FT2025
SWEDISH AEROSPACE AT THE CROSSROADS
The 12th Swedish Aerospace Technology Congress
FT2025 in Stockholm
October 14-15, 2025

The Large Language Model Pilot: Generative Artificial Intelligence for Cyber-Physical Flight Simulations

Jorge Lovaco[1], Raghu Chaitanya Munjulury[1,2], and Petter Krus[1]

[1]*Department of Management and Engineering (IEI), Linköping University, Linköping, Sweden*
E-mail:, jorge.lovaco@liu.se
[2]*SAAB Aeronautics, Linköping, Sweden*

Abstract

A previous study investigated the application of Large Language Models (LLMs) in Agent-Based Simulations for Systems of Systems (SoS) studies, where the LLM acted as an Incident Commander during a wildfire event [1]. Firefighting Concepts of Operations (CONOPS) were utilised as an initial prompt to limit the LLM's answers. The findings were promising, indicating that LLMs may effectively drive simulation agents. Building on these insights, the goal was to integrate an LLM directly into a particular agent's decision-making loop. The present work investigated whether LLMs can be used in a cyber-physical simulation environment. The LLM was used with a cyber-physical simulation to fly a Unmanned Aerial Vehicle (UAV) and study its adherence to mission objectives. The LLM's judgments and the computational resources utilised throughout the simulation are discussed with special interest in future improvements and implications for designing autonomous aerial firefighting systems.

Keywords: Large Language Models, System of Systems, Concept of Operations, Cyber Physical Simulations

1 Introduction

Large Language Models represent a significant leap in generative Artificial Intelligence (AI), enabling the automated production of human-like text [2]. These models typically require extensive training on large datasets to develop sufficiently complex neural networks. As a result, early versions of LLMs demanded substantial computational power and were predominantly accessed from remote servers [2][3]. Recent advancements, however, have introduced optimised LLMs, which run on average consumer-grade hardware at the cost of reduced reasoning and generative capabilities [3]. The present work will investigate an LLM's performance in a cyber-physical simulation while piloting a UAV. The LLM will carry out a mission related to wildfire monitoring and suppression. The LLM's judgments and the computational resources utilised throughout the simulation will be analysed.
SoS studies provide the necessary understanding for evaluating collections of Constituent Systems interacting with each other. For example, in the context of a firefighting mission, these Constituent Systems could include firefighters, helicopters, satellites, etc. The relevance of performing a SoS analysis comes from finding emergent behaviours that arise from communication and collaboration [4].
These studies provide insight into the possible challenges during operations, such as coordination, interoperability, or group value [5]. When a new constituent system that should fulfil capabilities is to be designed, a holistic perspective is necessary to decide how it will be operated, in what scenarios, the interoperability with other systems, and which stakeholders will be involved. This holistic view will provide better requirements for the engineers, and it can only be achieved by studying the whole SoS. The drawback of SoS studies for generating new system concepts (aircraft concepts for the present work) is the potentially prohibitive cost [6]. For this reason, modelling and simulation are used for these studies [7].
Models are a representation of reality, having different levels of fidelity that are the source of different errors. They are cheaper than a real environment, and if used correctly, they can bring the necessary understanding for trade-off discussions. For these reasons, it is common to use Agent-Based tools for SoS analysis, as they are oriented towards the study of interactions and behaviours [8]. Previous research explored the use of LLMs in Agent-Based Simulations for SoS studies, where the LLM assumed the role of an Incident Commander during a wildfire situation [1]. Firefighting Concepts of Op-

erations (CONOPS) were used as part of the initial prompt to constrain the LLM answers. The results were promising, suggesting that LLMs could effectively drive simulation agents. Building upon these findings, the goal is to embed an LLM directly within a specific agent's decision-making loop.

2 Implementation

The study performed here is exploratory to support design research methodologies [9], which support design research for product development [10]. The current level of knowledge and understanding of LLMs interacting in dynamic simulation environments is low, as it has not been possible until recently. For the study, the methodology followed was:

- The software used for the simulation was Hopsan [11], implemented in the C++ language.
- The LLM model used was the 3-Gigabytes Gemma [3].
- The model was called through the Ollama interface [12].
- Regular expressions were used to update the prompts and parse the answers, converting from strings to numerical values.

