Propellant sloshing effect modelling of spacecraft with Machine Learning

Oscar Ortiz Casanova[(1)], Oihana Coustié[(2)], Jules-Edouard Denis[(3)], Xavier Manuel Juanpere[(4)], Claire Thevenot[(5),], Hélène Evain[(6),], Denis Standarovski[(7)]

[(1)] Airbus Defence and Space, +33 698209437, oscar.ortiz-casanova@airbus.com
[(2)]Airbus Defence and Space, +33 562198475, oihana.coustie@airbus.com
[(3)]Airbus Defence and Space, +33 582521240, jules-edouard.denis@airbus.com
[(4)]Airbus Defence and Space, +33 562198811, xavier.manuel-juanpere@airbus.com
[(5)]Airbus Defence and Space, +33 562198939, claire.thevenot@airbus.com
[(6)]CNES, +33 561274070, helene.evain@cnes.fr
[(7)]CNES, +33 561273875, denis.standarovski@cnes.fr

Mailing address for Airbus Defence and Space:
31 Rue des Cosmonautes, 31400 Toulouse

Mailing address for CNES:
18 Avenue Edouard Belin, 31400 Toulouse

ABSTRACT

The sloshing effect of liquids occurring within spacecraft's fuel tanks is a determinant factor in the pointing performance of space missions with high-reactivity needs. Accurately modelling this phenomenon allows anticipating its detrimental effects on satellite control, and thus possibly improve the tranquillisation time after maneuvers.

This paper examines a modelling trade-off between the quick but low transient representativeness analytical approaches and the slow but accurate Computational Fluid Dynamics (CFD) simulations. Based on the hypothesis that the problem presents underlying regularity characteristics, and considering that dataset can be generated thanks to CFD simulations, Machine Learning techniques are explored, from a Proof of Concept approach to a fully representative agile space mission.

Using Machine Learning techniques such as Multi-Layer Perceptron (MLP), Long-Short Term Memory (LSTM) or Convolutional Neural Network (CNN), this study demonstrates high-accuracy results whilst remaining efficient in terms of computation time. Furthermore, its industrialization capability for future missions is assessed, providing promising results. For instance, it is shown that a relatively small dataset is needed to reach high representativeness and robustness. Besides, a methodology is developed to implement these models into a Guidance, Navigation and Control (GNC) closed loop simulator based on Matlab/Simulink.

1 INTRODUCTION

1.1 Context and missions

The central theme of the study is in the context of maneuverable space missions with a need for responsiveness and/or agility.

The angular acceleration profiles resulting from fast maneuvers stress the dynamics and amplify errors and non-linearities. The residual dynamics at the end of the maneuver lead to pointing transients that degrade line-of-sight stability, delaying the beginning of the next imaging opportunity. To avoid degrading overall agility and thus maximize system efficiency, it is necessary to minimize these settling times. This becomes even more important in the context of new acquisition principles, where each imaging opportunity requires a large number of small maneuvers. Since most of the error results from a mismatch between the actual dynamics and the model used to establish the guidance profile, it is important to improve the accuracy of this model.

Another related application area concerns the so-called "gyroless" architectures, which are becoming more and more common, where attitude determination requires filtering of noisy stellar measurements. In the absence of an inertial reference system, this filtering must rely on propagating a model of the dynamics rather than the gyroscopic measurements. Such dynamic filters are implemented onboard in the attitude control loop as well as on the ground for post-processing geolocation and geometric correction of images.

1.2 Study objective and potential applications

The objective of the study is to demonstrate that the technology is mature enough to establish, through Machine Learning techniques, a substitution model to represent the effects of propellant sloshing during manoeuvres. A model established by this approach will be specific to a given agile mission, with fixed tank geometry and fluid characteristics, and a reduced parameter space for amplitude, duration, shape, and direction of manoeuvres. The aim is to prove that the precision of this model will improve the propagation error with respect to simplified models.

The resulting model will be computationally efficient, making it suitable for deployment in the design and validation process, as well as in operations. Such a model will be generic enough to be useful at various stages of the mission.

During the design phase:

- For estimating closed-loop performance (settling time).
- For attitude restitution (image geolocation, deconvolution).
- For functional validation.

During the mission itself:

- Open-loop propagation in a ground filter for attitude restitution
- Closed-loop operation in an embedded dynamic filter.

1.3 Reminder on the interest of AI based methods for modelling the sloshing phenomenon

Currently, there is no macroscopic mathematical model to represent sloshing phenomena during attitude manoeuvres, except for conservative approaches suitable for design stages [1]. Such a model should be able to calculate the reaction torques (and forces) transmitted by the fluid to the satellite via the tank wall, depending on the current angular acceleration of the satellite and the current state of the fluid (or the history of the angular acceleration). However, due to the highly non-linearity and coupled nature of these phenomena, a macroscopic mathematical model probably

does not exist, except for some simple regimes (e.g., capillary regime at very low accelerations or full tank regime).

Recent advances in understanding the sloshing of propellants in microgravity have made it possible to simulate these phenomena accurately using numerical calculation codes [2]. The level of confidence in these CFD approaches have been significantly consolidated, particularly thanks to the FLUIDICS experiment on the International Space Station [3], as experimental data acquired in microgravity conditions made it possible to calibrate the numerical models precisely. Unfortunately, these numerical models are still too computationally expensive to be implemented in real-time, either on-ground or on-board.

We thus find ourselves in a situation where the application of Machine Learning techniques is naturally necessary:

- The solution to the problem does not have a simple structure that allows it to be summarized by an analytical formulation or even an ad-hoc heuristic.
- We can assume that the problem has underlying regularity characteristics, which could be extracted from the training data.
- We can generate training data using CFD calculation. We even have real experimental data available to potentially verify the validity of the training dataset on a few points.

2 REFERENCE CASE

During this study, we analyzed two reference cases. Firstly, we followed a proof-of-concept approach with a simplified tank and manoeuvre profile. Then, we considered a tank and manoeuvres representative of an agile mission. While we will mainly focus on the second case in this paper, we will also make some comments on the proof-of-concept analysis.

2.1 Proof-of-concept reference case (PoC Case)

The proposed tank for the study is a spherical tank with a diameter of 10cm. This geometry is simple and presents many symmetries that can be exploited during learning. The computational domain is limited due to the small size of the tank and the relaxed constraint on mesh quality. Indeed, the comparison between experimental data from the ISS experiment with this tank (FLUIDICS) and open-loop simulations with Flow3D allowed us to optimize the physical models, numerical methods, and determine the mesh fineness.

