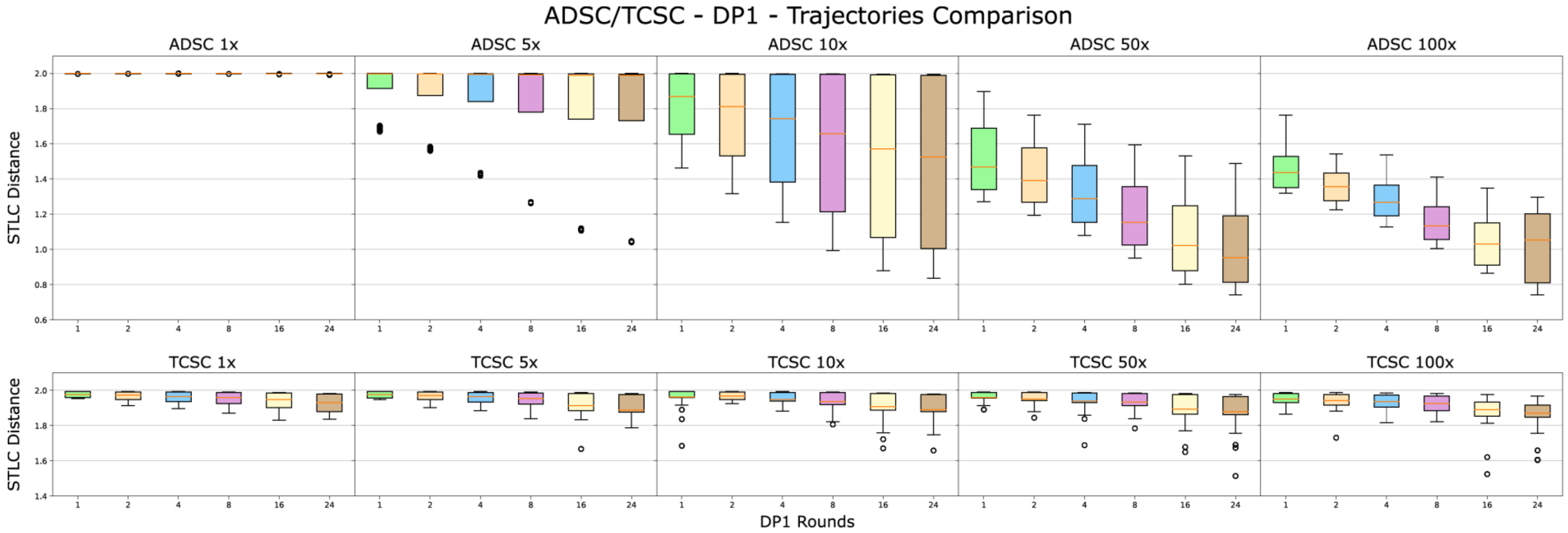
3. Impact of the agent performance trained with RL.

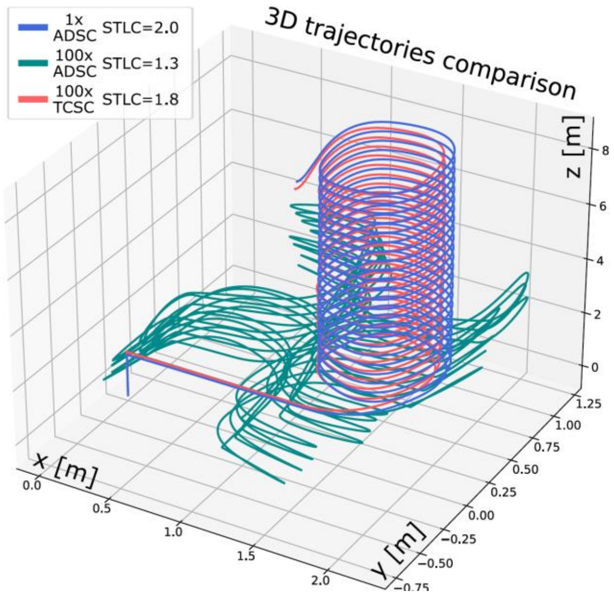
Replicate an obstacle avoidance task in a lane (**AirSim RL Environment**) using event camera images

- **RL** training procedure of 5000 games.
- **RL** inference in real-time with **ADSC.**
- Evaluation of the **UAV** *success ratio* (number of times the **UAV** reaches the end of the environment without collisions) in inference over 100 games.
  - interchangeability of commands when the simulation happens in real-time.
  - comparison of the UAV success ratio computed with different AirSim acceleration factors.