The inputs taken by the LLM navigator component are shown in Figure 4 were:

- Battery State of Charge (SoC)
- Remaining Distance
- Distance to Runway
- Distance to Waypoint
- State

The outputs of the LLM navigator component were:

- Latitude
- Longitude
- Return

The inputs "State" and "Distance to Waypoint" were respectively used to enable the component and as a trigger for the prompting. The remaining inputs were used in the prompting with a list of coordinates by replacing the values at the appropriate place. To improve the simulation performance, the LLM calls are limited to every 10 simulated seconds instead of each simulation time step, which was 0.005 seconds. The given prompts were as follows:

Flight prompt: *You are piloting an Aircraft. The next waypoint in your list is [LATITUDE] latitude [LONGITUDE] longitude. Your current remaining battery capacity is [BATTERY STATE OF CHARGE]% (theoretical remaining flight distance is [REMAINING DISTANCE]m), and the distance to the runway is [DISTANCE TO RUNWAY]m. Please answer '1' if you want to fly back to the runway for landing, or '0' if you want to fly to the designated coordinates.*

Distress prompt: *You are piloting an Aircraft. There is a distress call from firefighters in need of aerial support at coordinates [LATITUDE] latitude and [LONGITUDE] longitude. Your current remaining battery capacity is [BATTERY STATE OF CHARGE]% (theoretical remaining flight distance is [REMAINING DISTANCE] m and the distance to the runway is [DISTANCE TO RUNWAY] m. Please answer just '1' if you want to fly back to the runway for landing, or just '0' if you want to fly to the designated coordinates to provide support.*

The mission to perform consists of a take-off phase, an airborne phase, and a landing phase. It is during the airborne phase that the LLM navigator is enabled to visit the coordinates that define the fictional area where a wildfire would be located. If the LLM decides to continue with the mission, the outputs are the latitude and longitude coordinates, and a value of 0 for the "Return" output. A value of 1 for the "Return" variable disables the LLM navigator component and enables the phase where the UAV returns to base for landing, becoming controlled all the way by a classical physics-based control system. The distress prompt is triggered after 2500 seconds in the simulation. This value could be randomised or parametrised for the user, but kept as a constant for consistent experiments.

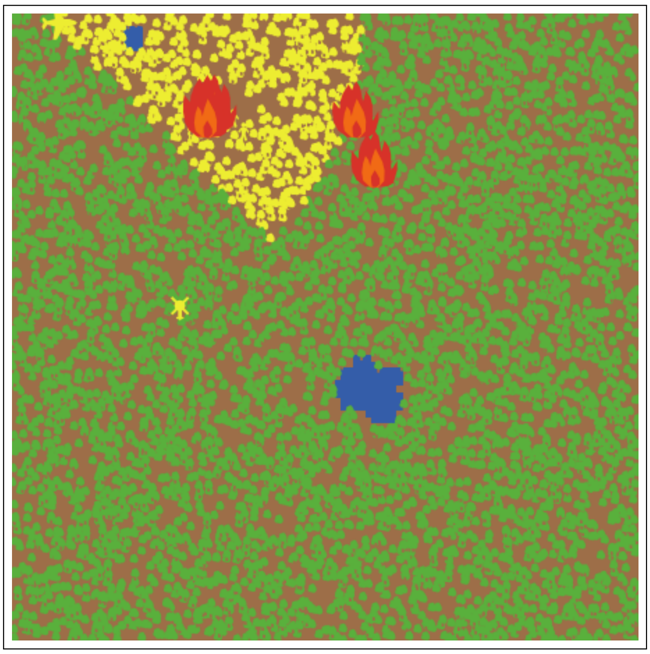

Figure 1: Agent-Based Simulation of a fire suppression case. (Adapted from [1] with permission.)