To limit the scope of the study, we propose to generate reference cases with a single filling ratio of 50% (in the range where sloshing phenomena is the most significant).

Therefore, this model has a triple advantage:

- It does not require any setup effort.
- It has already been validated with real microgravity data [2].
- Its mesh has been optimized to have a low calculation time compared to automatic meshing.

In terms of maneuvers, they have been simplified as a succession of bang-coast-bang profiles in acceleration with an initial and final velocity condition to be achieved in a determined time, interspersed with constant velocity profiles (which correspond to the shooting). Additionally, it is important to note that these maneuvers only affect a single axis.

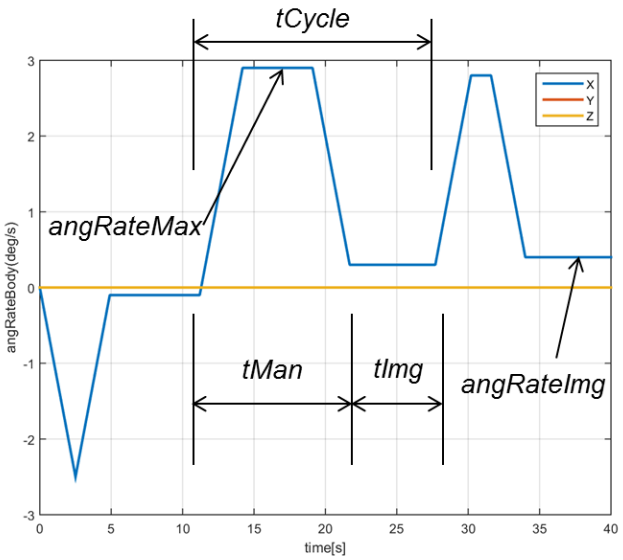

Figure 1: Example of manoeuvre.

During a mission, layout constraints do not allow the tanks to be placed exactly at the satellite center of mass. To represent this phenomenon, it was decided to place the tank on the Z-axis so that it is also subjected to the liquid forces, not just the torques. It was decided that the tank will be placed at a distance of -200 mm on the Z axis.

2.2 Mission representative reference case (MR Case)

In comparison to the previous tank, this one is significantly larger. It is also a spherical tank, but with a diameter representative of an Earth observation agile mission, measuring 1.16m (817L). We have considered three filling ratios, namely 20%, 50%, and 80%.

When it comes to the maneuvers, we aim to have a liquid initial state that differs from the one provided by the CFD simulation initialization. To achieve this, we need to conduct at least two maneuvers: the first one to disturb the liquid state, and the second one to start with the disturbed initial state. This approach will help us cover the effects of "memory".

On the other hand, we want to maximize the number of different cases while keeping the simulations as short as possible. Therefore, we plan to use a sequence of two maneuvers, as illustrated in Figure 2. This satisfies the requirement of having at least two maneuvers and short CFD simulations. The study will focus on the model's ability to be representative over a sequence of maneuvers.

For this study, the tank is positioned at a fixed distance of -446mm from the satellite center of mass.

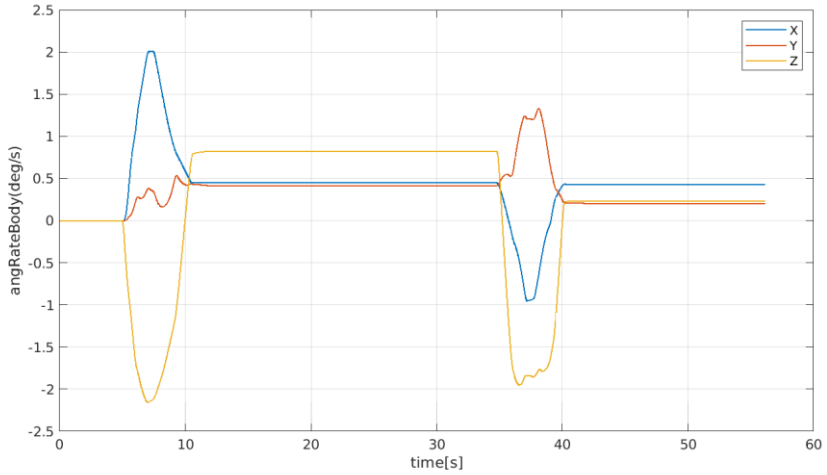

Figure 2: Example of manoeuvre.

It is worth noting that this time the maneuvers will be about all three axes.

3 DATA GENERATION

First, data for the maneuvers is generated to feed into CFD simulations. Then, this data, along with the outputs from the CFD simulations, is used to train the Machine Learning model.

3.1 Maneuvers Design of Experiment (DoE)

The experimental plan should offer the best parametric coverage while minimizing the number of simulations to be performed.

In the table below, we can find all the parameters scanned for maneuvers data generation regarding the MR Case. Furthermore, the domain associated with each parameter is defined so that it is representative of an agile mission. As a reminder, the number of chained maneuvers is fixed and equal to two.

Table 1: Maneuvers Design of Experiment.

Parameter	Domain
X velocity at the end of 1st maneuver	[0.1, 1] deg/s
Y velocity at the end of 1st maneuver	[0.1, 1] deg/s
Z velocity at the end of 1st maneuver	[0.1, 1] deg/s
X velocity at the end of 2nd maneuver	[0.1, 1] deg/s
Y velocity at the end of 2nd maneuver	[0.1, 1] deg/s
Z velocity at the end of 2nd maneuver	[0.1, 1] deg/s
Imaging time at 1st maneuver	[2, 30] s
Imaging time at 2nd maneuver	[2, 30] s
Angle swept during 1st maneuver	[1, 30] deg
Angle swept during 2nd maneuver	[1, 30] deg
Rotation direction of 1st maneuver: X component	[-1, 1]
Rotation direction of 1st maneuver: Y component	[-1, 1]
Rotation direction of 1st maneuver: Z component	[-1, 1]
Rotation direction of 2nd maneuver: X component	[-1, 1]
Rotation direction of 2nd maneuver: Y component	[-1, 1]
Rotation direction of 2nd maneuver: Z component	[-1, 1]
Tank filling ratio	{20, 50, 80} %

Sweeping method

The number of simulated maneuvers is limited due to the computational cost of CFD simulations. 600 simulations were considered a good compromise between the CFD and AI teams. The strategy chosen to define the DoE, and thus to sweep all the parameters defined in Table 1, is Latin Hypercube sampling (see Figure 3). This sampling follows a simple rule: two sampling points in the variable space never share the same value for a given variable. Additionally, a strategy is proposed to maximize space filling (see Figure 4).

