HOW WE SAVED THE ROSETTA MISSION

Pierre APKARIAN[1], Dominikus NOLL[2], and Alexandre FALCOZ[3]

Abstract. The Rosetta mission was launched in 2004 and ended in 2016, after the lander Philae had been successfully placed on the comet Churyumov - Gerasimenko in 2014. What is less known is that the mission came within an ace of failure due to a fault in one of the thrusters, which was discovered in 2011. The problem was serious, and it took until 2014 to design new robust controllers and to uploaded them. Ultimately this became possible only with the help of our control software `hinfstruct`. When we invented it in 2004, we inadvertently saved the Rosetta mission.

Key words: Rosetta mission · how we saved it · H_∞-control · our software · `hinfstruct`

1. The Rosetta story

The Rosetta mission of the European Space Agency is one of the great success stories of recent space exploration. Launched on march 2, 2004, the *Rosetta* orbiter reached the comet Churyumov - Gerasimenko, for short *Chury*, in 2014 after a 10 year flight, and subsequently Rosetta's lander *Philae* was placed on the comet on november 12, 2014.

What is less known is that the mission came within an ace of failure, because in 2011 a fault in thruster no. 9 was diagnosed, see Fig. 1. The thruster had a loss of efficiency of 7.4%, which caused strong parasite torques. The fault could not be compensated by the original feedback control software, putting the mission at stake. Good advice was dear, and it took indeed 4 years until new controllers could be designed and uploaded in 2014. The design of these new controllers was possible due to our seminal control software `hinfstruct`, which we had invented between 2004 and 2006 [1], and which we had made available to the control engineering community since 2010 by dint of MathWorks [5]. This is how we inadvertently had saved the mission even before it run into trouble.

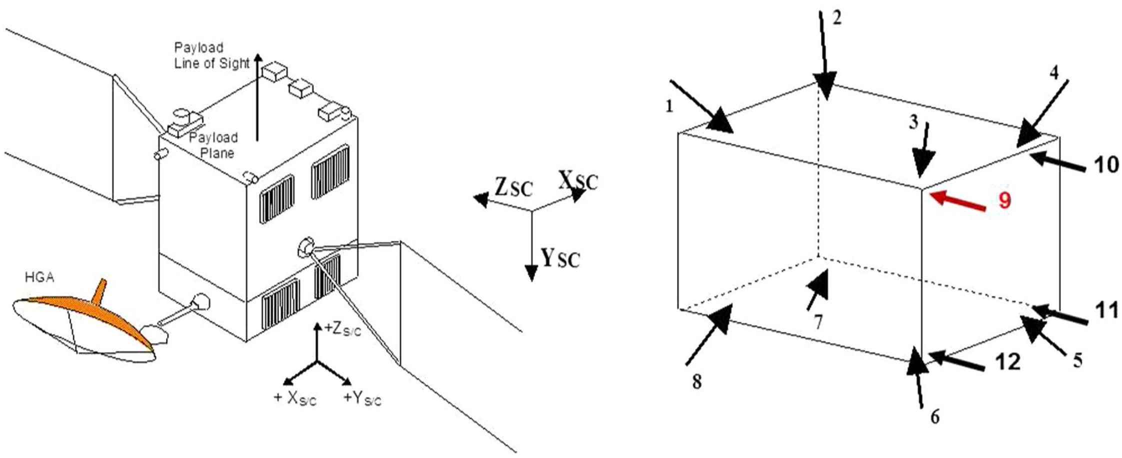


Fig. 1. Axis off-pointing of thruster no. 9 discovered in 2011 just before Rosetta went into a 31 month hibernation. At that time our software was available on request, but not known to operators Airbus or

[1]ONERA, Department of System Dynamics, Toulouse, France.
[2]Institut de Mathématiques, Université de Toulouse, France.
[3]Airbus Defence & Space.

CNES. It took full 5 years before the problem could be fixed in 2014. Only now could the final insertion maneuver of the probe positioning Rosetta behind the comet could be accomplished.