The navigation components in a cyber-physical aircraft model (Figure 2) were replaced with the LLM navigator agent. In the existing setup, mission waypoints and flight logic are connected to a control system that flies the aircraft (Figure 3). Here, the LLM acts as a simplified human pilot, evaluated in terms of communication and decision-making. Introducing such

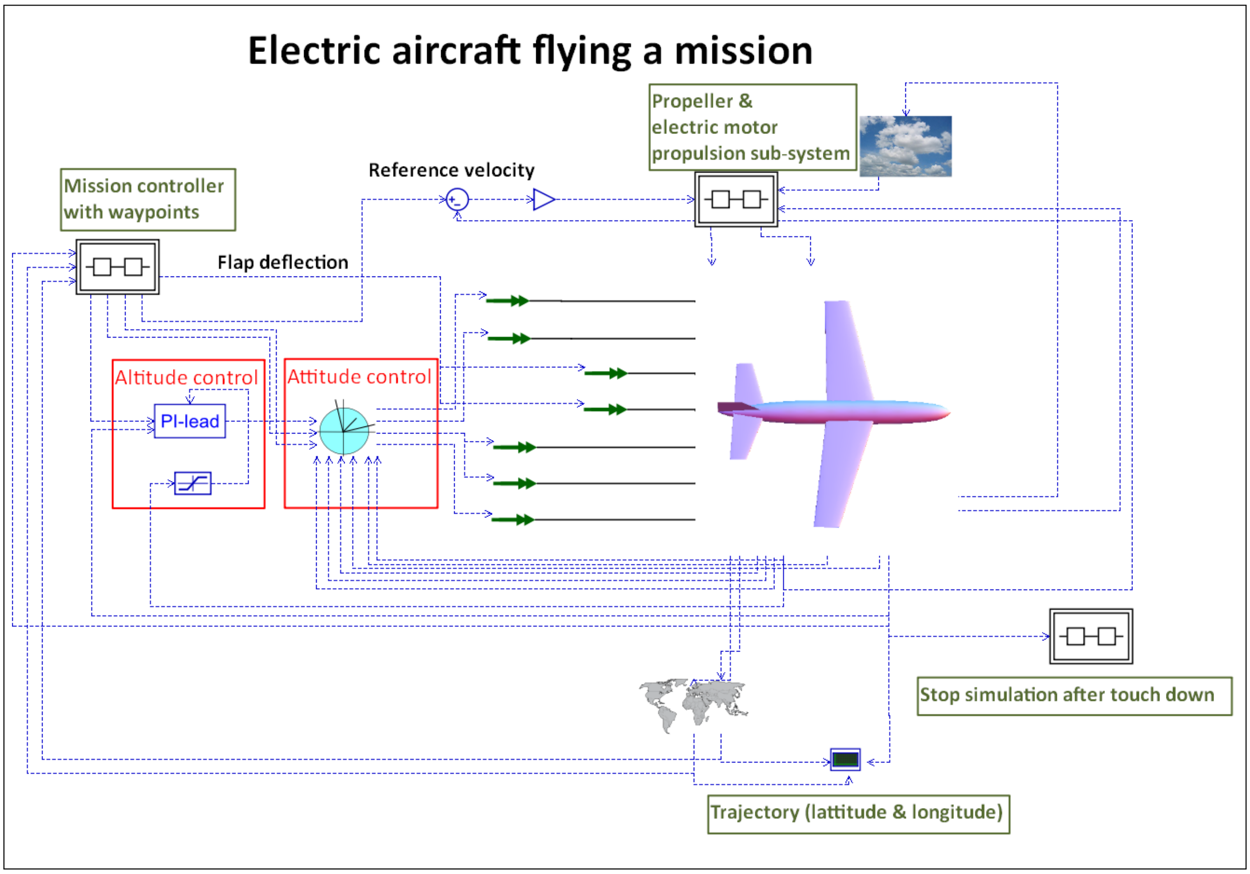

Figure 2: Electric Aircraft model in Hopsan.

