RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

arianeGROUP

ONERA
THE FRENCH AEROSPACE LAB

On sloshing control for launcher upper stages

Sloshtember InterAgency Lectures

Périclès Cocaul[1], Martine Ganet[2]
pericles.cocaul@onera.fr
martine.ganet@ariane.group

- September 29^{th} 2022 -

[1] 1^{st} year PhD student ONERA/ArianeGroup , [2] GNC Expert, ArianeGroup, Les Mureaux

Ce document est la propriété de l'ONERA. Il ne peut être communiqué à des tiers et/ou reproduit sans l'autorisation préalable écrite de l'ONERA, et son contenu ne peut être divulgué.
This document and the information contained herein is proprietary information of ONERA and shall not be disclosed or reproduced without the prior authorization of ONERA.

Sloshing problem in attitude control

- Liquid movement in tanks
- Lead to unmodelled dynamics
- Can degrade stability margins
- Microgravity increases sloshing impact
- Can lead to important anomalies: NEAR (safe mode in 1998), Solar Dynamics Observatory (2010) **[1]** or Falcon 1 (failure in 2007)
- Difficulties to model

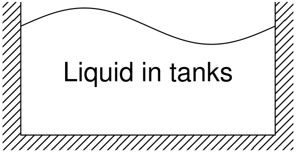


Figure: Sloshing in tanks

Desired characteristics

Performances:

- Ability to reject disturbances
- Ability to stabilize the system in case of unmodelled dynamics

Goals:

- Simplify of the design phase
- Widen the operating range: more robustness
- Complete dynamics to take into account or adaptation to non modeled problems

Tracks considered:

- Avoiding a particular model $\rightarrow$ need for model-free methods
- Development of controllers from the automatic and machine learning point of views
- Search for stability guarantees
- Adaptability of the chosen solution for application on several domains and missions

Actual solutions

- Actual ways of control:
  - Robust linear control **[2],[3]**
  - Structured H_∞ synthesis **[4]**
  - Reference control with a switched adaptive strategy **[5]**
  - Adaptive augmenting control algorithms **[6]**
- Concluding results of model-free control in various sectors led to study this possibility **[7],[8]**

Case of study

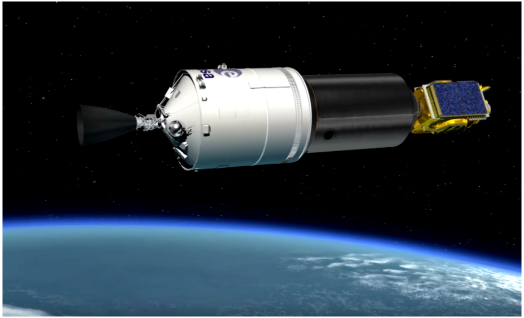


Figure: Exo-atmospheric phase with Ariane 6's third stage

Problem Statement:

- Short boosts
- Mitigate sloshing phenomena
- Reach the expected launcher attitude
- Avoid to rely on the dynamics representation

Command law for A6 Upper Liquid Propulsion Module (ULPM) with sloshing:

- Automatic reference: Structured analysis
- Find a solution on a test case with neural networks
- Step-by-step approach on a constrained system

Objectives

- Control of the sloshing phenomena
- Control of the launcher's attitude
- Ensuring the stability of this nonlinear system
- Minimizing the transverse acceleration
- Handling the saturations inherent to the device design
- Uncertainties on the physical parameters which do not allow to perfectly represent the full dynamics
- Compensate delays

Model representation

- Computational Fluid Dynamics (CFD) can represent this issue **[9]**
  - High computational time
  - Need data in microgravity
- Equivalent Mechanical Model: spring-mass or pendulum **[10]**

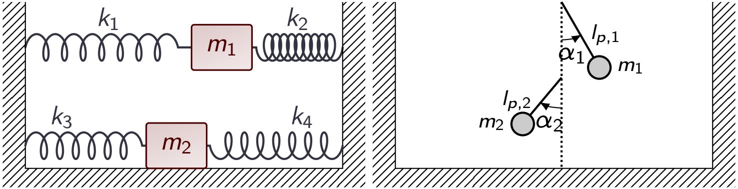


Figure: Sloshing representation with spring-masses or pendulums

Sloshing model

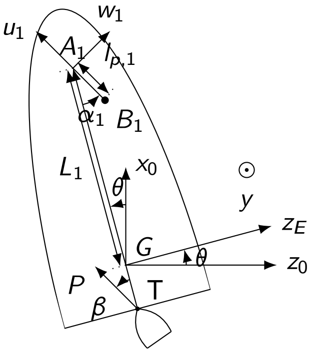


Figure: Sloshing model with one oscillating mass

- Control is effected by deflection of the nozzle: angle β
- Complex problem: two pendulums with different attachment points, lengths and masses