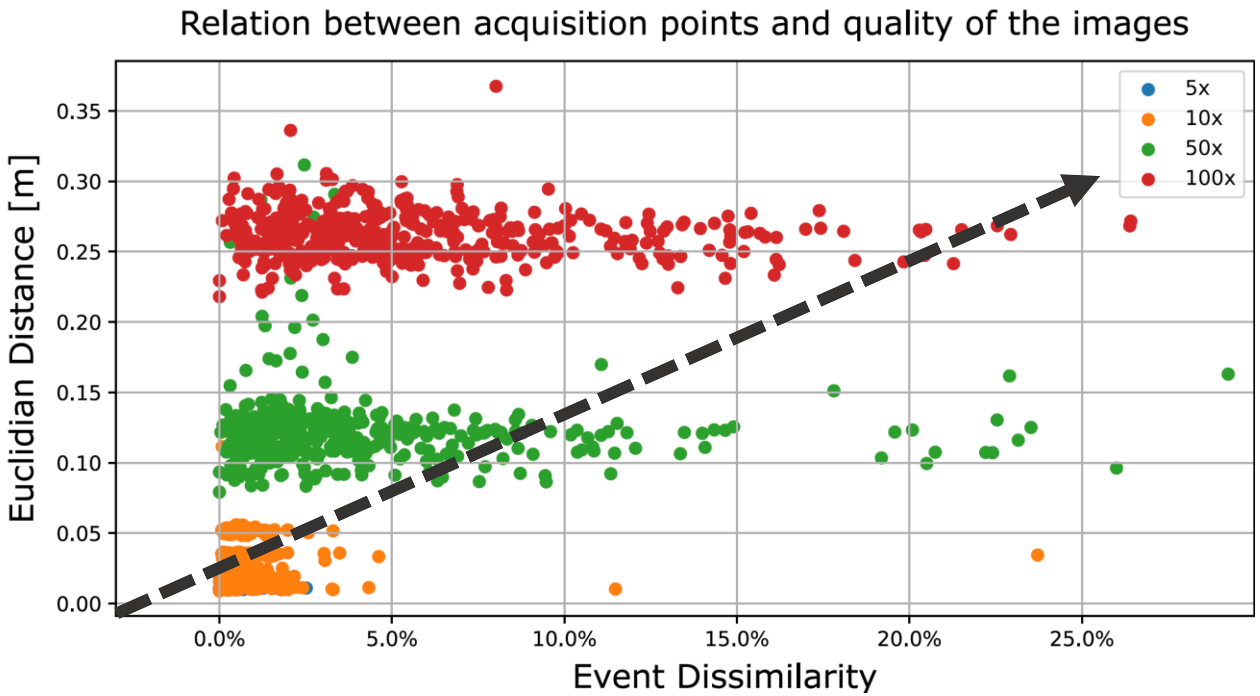
I - Trajectory accuracy

- For a given AirSim acceleration factor, the trajectory accuracy worsens with the command length ⇒ *the error accumulates between consecutive commands.* This is expected as the source of the trajectory error is the command latency.
- The proposed **TCSC** command significantly *outperforms* the **ADSC** trajectory accuracy.
  - The proposed **TCSC** provides a *distance higher than 1.8 in the median case.*
  - The **ADSC** reaches it for acceleration lower than 5x.





II - Visual Perception accuracy



Increasing the acceleration factor to *slightly increases the **ED** and the Euclidean distance*.

If the ***ED*** depended on the position error, the points would be *along the oblique arrow*.