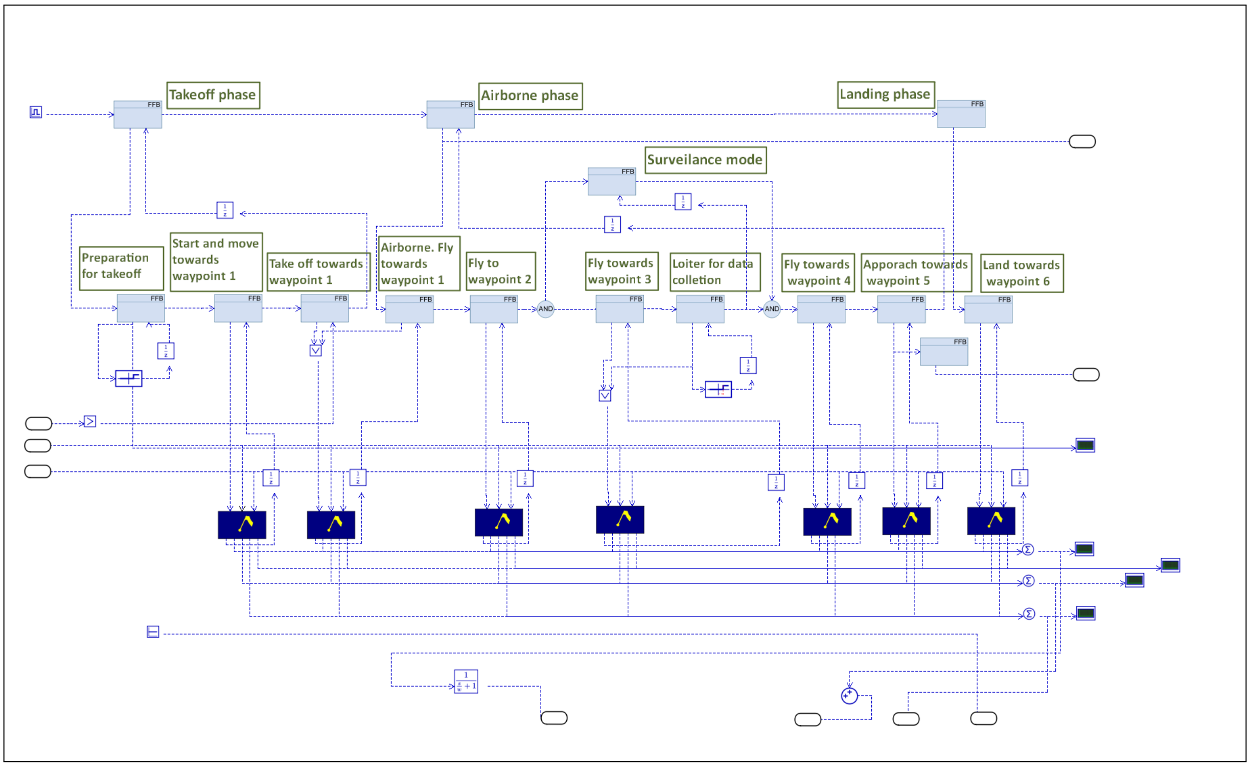

Figure 3: Flight logic diagram and waypoint components in Hopsan.

smart agents can support early trade-off discussions, helping with the definition of functional requirements and the establishment of operational guidelines for new aircraft concepts.

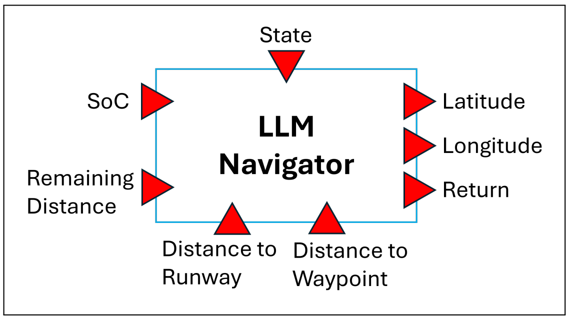

Figure 4: Large Language Model Navigator block.

3 Results

The scenario described in Section 2 was simulated both with and without a discretionary return choice for the LLM navigator. The simulations ran for approximately 20 minutes (1175 seconds) on a desktop computer, whereas the baseline model without the LLM component (Figure 2) completed the same in 12 seconds. For a simulated duration of 5000 seconds, the model still ran faster than real time. Figure 1 presents a simulation of a fire suppression scenario, with a UAV flying around the fire perimeter (the lighter, cone-shaped area indicating its visual range), which provided the trajectory used in the present study. Figures 5 and 6 show the latitude–longitude flight path, where the square-shaped trajectory is clear. The main difference in the flight path is the loop around the centre of the square, where the UAV performs a loiter task to provide aerial support to firefighters in distress. Figure 8 shows the

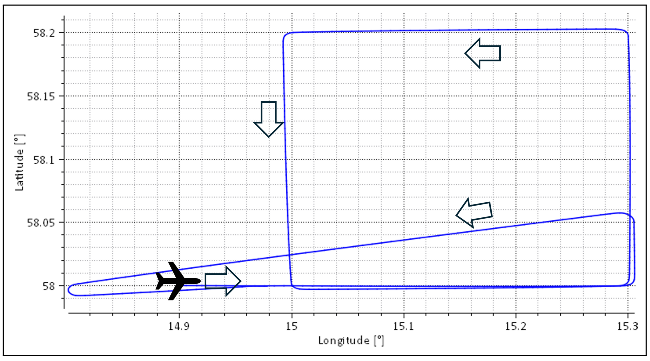

Figure 5: Path flown by the UAV without distress calls.