State space representation

$$X = \begin{pmatrix} \alpha_1 & \dot{\alpha_1} & \alpha_2 & \dot{\alpha_2} & \dot{z} & \theta & \dot{\theta} \end{pmatrix}^T$$

$$\dot{X} = AX + BU$$
$$Y = CX + DU$$

$$\text{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ a_{11} & da_{11} & a_{12} & da_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_{21} & da_{21} & a_{22} & da_{22} & 0 & 0 & 0 \\ c_1 & dc_1 & c_2 & dc_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ g_1 & dg_1 & g_2 & dg_2 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{B} = \begin{pmatrix} 0 & b_1 & 0 & b_2 & -\frac{T}{M} & 0 & \frac{TL_t}{I} \end{pmatrix}^T$$

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$a_{i,j} = -\frac{\gamma}{l_{p_i}} \left(\delta_{i,j} + \frac{m_j}{M} - \frac{m_j\, L_j\, (l_{p,i} - L_i)}{I} \right)$$

$$da_{i,j} = -2\, \xi_j\, \frac{\sqrt{\gamma l_{p_j}}}{l_{p_i}} \left(\delta_{i,j} + \frac{m_j}{M} + \frac{m_j\, (l_{p,j} - L_j)\, (l_{p,i} - L_i)}{I} \right)$$

$$c_i = \frac{m_i\, \gamma}{M}$$

$$dc_i = \frac{2\, m_i\, \xi_i}{M} \sqrt{\gamma\, l_{p,i}}$$

$$g_i = -\frac{\gamma m_i\, L_i}{I}$$

$$dg_i = -2\, \xi_i \sqrt{\gamma l_{p_i}}\, \frac{m_i\, (l_{p,i} - L_i)}{I}$$

$$b_i = -\frac{T}{ML_i} \left(\frac{1 - ML_t(l_{p,i} - L_i)}{I} \right)$$

with: γ: launcher's acceleration

ω_i: eigen-pulsation of sloshing mode i

ξ_i: natural damping of the sloshing mode i

Structured H_∞ synthesis

- Comparison made with a structured H_∞ controller **[11]**
- Obtention of PID and estimator gains
- Estimator takes the place of the integrator $\rightarrow$ PD-controller considered

Objectives

Establish a baseline with a well developed method
Minimize response time and overshoot + stabilize via phase control
$\rightarrow$ Definition of an exclusion contour

Objectives of the model-free method

- Reject perturbations and unknown deviations from the rigid dynamics
  $\rightarrow$ Estimation of a deflection bias
  $\rightarrow$ Adaptation to delays, uncertainties, sloshing perturbations, saturations
  Control law based on an ultra-local model
- Main characteristic evaluated: capacity to control the sloshing modes
- Stability guarantees not investigated further than with the following study **[12]**

Model-free control: Structural basis

Ultra-Local Model: plant behavior supposed to be well approximated by a system of ordinary differential equations:

$$y^{(n)}(t) = \phi(t) + \zeta\, u(t) \tag{1}$$

here: $$\ddot{y}(t) = \phi(t) + \zeta\, u(t) \tag{2}$$

With: $\phi(t)$: structural information

ζ: non-physical constant parameter

Using a Laplace transform, Cauchy's formula and bringing it back in the time domain:

$$\hat{\phi}(t) = \frac{60}{T^5}\int_0^T \left(T^2 - 6T\tau + 6\tau^2\right) y(\tau)d\tau - \frac{30\zeta}{T^5}\int_0^T (T-\tau)^2\tau^2 u(\tau)d\tau \tag{3}$$