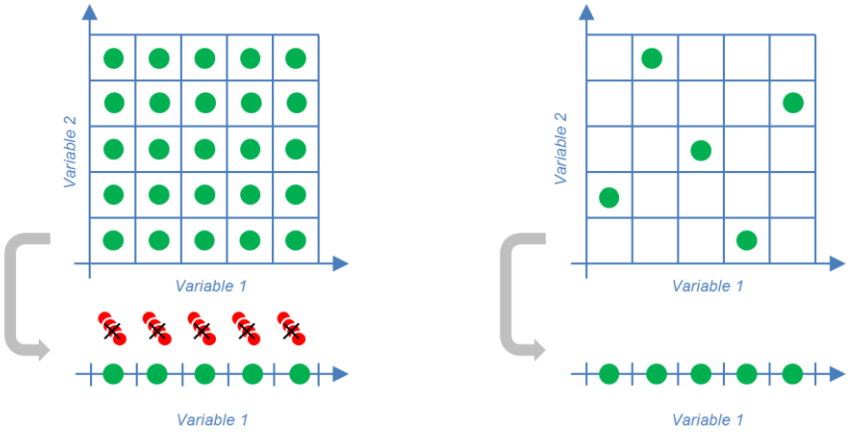

Figure 3: Latin Hypercube sampling strategy.

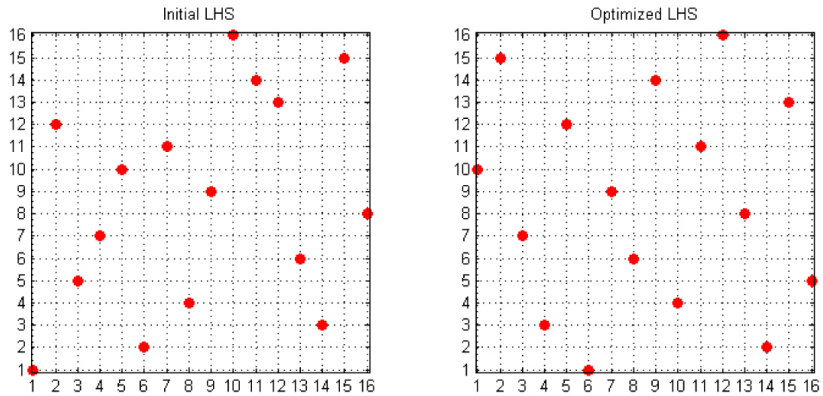

Figure 4: Strategy to maximize space filling.

3.2 Computational Fluid Dynamics Simulations

The maneuvers generated as described previously were implemented and executed using the CFD software FLOW3D. This software was validated by comparison with FLUIDICS results and was used during the PoC study. In Figure 5, screenshots of the gas bubble within the tank for different filling ratios are illustrated. In Figure 6, an example of forces and torques obtained by CFD simulation is illustrated. This type of data will be used for training the Machine Learning model.

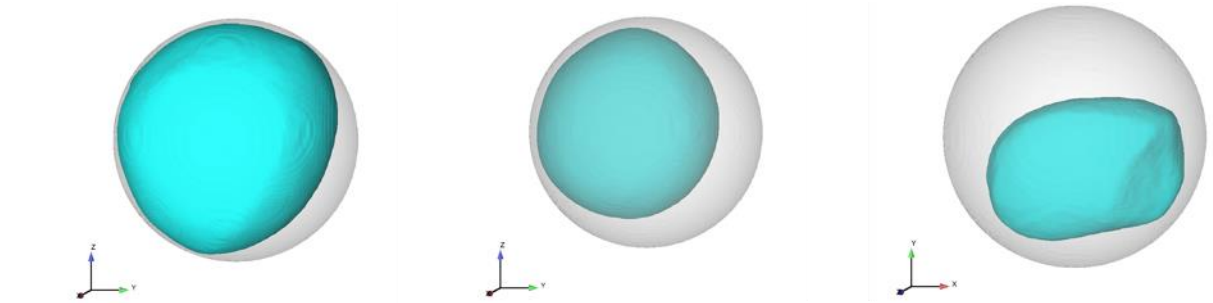

Figure 5: Position and deformation of gas bubble for filling ratio of 80% (left), 50% (center) and 20% (right).

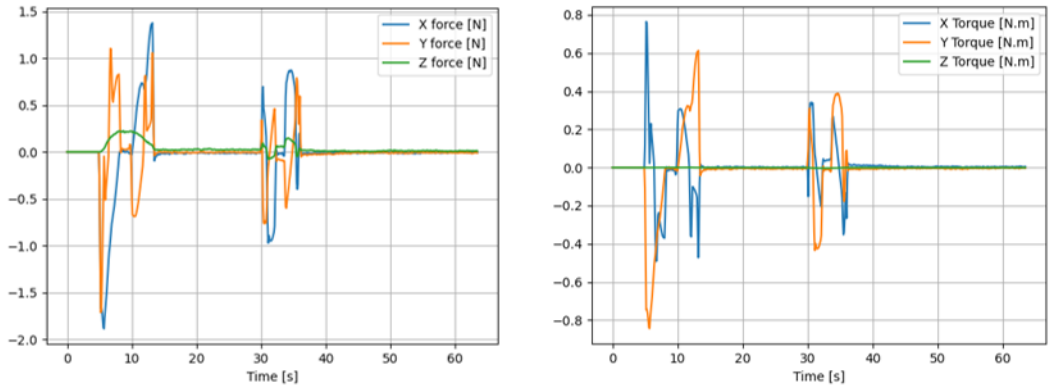

Figure 6: Examples of forces and torques obtained by a CFD simulation.

4 SOLUTION ARCHITECTURE

The objective is to predict the fluid forces and moments inside the tank based on the input of satellite angular velocity and acceleration. The initial analysis of the data, where a linear relation between some inputs and outputs was found, led us to construct a hybrid architecture for the prediction, one part being linear, and another non-linear used for the residual component. Here, we present some specificities that guided our choice of architecture.