altitude during the mission (notice the variations during turns) and the changes during climbing or landing. Figure 7 shows the remaining SoC of the battery. Figure 9 shows the flight path when the LLM was allowed to choose between returning or continuing the flight at its own discretion. Figure 11 illustrates the altitude variation during climbs and turns. In this case, however, at the end of the simulation, the UAV remains in the airborne phase but is gradually losing altitude. As shown in Figure 10, the battery is nearly depleted and unable to provide the energy required to sustain flight.

Additionally, to test the image recognition capabilities of the LLM, Figure 1 was fed to it, providing the following output:
"Here's a description of what I see in the image:
Overall Scene:

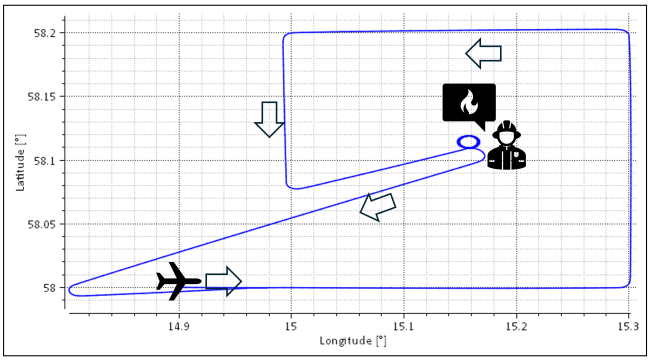

Figure 6: Path flown by the UAV with a distress call and forced return.

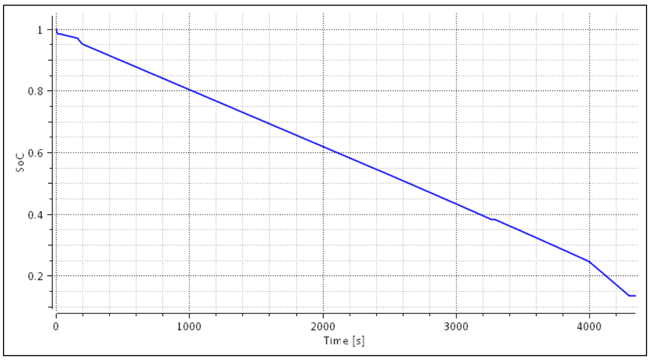

Figure 7: Evolution of the battery state of charge during flight.

The image depicts a pixelated, top-down view of a grassy landscape, reminiscent of an early computer game. The background is a detailed texture of green grass with brown patches, suggesting a field or meadow.
Elements:
Yellow "Fire": A cluster of yellow pixels is arranged in a triangular shape, clearly meant to represent flames or fire.
Blue Character: A single blue pixel character is positioned in the lower centre of the scene.
White Character: A small, white pixel character is located on the left side of the image.
Style & Impression:
The image has a retro, 8-bit aesthetic, likely from an older computer game. The limited colour palette and blocky shapes create a nostalgic and simplistic visual style."

4 Discussion

The results obtained demonstrate not only the feasibility of employing an LLM as a form of AI within a simulation but also its potential for operational analysis relevant to aircraft concept generation. The most relevant outcome was the LLM's decision to respond to a distress call by holding position and loitering above the personnel in distress, as shown in Figure 9. This decision highlighted the importance of equipping the system with the capability to land outside designated facilities if necessary. Functional requirements to achieve this could include, for example, a deployable parachute once the

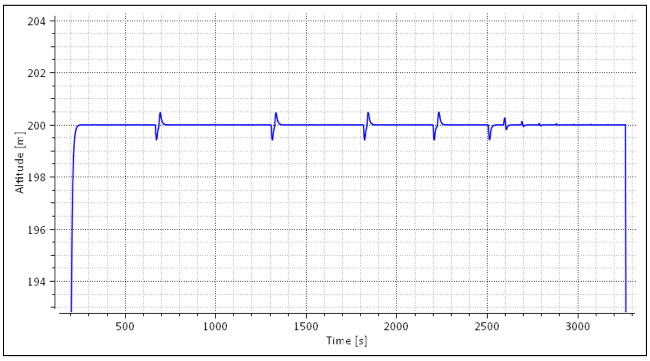

Figure 8: Evolution of the altitude during flight without discretionary return.