2. The time table

Rosetta launched march 2, 2004 on Ariane 5 carrier
First fly-by-earth march 3, 2005
    2004-06 `hinfstruct` prototype. Published 2006 in [1]
Fly-by-mars february 26, 2007
Second fly-by-earth november 14, 2007
Fly-by Asteroid Steins september 5, 2008
Third fly-by-earth november 11, 2009
Fly-by Asteroid 21 Lutetia july 10, 2010
Deep-space maneuvers 2011
    Off-pointing of thruster no. 9 discovered just before hibernation[1]
Rosetta hibernation june 2011 – january 2014
Before 2014 Airbus and CNES fail to design new controllers due to inappropriate software.
    March 2014 new controllers designed using our software `hinfstruct`
New controllers uploaded may 2014 – just in time.
Breaking and final insertion maneuver august 6, 2014
Philae lands on surface november 12, 2014
Mission ends september 30, 2016.

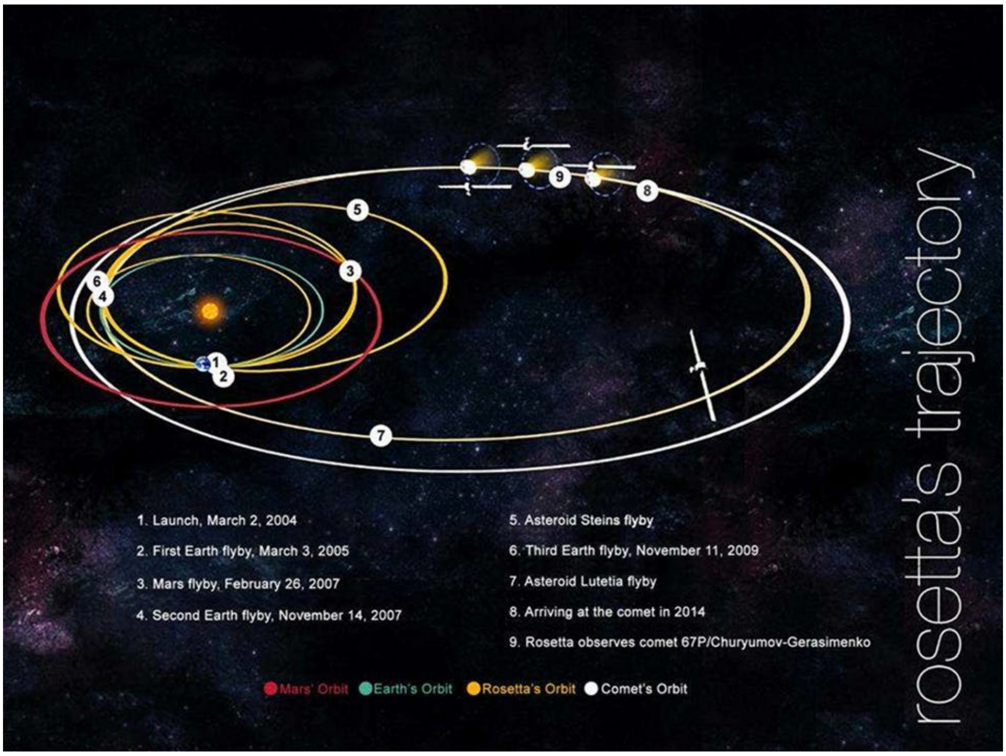


Fig. 2. Rosetta time table revisited. (1) Rosetta launched and `hinfstruct` conceived. (2) `hinfstruct` prototype. (3) Mars fly-by and `hinfstruct` algorithm already published [1]. After (6) failure of thruster no. 9 discovered. Hibernation starts. During hibernation analysis of problem. Operators do not succeed

[1]Teams of the European Space Operations Centre (Darmstadt) were instrumental in detecting and diagnosing the dysfunction. Airbus Defence and Space was charged to synthesize new control laws taking into account the unexpected attitude of the probe, and accomplished this using our code `hinfstruct`. Controllers were uploaded by ESOC.

to fix problem with their own software, but fortunately realize that `hinfstruct` exists and use it to design robust controllers. Just before breaking at (8) `hinfstruct` controllers uploaded.