Firstly, the input variables are exogenous, meaning they differ from the variables to be predicted. In contrast, the variables used in self-prediction cases (for future time steps) are called endogenous. We particularly want to exploit the linear relation between the output data and the exogenous input data.

Furthermore, we observed that the prediction quality by a linear model was deteriorated between the first and the second maneuver. Assuming this degradation is due to the fact that the fluid does not return to its initial state between maneuvers, we want to study methods that can abstract information about the previous state. For this purpose, we will study Deep Learning methods that include hidden states.

Finally, the architecture should respect, as much as possible, the business constraints on data availability. In this case, it is difficult to access prior knowledge of fluid forces and torques during flight. While this data can be used during the training phase, real-time prediction should not rely on its availability. Therefore, we will strive to minimize our dependence on output data for the prediction phase.

The generic solution selected is presented in Figure 7. It should be noted that at this stage, only the

data flow and method types are defined: any neural network can be implemented, as long as it respects the specified input and output data. Our method operates in a distinguished way between the learning phase (blue), where the output data is available, and the prediction phase (orange), where it is not. During the training phase, the following are trained:

- The inverse linear regression, which predicts angular accelerations based on forces and torques.
- A neural network that predicts the residuals of the linear regression based on angular accelerations.

The prediction phase follows the schema in reverse order. The angular accelerations are provided to the neural network, which predicts the residuals. These residuals allow the reconstruction of the angular accelerations, as they would have been altered by the linear regression. Finally, these reconstructed accelerations are transformed by linear regression (orange) to obtain the forces and torques.

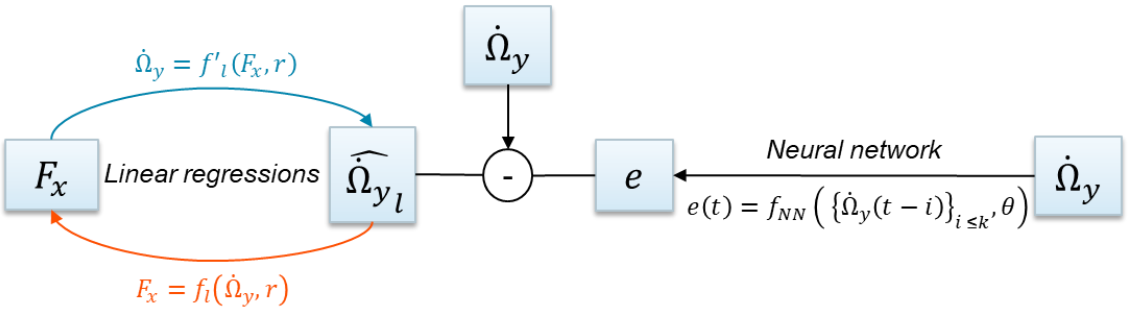

Figure 7: Architecture selected for this study. In blue, path used during the training phase. In orange, path used during the prediction phase.

Comment on the PoC Case

Within the PoC Case study, this linear relation between inputs and outputs was not observed. Therefore, the hybrid approach presented could not be used, and prediction was done completely with neural networks. This fact shows that the Machine Learning solution shall be adapted to the current use case.

5 MODEL EVALUATION

5.1 Evaluation frame

The data is divided into three subsets:

- The training data is used to train the models.
- The validation data is used to ensure the genericity of the learned model.
- The test set is the last set on which no learning is performed. The final evaluation is generally carried out on this set.

The following distribution is proposed:

Table 2: Data distribution.

	Training	**Validation**	**Test**
Simulations number	360	180	60
Ratio	60%	30%	10%

This distribution ensures significance in both learning and validation. Since the remaining data for

testing is very small, we will perform our evaluation on the set formed by the test and validation data.

The model is compared to several reference methods. These methods are:

- A linear model (LM) that predicts the output with a single linear regression.
- The non-hybrid Multi-Layer Perceptron (MLP) model implemented during the PoC Case, with a lag of 30, i.e. the prediction is made based on the 30 previous time steps.
- A hybrid implementation of this same MLP model (h-MLP).

Finally, the metric Mean Squared Error (MSE) normalized is used for the performance evaluation.

5.2 Results on initial data

The prediction score obtained for each model is illustrated in the table below.

Table 3: Prediction score on the validation and test data. LSTM stands for Long Short-Term Memory. GRU stands for Gated Recurrent Unit. CNN stands for Convolutional Neural Network.

Model	***Fx***	***Fy***	***Fz***	***Tx***	***Ty***	***Tz***	**Average**
MLP	0,3429	0,3901	0,6678	0,3846	0,3455	0,1727	0,3648
LM	0,0193	0,0248	0,2992	0,0250	0,0194	0,2162	0,0260
h-MLP	0,0191	0,0250	0,4705	0,0250	0,0191	0,2224	0,0265
h-LSTM	0,0182	0,0239	0,3900	0,0239	0,0182	0,1934	0,0248
h-GRU	0,0190	0,0243	0,4559	0,0243	0,0190	0,2042	0,0258
h-CNN	0,0175	0,0233	0,4084	0,0233	0,0174	0,2183	0,0246
Mass-spring	0.3286	0.3580	0.0036	N/A	N/A	N/A	N/A

Firstly, it appears that adding the linear aspect has a major impact on the score: the error is reduced by a factor greater than 30. The three hidden state models present similar results, with a slight advantage for hybrid Convolutional Neural Network (h-CNN). These latter models slightly improve the hybrid model without hidden state, h-MLP, which itself offers an improvement over linear prediction.

The improvements offered by the hybrid models are therefore significant, except for the variable *Fz*, whose prediction is of poor quality (worse than the linear model), regardless of the model. A probable explanation would be related to the low reliability of linear regression: the linear relation seems to be less verified on this data than on the other learned variables. The residuals are therefore more diffuse, and the neural network fails to model their behavior.

Figure 8 shows the results for all variables of the h-GRU model compared to real data and linear prediction. The previous observations apply to the prediction of all variables, except for *Fz*, whose prediction is coarse and slightly worse than that of the linear model.

Figure 9 shows that, while the predictions of the h-CNN model and the original data are almost superimposed on the entire dataset, the mass-spring model exhibits some inaccuracies: amplitude errors during maneuvers, and incorrect oscillations before and after maneuvers. Besides, Figure 10 shows that the mass-spring model does not capture the high-frequency content of the CFD signal.