Model-free control: Structural basis

A trapezoidal approximation gives:

$$\hat{\phi}(t) = \frac{60}{T^5}\sum_{k=0}^{N} a_k\left(T^2 - 6kT_s + 6(kT_s)^2\right)y(k) - \frac{30\zeta}{T^5}\sum_{k=0}^{N} a_k\,(T - kT_s)^2\,(kT_s)^2\,u(k) \tag{4}$$

$$\text{with : } a_k = \begin{cases} T_s/2, & k = 0 \text{ and } k = NT_s \\ T_s, & k = 1, \quad \ldots, n-1 \end{cases}$$

where T_s is the sampling period and N is chosen such as $N.T_s = T$.
Controller form:

$$u(t) = \frac{-\hat{\phi}(t) + \ddot{y}^* - K_I\int e - K_D\dot{e} - K_P e}{\zeta} \tag{5}$$

With: $\ddot{y}^*$: the second order derivative of the reference output trajectory

e: error between the measured output y and the reference value

K_p, K_i, K_d: gains values

Approach parameters

Parameters to tune for this method:

- ζ: has a major impact on response. Chosen to have $\zeta\ u$ and $\ddot{y}$ of the same order of magnitude
- T_s: step of integration window
- N: number of points considered

The integration window is given by: $T = T_s * N$
$\rightarrow$ As in every control method, a trade-off needs to be found and performances are limited.

Reinforcement Learning

Objectives

- Complexity for non-linear systems
- Whole controller into a single function (usually a Neural Network) directly designed from actions and observations
- Good performances reached with RL for control **[13]**
- Find stability and exploration guarantees **[14],[15],[16]**

Reinforcement Learning: Characteristics

Optimal policy

The agent builds a policy to map states to actions. Using action-value functions, policies can be compared and an optimal one can be found. The desired behavior is reached for the best policy.

Search method

Constrained optimization problem needs to be solved. Use of policy gradient methods: Proximal Policy Optimization (PPO).

- Big improvement step without stepping too far.
- Use of a clipping ratio in the objective function to remove incentives from the new policy to get far from the old one.

Reinforcement Learning: Characteristics

Approximate dynamic programming

Approximates the optimal control cost from data. Computes optimal controls by applying the policy maximizing the Bellman equation

Actor/critic

Combination of policy and value searches. Composed of two subsystems:

- Actor learns an action from a state given by the environment
- Critic receives the state and reward resulting from the previous step

Tradeoff between variance reduction (policy gradients) and bias introduction (value function methods)

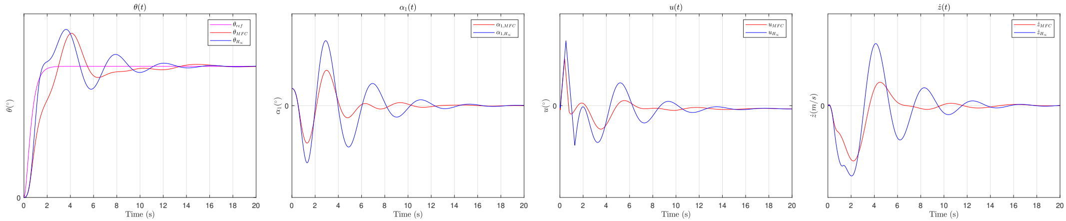
Reward function

A reward function maps states and actions to real number

Simulation features

1. Initial conditions: non-zero sloshing angles
2. Deflection bias
3. Delays
4. Saturations
5. Two configurations with one then two sloshing modes:
   - Attitude command
   - Recovery from initial conditions

MFC/H_∞, step in attitude, one sloshing mode

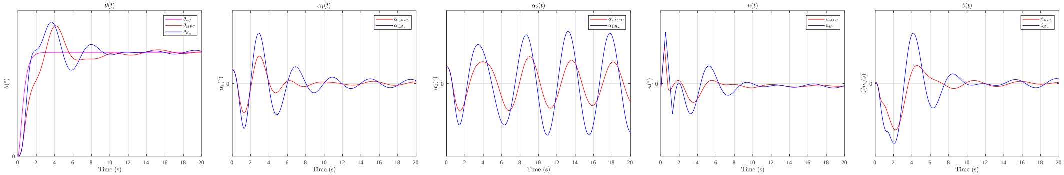


(a) $\theta(t)$ (b) $\alpha_1(t)$ (c) $u(t)$ (d) $\dot{z}(t)$

Figure: Step in θ with one sloshing mode (red: MFC, blue: H_∞, magenta: consign)