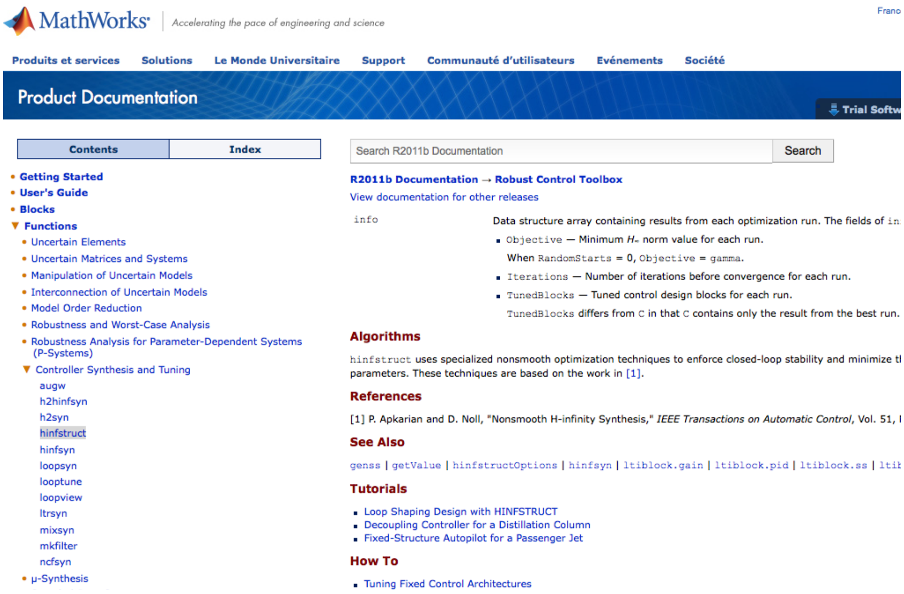


Fig. 3. Function `hinfstruct` based on [1] was developed 2004-06 and adopted by MathWorks in 2010. This represents a fundamental change in feedback control, as for the first time since the 1960s the H_∞-control paradigm can be used in practice.

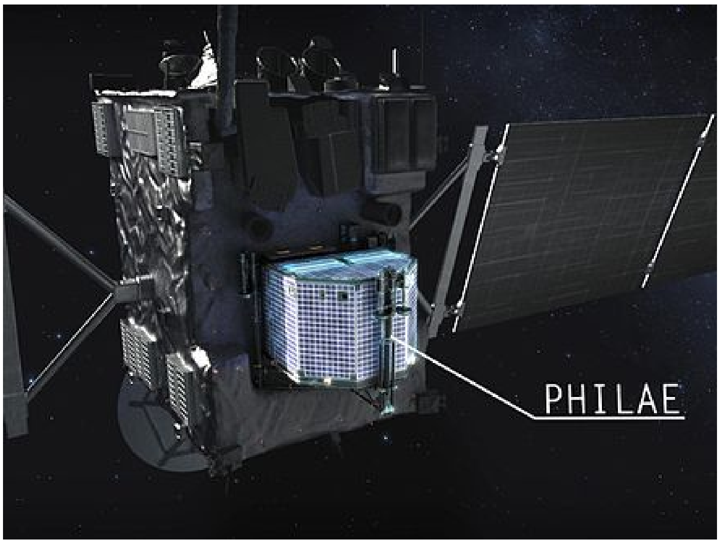
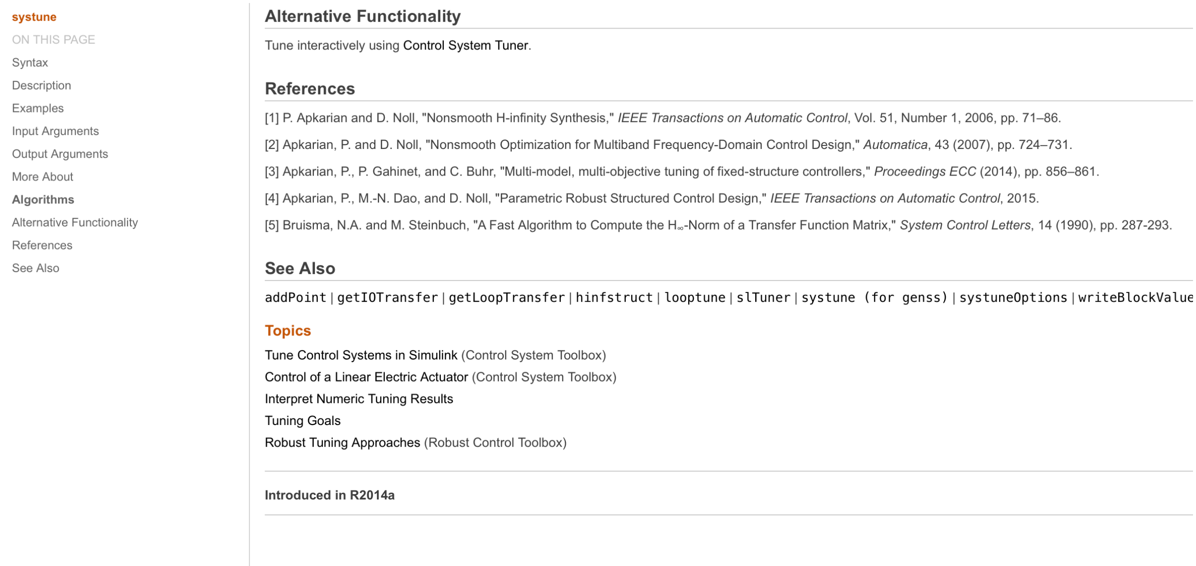