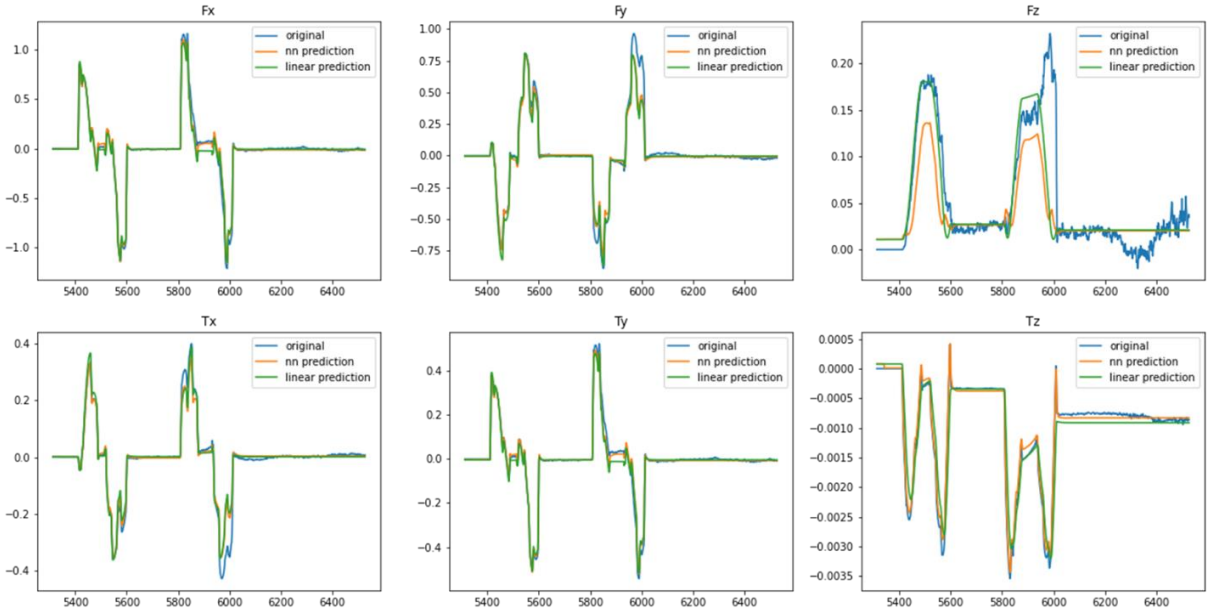

Figure 8: Prediction with h-GRU model, compared to linear model and CFD data.

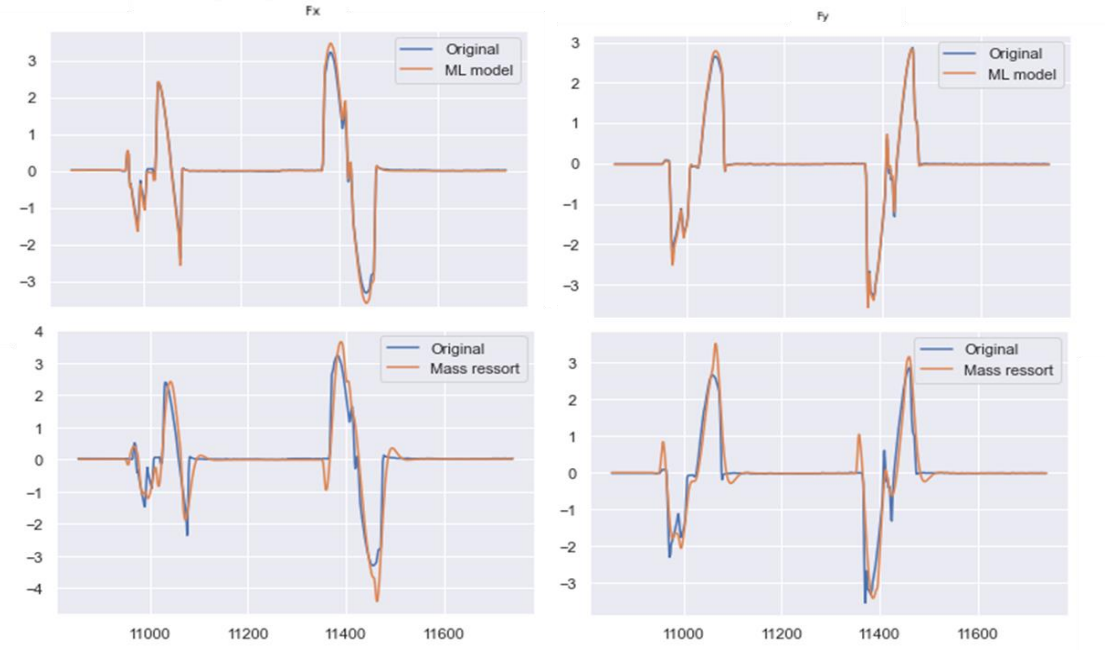

Figure 9: Prediction comparison between h-CNN and mass-spring models.

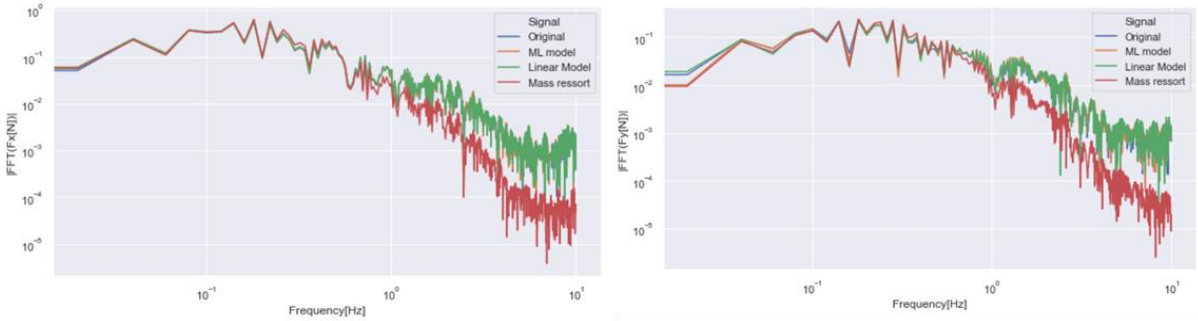

Figure 10: Frequency content of forces signals for different models.

The table below presents the training time for each method:

Table 4: Performance of each model.