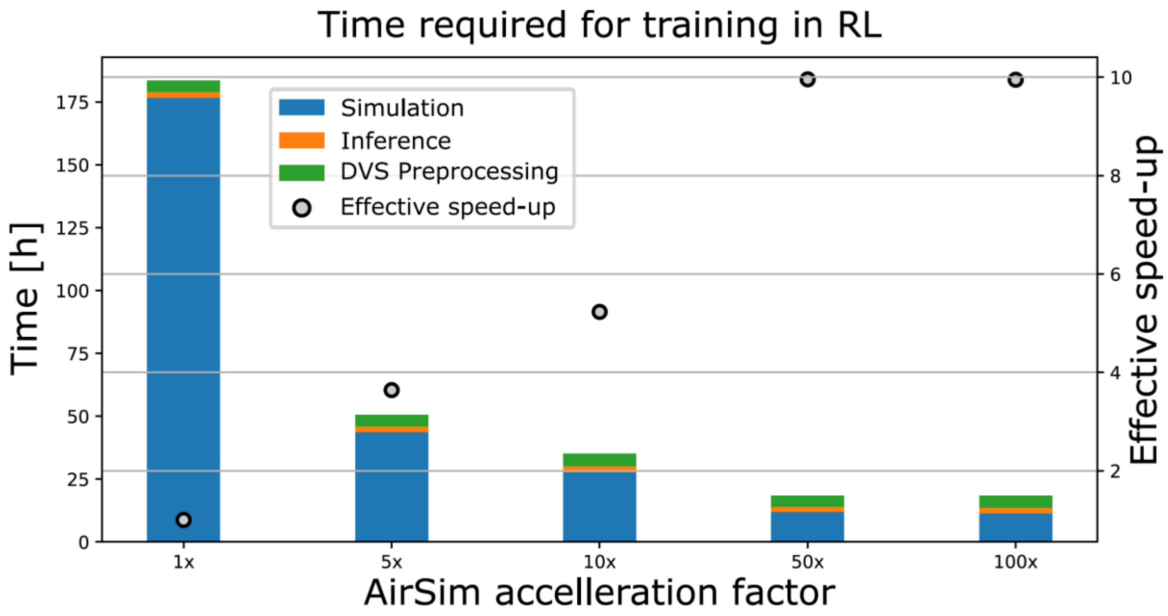
For each acceleration factor, *horizontal clusters are generated*.

*The difference in the images seen from the **UAV's** camera does not depend only on the position error induced by the simulation acceleration but that the latency in command can lead to a different camera orientation while the **UAV** is in a relatively similar position.*

III - RL Agent Performance

Simulation acceleration with **TCSC** applied to **RL** algorithms does not alter the simulation environment.

By increasing the acceleration of the simulation, the trajectories and perception perturbations increase, impacting the performance of the **RL**.



	ClockSpeed				
	1x	*5x*	*10x*	*50x*	*100x*
ADSC	**23**	**1**	**0**	**0**	**0**
TCSC	**21**	**30**	**23**	**14**	**13**

- By accelerating the simulation to 5.2x the **UAV** success ratio is the same as the inference with training performed in real-time
- Training with *ClockSpeed* of 50x and 100x achieved effective speed-up of 14.8x and 15.4x and replicated more than 62% of the **UAV** success ratio in real-time.

Accelerated simulation training with **ADSC** failed to complete games in real-time inference.

With **TCSC**, training that in real-time simulation requires one week of simulations has been replicated in less than two days.

Conclusions

A method to accelerate the training of **UAV** agents trained in **RL** by reducing the simulation time.

Time-Controlled Simulation Command (**TCSC**) in opposition to the AirSim Default Synchronous Command (**ADSC**)

Conclusions

A method to accelerate the training of **UAV** agents trained in **RL** by reducing the simulation time.

Time-Controlled Simulation Command (**TCSC**) in opposition to the AirSim Default Synchronous Command (**ADSC**)

1

Mitigate the error on the trajectories of UAV agents in accelerated simulation.

Conclusions

A method to accelerate the training of **UAV** agents trained in **RL** by reducing the simulation time.

Time-Controlled Simulation Command (**TCSC**) in opposition to the AirSim Default Synchronous Command (**ADSC**)

1

2

Mitigate the error on the trajectories of UAV agents in accelerated simulation.

Mitigate the noise due to the acceleration of the simulation that generates perturbations in the camera images.

Conclusions

A method to accelerate the training of **UAV** agents trained in **RL** by reducing the simulation time.

Time-Controlled Simulation Command (**TCSC**) in opposition to the AirSim Default Synchronous Command (**ADSC**)

1

Mitigate the error on the trajectories of UAV agents in accelerated simulation.

2

Mitigate the noise due to the acceleration of the simulation that generates perturbations in the camera images.

3

Perform accelerated training with **TCSC** and **ADCS**, comparing them in 100 games inferences on real-time simulations:

- Training with **TCSC** on accelerated simulation up to **5.2x** has the same UAV success ratio of the inference with training performed in real-time
- Training with a speed-up of **15.4x** replicated more than **62%** of the UAV success ratio in real-time.
- Accelerated simulation training with **ADSC** failed to complete games in real-time inference.

Thanks for your attention

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

IARIA