- Faster damping in α with MFC than with H_∞ synthesis: about 36 % quicker to reach the admissible set
- $u(t)$ respects saturations
- $\dot{z}(t)$ reaches zero

MFC/H_∞, step in attitude, two sloshing modes



(a) $\theta(t)$ (b) $\alpha_1(t)$ (c) $\alpha_2(t)$ (d) $u(t)$ (e) $\dot{z}(t)$

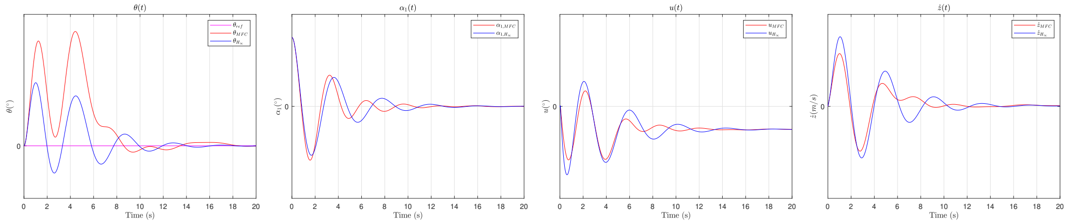
Figure: Step in θ with two sloshing modes (red: MFC, blue: H_∞, magenta: consign)

Adding a sloshing mode $\rightarrow$ second mode angle is not controlled + small oscillations on first mode angle

$u(t)$ and $\dot{z}$ converge with small oscillations due to second sloshing mode

MFC/H_∞, recovery from initial conditions, one mode

Another possible goal: recover from initial conditions with perturbation and uncertainties + keeping θ close to zero



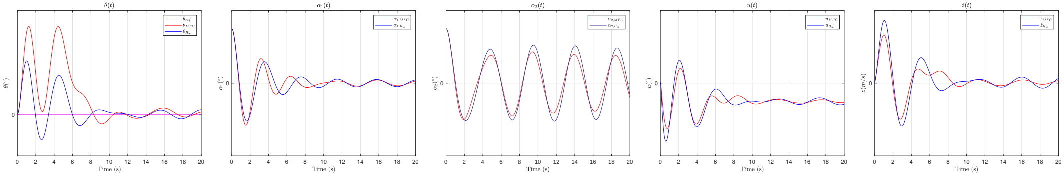
(a) $\theta(t)$ (b) $\alpha_1(t)$ (c) $u(t)$ (d) $\dot{z}(t)$

Figure: Recovery with one sloshing mode (red: MFC, blue: H_∞)

Sloshing angle well controlled: reaches admissible set in about 10 % less time than with structured synthesis

Wider variations in θ needed at the beginning

MFC/H_∞, recovery, two sloshing modes



(a) $\theta(t)$ (b) $\alpha_1(t)$ (c) $\alpha_2(t)$ (d) $u(t)$ (e) $\dot{z}(t)$

Figure: Recovery with two sloshing modes (red: MFC, blue: H_∞)

As for the θ command, second mode brings small oscillations on every state variable. Otherwise same results as with one sloshing mode

Main notable difference for model-free: quicker mitigation of sloshing modes (10 % faster than with the structured analysis) but with a wider attitude range of values

Recovery with RL, two sloshing modes, same study case

- Promising results found by ArianeGroup on a different study case
- Still working on adapting the algorithm to this issue
- Some differences in the parameters take time to adjust
- Comparison of the three methods still to come
- Two comparison baseline objectives:
  $\rightarrow$ Time to reach admissible sloshing angle
  $\rightarrow$ Set a Monte Carlo to compare the robustness of each method

Results

- Complex problem with uncertainties and perturbations $\rightarrow$ quick control of the sloshing phenomena
- Second sloshing mode not amplified but not controlled $\rightarrow$ brings small oscillations for every state variable
- MFC controller: quicker control of $\alpha_1(t)$ than H_∞ synthesis but more difficulties to reach the θ command
- Transverse velocity $\dot{z}$ brought to zero for the first two methods
- Saturations quickly limit the range of possible values with H_∞ synthesis. Three times wider range of values with model-free. Not yet studied for RL
- RL is currently applied on this study case. Seems promising for such studies.