	MLP	h-MLP	h-LSTM	h-GRU	h-CNN
Number of parameters	1876	2964	51460	50628	38308
Epoch number	25	26	17	22	19
Training time (s)	1052	1118	712	768	**665**

Comment on PoC case

Since for the PoC case this linear relation between the variables did not exist, the predictions score is significantly worse. In Figure 11, a prediction with the best model for the PoC case (MLP with lag) is shown. However, it still remains better than the mass-spring model.

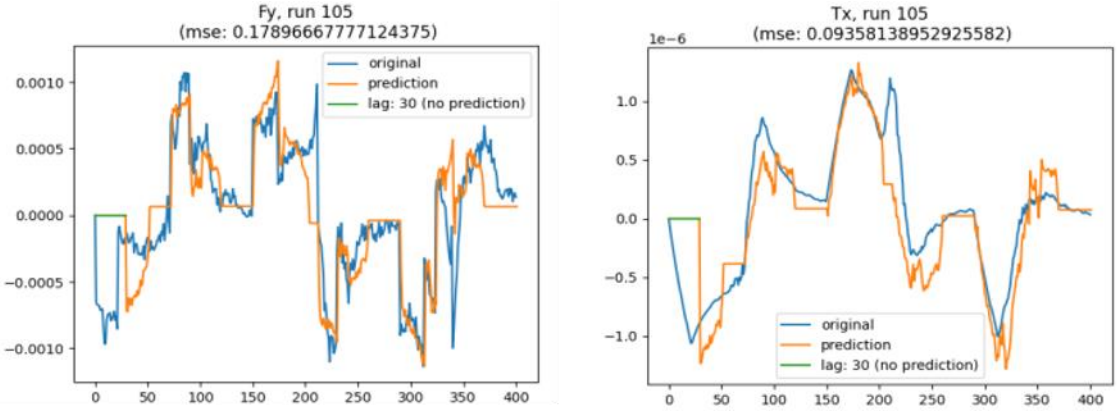

Figure 11: Prediction for PoC case with MLP model with lag.

5.3 Results on additional data – robustness analysis

In this section, we propose to delve deeper into the evaluation of our models by assessing their robustness. We aim to evaluate the models' ability to generalize the learned phenomena. To achieve this, we will first analyse the models' sensitivity to the size of the training sample. Additionally, we suggest generating new data to investigate whether the models trained on the 600 initial simulations can be applied to other data. Specifically, we will generate simulation data with different tank filling ratios (multi-ratio data) and with more consecutive maneuvers (multi-maneuver data).

For both cases, the prediction score (MSE) remains very similar. An example of prediction for multi-ratio data is shown in Figure 12, and for multi-maneuver in Figure 13. Furthermore, Figure 14 shows an oscillation of the MSE depending on the maneuvers, and not a monotonic increase of the error measurement. Therefore, the degradation observed previously appears to be bounded.

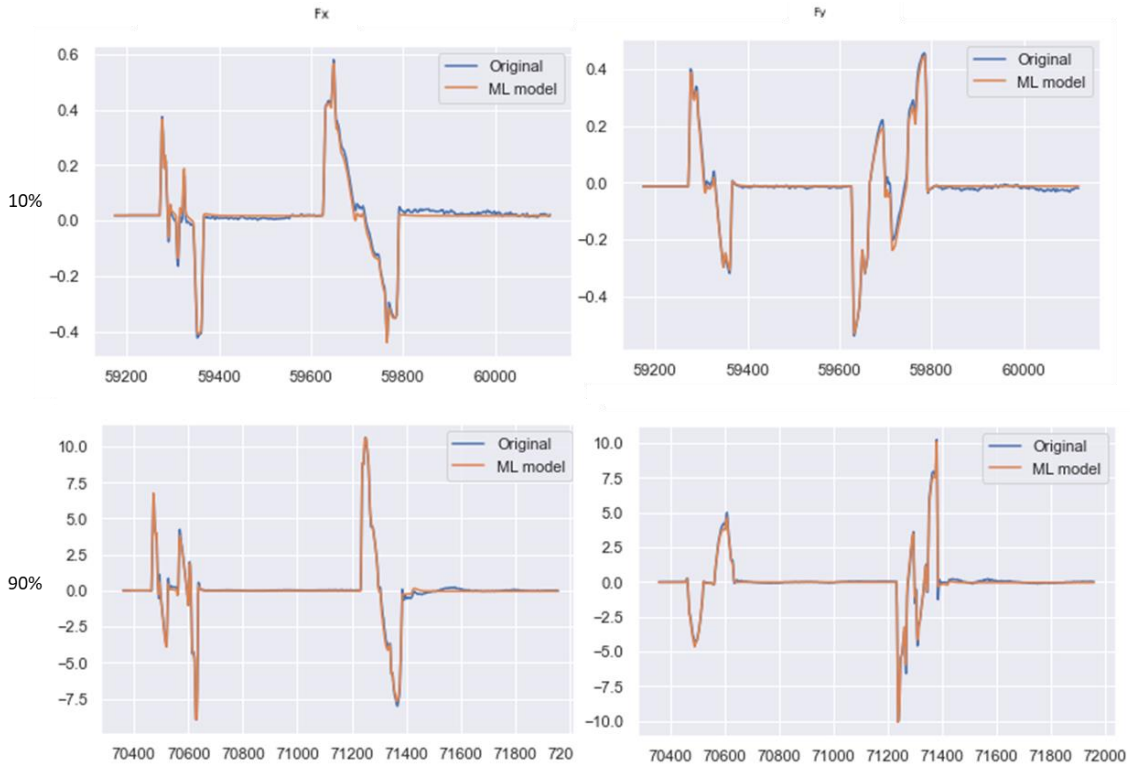

Figure 12: Machine Learning model prediction for filling ratios of 10% (above) and 90% (down).