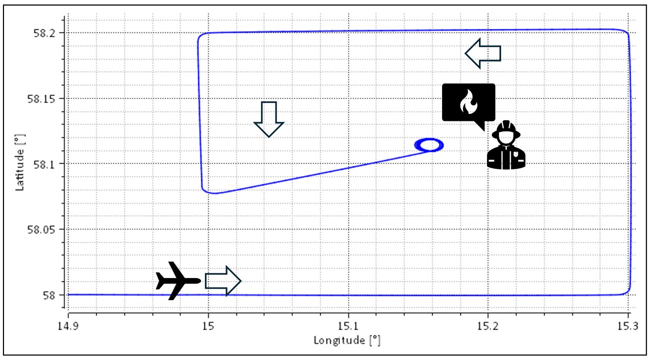

Figure 9: Path flown by the UAV with distress call and discretionary return.

battery is depleted (Figures 11 and 10), or vertical landing capability through a tilt-rotor or tilt-wing design. Alternatively, a more conventional option might involve a safety system that overrides the LLM navigation, such as an energy-monitoring mechanism (similar to the "Bingo Fuel" threshold in military aviation) that enforces the return to base.
The inputs provided to the LLM were latitude and longitude positions, the battery SoC, and the distances to the landing runway and the theoretical maximum flight range under current conditions. These were considered the minimum necessary to support reasonable decision-making, with the combination of SoC and theoretical distance regarded as redundant. The rationale was to assess whether the LLM would prioritise a specific battery SoC percentage, or whether such a threshold needed to be specified explicitly. Both Figures 7 and 10 indicate that, without a defined threshold in the prompt, the LLM did not consider it necessary to return, which is particularly concerning in scenarios without enforced return logic. Such behaviour is an important finding, as a UAV required to land outside designated areas could compromise the safety of firefighters already under distress. Therefore, when generating the CONOPS for the system and the SoS, this must be taken into consideration.
The prompts used in Section 2 were highly constrained, which limited the evaluation. Since the simulation software is restricted to numerical inputs and outputs, the responses had to conform to a fixed structure for repeatability. A broader assessment would require more advanced methods, such as

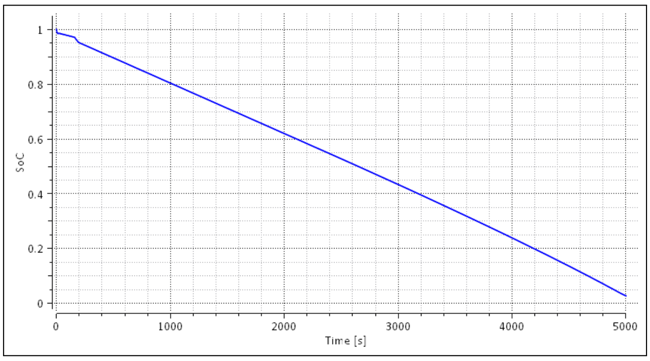

Figure 10: Evolution of the battery state of charge during flight with discretionary return.

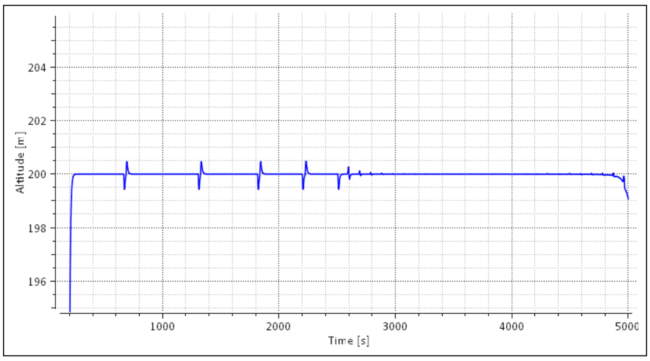

Figure 11: Evolution of the altitude during flight with discretionary return.