Conclusion of the study

- Ultra-local representation of the dynamics $\rightarrow$ evaluation of the dynamics unknown parts
- Careful choice of design parameters
- MFC proved to be useful for a launcher's attitude control. Tests are still running with RL for the same study case
- Methods seem useful for future developments, particularly RL
- Need to run a Monte Carlo to examine the robustness
- Several possibilities to guarantee safety: use of Lyapunov functions during the learning part, construction of a Lyapunov function with an additional Neural Network layer, MPC with RL.

Perspectives for the study:

- Improve the choice method of parameters in particular ζ and T_s
- Adaptive values for the parameters to improve robustness for the MFC
- Finish and widen the scope of comparisons between the three methods

Perspectives for the PhD

Main objective

Bridge the gaps between the ML and Automatic communities

$\rightarrow$ Use the performances of ML and guarantees from automatic

- Leads for future work:
  - RL controllers: on-policy model-free RL algorithms
  - Reward Engineering: easy way to design rewards, to use them on different applications (Guidance, FDIR)
  - Deep Stability Analysis: NN to find a Lyapunov function, explicit safety guarantees during learning
  - MPC with RL: Receding Horizon Control

Future applications

Application to a multi-engine bay

→ Command adaptation with actuators faults

Trajectory replanification to adapt a guidance law

→ Use of an evaluation function which will allow to guarantee stability and good performances
→ Definition of criteria based on reachable and mesurable states to match the mission objectives

Thank you for your attention !

Questions and comments are welcome

pericles.cocaul@onera.fr

martine.ganet@ariane.group

References I

[1] P. Mason and S. Starin, "The effects of propellant slosh dynamics on the solar dynamics observatory," in *AIAA Guidance, Navigation, and Control Conference*, p. 6731, 2011.

[2] O. Voinot, D. Alazard, and A. Piquereau, "A robust multi-objective synthesis applied to launcher attitude control," *IFAC Proceedings Volumes*, vol. 34, no. 15, pp. 203–208, 2001.

[3] M. Imbert, M. Gauvrit, and A. Piquereau, "Robust techniques application for attitude control of a launcher during atmospheric flight," *IFAC Proceedings Volumes*, vol. 22, no. 7, pp. 197–201, 1989.

[4] P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

[5] A.-R. Luzi, D. Peaucelle, J.-M. Biannic, C. Pittet, and J. Mignot, "Structured adaptive attitude control of a satellite," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 7-8, pp. 664–685, 2014.

[6] J. H. Wall, J. S. Orr, and T. S. VanZwieten, "Space launch system implementation of adaptive augmenting control," in *Annual American Astronautical Society (AAS) Guidance, Navigation, and Control Conference*, no. AAS 14-051, 2014.

[7] J. Villagra and C. Balaguer, "A model-free approach for accurate joint motion control in humanoid locomotion," *International Journal of Humanoid Robotics*, vol. 8, no. 01, pp. 27–46, 2011.

[8] H. Abouaïssa, M. Fliess, V. Iordanova, and C. Join, "Freeway ramp metering control made easy and efficient," *arXiv preprint arXiv:1206.5937*, 2012.

References II

[9] H. Yang, R. Purandare, J. Peugeot, and J. West, “Prediction of liquid slosh damping using a high resolution cfd tool,” in *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 4294, 2012.

[10] F. T. Dodge *et al.*, *The new” dynamic behavior of liquids in moving containers”*. Southwest Research Inst. San Antonio, TX, 2000.

[11] P. Apkarian, “Tuning controllers against multiple design requirements,” in *2012 16th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 1–6, IEEE, 2012.

[12] E. Delaleau, “A proof of stability of model-free control,” in *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, pp. 1–7, IEEE, 2014.

[13] G. Waxenegger-Wilfing, K. Dresia, J. C. Deeken, and M. Oschwald, “A reinforcement learning approach for transient control of liquid rocket engines,” *arXiv preprint arXiv:2006.11108*, 2020.

[14] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” *Advances in neural information processing systems*, vol. 30, 2017.

[15] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, “Lyapunov-stable neural-network control,” *arXiv preprint arXiv:2109.14152*, 2021.

[16] S. M. Richards, F. Berkenkamp, and A. Krause, “The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems,” in *Conference on Robot Learning*, pp. 466–476, PMLR, 2018.