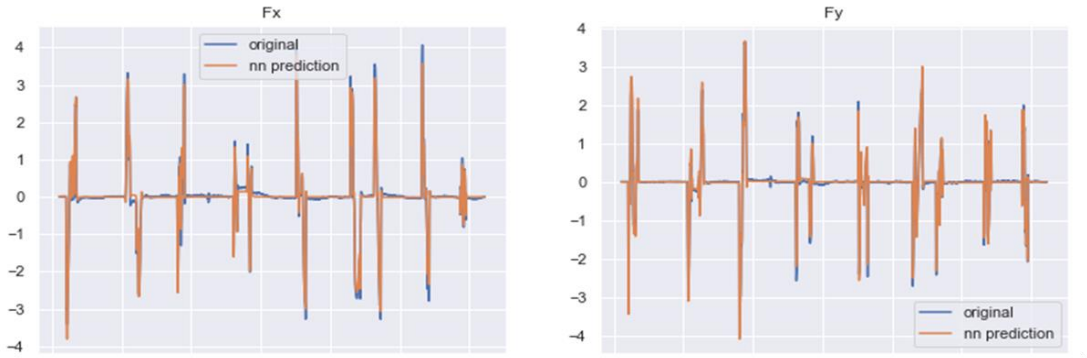

Figure 13: Machine Learning model prediction for profiles with 10 maneuvers.

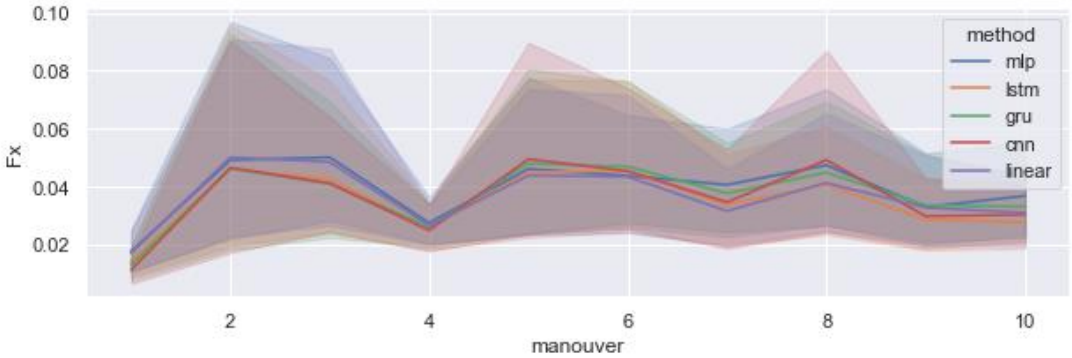

Figure 14: *Fx* MSE prediction distribution as function of the maneuver.

5.4 Training data size impact

In this section, the impact of the training data size on the prediction error is assessed. This evaluation is performed on the 600 initial simulations, with the model that provides the best scores, the h-CNN. In the previously presented results, 60% of the data was used for training. We want to know if equivalent quality results can be obtained with less data.

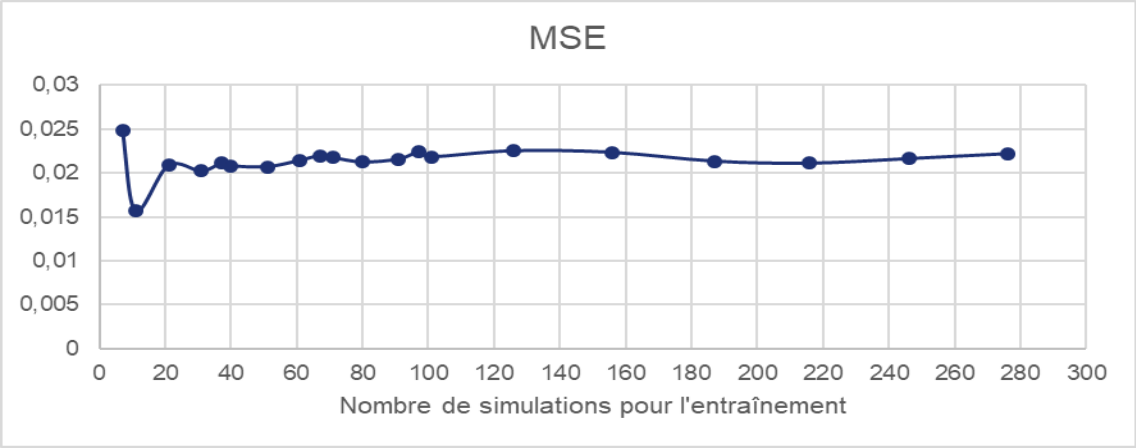

Figure 15: Prediction MSE as function of the number of simulations used for the training.

It can be observed that the final order of magnitude is obtained from about only 20 simulations. This result is fundamental for the applicability of our method, since it reduces the necessary CFD simulations to parameterize our method.

5.5 Computational time performance

We finally discuss the performance of the models in terms of computation time. Generating data through CFD simulations is very expensive. In contrast, the mass-spring model predicts the result quasi-instantaneously. Regarding models based on Deep Learning methods, two distinct computation times need to be considered: the training time and the prediction time.

The model training takes place only once for each use case. In the case of training on about 360 simulations, we estimate the training time of Deep Learning models to be around 20 minutes. To include hyperparameter optimization, 30 trials of different configurations were performed, for a total time of 25 hours. Two significant levers can be used to reduce this total time:

- The automatic stopping of the hyperparameter optimization search: we estimate that we can settle for half of the 30 trials performed to obtain a converged solution; an automatic stopping criterion would optimize the number of trials.
- The reduction of the number of simulations in the training set: we have seen that the errors obtained with 360 simulations can actually be obtained with only about twenty simulations. Reducing the amount of input data for learning can significantly reduce the time of each execution.

The prediction time is the time required to predict the outputs of a simulation on new data with a previously trained model. The following table presents the prediction times (in seconds) for one simulation with n maneuvers for each of the models, compared to the mass-spring model and an estimate of the CFD simulation time:

Table 5: Computational time (in seconds) of each method.

N° Maneuvers	MLP	h-MLP	h-CNN	h-LSTM	h-GRU	LM	Mass-spring	CFD
2	0.080	0.194	0.199	0.320	0.319	0.150	0.282	480
10	0.100	0.430	0.456	1.004	1.055	0.331	0.285	2700

We notice the significant difference in orders of magnitude between the prediction times of the different models and the CFD simulation times: the latter are about 1000 times higher. Therefore, with a trained model, it is possible to perform 1000 times more simulations for a constant computation time budget.

6 CLOSED LOOP SIMULATIONS: MACHINE LEARNING IN SIMULINK

A final section is dedicated to the validation of the Machine Learning model embedded into a simplified Guidance, Navigation and Control (GNC) simulator (cf. Figure 16) in Matlab/Simulink. This is a key asset because it will allow us to implement the Machine Learning model, previously coded in Python, all along the life cycle of the project (feasibility, design and validation phases).