another LLM acting as a filter to extract the relevant outputs from unconstrained responses.
The computational performance is relevant for generating data for the analysis of different situations. The simulation speed was faster than a real-time operation, which means it has value for research, especially if parallelisation is used. However, only one LLM component was used; thus, a simulation with multiple LLM agents is still to be studied. Having several agents with an AI based on an LLM could provide insightful results for studying behaviours and operations, but if the computational costs are too demanding, other forms of AI, such as machine learning or reinforcement learning, should still be considered.
Finally, the description of the simulation plot (Figure 1) given by the LLM was compelling. With no prior prompt or context, it recognised the fire symbols, identified human figures, and correctly characterised the pixelated visual style, even noting the blue area corresponding to a lake in the original model. This performance suggests potential for experimenting with aerial imagery of wildfires. If sufficiently reliable, this opens the possibility of employing UAVs equipped with LLMs for image recognition, capable of reporting wildfire detections in natural language and providing descriptive situational information.

5 Conclusion

The present work employed an LLM to make decisions during a simulated mission involving a UAV. As a form of AI, the LLM interpreted the provided inputs and determined whether to continue the mission or return to base.
Notably, when human lives were at stake, the LLM chose to continue providing aerial support, even as the battery was close to depletion. Albeit an early evaluation, the outcome is considered both successful and promising, particularly given the use of a low-performance LLM. These findings suggest future potential for Agent-Based Modelling and Simulation, where LLMs could be integrated into new types of agents, replacing rigid, code-based rules. This integration would provide greater flexibility and adaptability in simulated agents for more extensive studies of SoS operations.
Challenges that remain and are difficult to overcome are the variability of the answers from the LLM, even when identical prompts are used. However, another future possibility is to examine the suitability of a general purpose LLM against a fine-tuned LLM. For agent behaviour research, it would be interesting to find the issues or benefits related to using a general purpose or a fine-tuned LLM, or even some specific combination of them, that allows to keep the freedom and also enhances repeatability.

ACKNOWLEDGEMENTS

The research presented in this paper has been performed in the framework of the COLOSSUS project (Collaborative System of Systems Exploration of Aviation Products, Services and Business Models) and has received funding from the European Union Horizon Europe program under grant agreement No. 101097120. The Swiss participation in the Colossus project is supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00609. The authors would also like to acknowledge the Swedish Innovation Agency (VINNOVA) and The Swedish Defence Material Administration (FMV) for financial support through the grant 2019-05371.

References

[1] Jorge Lovaco, Raghu Chaitanya Munjulury, Ingo Staack, and Petter Krus. Large language model-driven simulations for system of systems analysis in firefighting aircraft conceptual design. In *34th Congress of the International Council of the Aeronautical Sciences*, 2024.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Google-DeepMind. Introducing gemma 3: The most capable model you can run on a single gpu or tpu, 20254. Accessed: March 13, 2025.

[4] Mark W Maier. Architecting principles for systems-of-systems. *INCOSE International Symposium*, 6(1):565–573, 1996.

[5] Dominique Luzeaux and Jean-Rene Ruault. *Systems of systems*. Wiley Online Library, 2010.

[6] Kilian J. Murphy, Simone Ciuti, and Adam Kane. An introduction to agent-based models as an accessible surrogate to field-based research and teaching. *Ecology and evolution*, 10:12482–12498, 2020.

[7] Michael Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.

[8] Daniel A DeLaurentis, Kushal Moolchandani, and Cesare Guariniello. *System of systems modeling and analysis*. CRC Press, 2022.

[9] Kristina Säfsten and Maria Gustavsson. *Research methodology: for engineers and other problem-solvers*. Studentlitteratur AB, 2020.

[10] Lucienne TM Blessing and Amaresh Chakrabarti. *DRM: A design reseach methodology*. Springer, 2009.

[11] Robert Braun, Peter Nordin, Liselott Ericson, L Viktor Larsson, Petter Krus, and Maria Pettersson. Hopsan: an open-source tool for rapid modelling and simulation of fluid and mechatronic systems. In *Fluid Power Systems Technology*, volume 83754, page V001T01A047. American Society of Mechanical Engineers, 2020.

[12] Francisco S Marcondes, Adelino Gala, Renata Magalhães, Fernando Perez de Britto, Dalila Durães, and Paulo Novais. Using ollama. In *Natural Language Analytics with Generative Large-Language Models: A Practical Approach with Ollama and Open-Source LLMs*, pages 23–35. Springer, 2025.