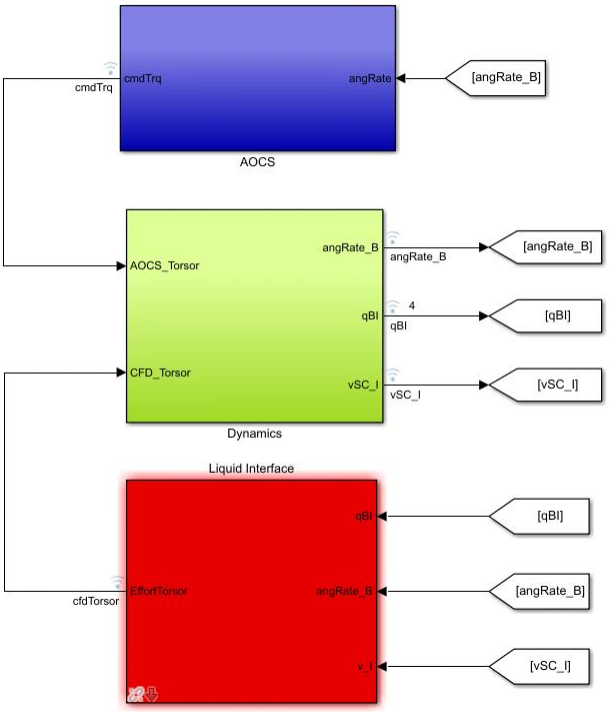

Figure 16: GNC simplified simulator (Simulink model).

Within this simulator, three models were implemented in the block "Liquid Interface" and compared:

- The Machine Learning model implemented in Python
- The CFD Model implemented in FLUENT software
- The mass-spring model

Note that this closed loop analysis has been only done on the PoC case.

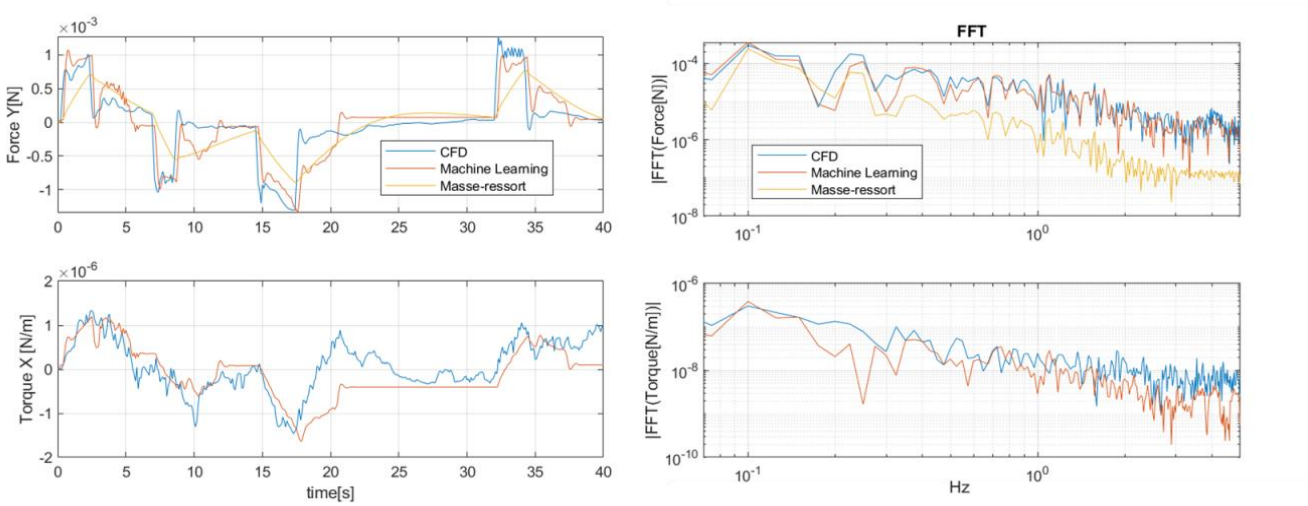

Figure 17: Closed-loop comparison prediction (left) and FFT of the signals (right).

For 40 seconds of simulation, the computational time for the three models are:

	Mass-spring	**Machine-Learning**	**CFD**
Time[s]	3	150	7480

Once again, we notice that the CFD computational time is an order magnitude above the Machine-Learning model. Furthermore, this difference is observed for the PoC case, where the tank diameter is short (10 cm diameter). We expect a significant increment of this difference with a representative case, as the MR case. We expect to carry out this analysis in the short-term.

7 CONCLUSIONS

Hybrid AI methods produce high-quality and reliable predictions.
Hybrid methods presented can accurately predict forces and moments, both in the time and frequency domain, with a fidelity to CFD simulation data that is approximately ten times better in terms of MSE than simplified models.

The incorporation of contextual knowledge (linear regime) significantly improves predictive capabilities.
The generation and analysis of the data revealed a strong numerical and physical singularity, namely its linearity. Exploiting these peculiarities, supported by domain knowledge, guarantees a significant improvement in the prediction capabilities. Therefore, we emphasize the importance of incorporating domain knowledge in the choice of the models architecture to facilitate the neural network's learning task.

Hybrid AI methods are generalizable.
Robustness analysis concludes that hybrid models trained on a dataset could be applied to predict data with different characteristics, such as different filling ratios and numbers of maneuvers. Moreover, our experiments on new datasets yield to similar results to those of the initial dataset, with comparable error magnitudes.

AI methods are computationally efficient.
Finally, all the proposed AI methods are particularly efficient in terms of prediction time (after training). Like the mass-spring model, AI models are drastically faster than CFD simulations. On average 1000 times faster than CFD simulations.

8 ACKNOWLEDGMENTS

The work described in this paper was carried out under a R&D study cofounded with CNES, who has followed all the activities related to modelling the Sloshing phenomena using Artificial Intelligence. Airbus would like to thank CNES for their trust in Airbus technical teams to carry out such promising solutions.

9 REFERENCES

[1] N.H. Abramson, *The dynamic behaviour of liquids in moving containers, with applications to space vehicle technology*. Tech. Rep. NASA SP 106.
[2] A. Dalmon, *Simulation numérique du ballottement d'ergol et modélisation de l'interaction fluides-membranes*, PhD Thesis, University of Toulouse 3, 2018
[3] R. Pierre, R. Roumiguié, D. Lasnet and J. Mignot, "Fluid dynamics in space experiment", *ESA GNC conference*, 2017