Fig. 4. Rosetta satellite with lander Philae. If not repaired, the dysfunction of thruster no. 9 would have made the final insertion maneuver of Rosetta behind the comet Chury impossible. Once this position was reached, the Philae story was ready to be told.

3. The context

Designing new attitude controllers for the Rosetta probe turned out impossible with the original software developed by CNES and Airbus Defence and Space. Only with our novel structured H_∞-control method `hinfstruct` based on [1, 2] was it possible to design appropriate regulators [6], the main limitations being that the on-board architecture accepted only controllers of order 8 or less. This is a typical example of what we call a *structural contraint* on the control law. Applying the H_∞-paradigm to design structured controllers was not possible before the seminal work [1, 2].

While robust control design has now become the standard approach, inspecting [6, Sect. 4] shows that designing the new controllers for thruster no. 9 could have benefitted by an extended version of `hinfstruct`, which includes *parametric robustness*. And this is where history repeats itself. Namely, in 2014 when this crisis was at its peak, we were just developing this next generation software for parametric robust H_∞-synthesis. In 2014 the parametric robust version was not published, but would have been available on request. In the meantime this new function is also made available to the control engineering community through the design tool `systune` in [5], which is an outcome of our papers [1, 2].



systune

ON THIS PAGE

Syntax

Description

Examples

Input Arguments

Output Arguments

More About

Algorithms

Alternative Functionality

References

See Also

Alternative Functionality

Tune interactively using Control System Tuner.

References

[1] P. Apkarian and D. Noll, "Nonsmooth H-infinity Synthesis," *IEEE Transactions on Automatic Control*, Vol. 51, Number 1, 2006, pp. 71–86.

[2] Apkarian, P. and D. Noll, "Nonsmooth Optimization for Multiband Frequency-Domain Control Design," *Automatica*, 43 (2007), pp. 724–731.

[3] Apkarian, P., P. Gahinet, and C. Buhr, "Multi-model, multi-objective tuning of fixed-structure controllers," *Proceedings ECC* (2014), pp. 856–861.

[4] Apkarian, P., M.-N. Dao, and D. Noll, "Parametric Robust Structured Control Design," *IEEE Transactions on Automatic Control*, 2015.

[5] Bruisma, N.A. and M. Steinbuch, "A Fast Algorithm to Compute the H_∞-Norm of a Transfer Function Matrix," *System Control Letters*, 14 (1990), pp. 287-293.

See Also

`addPoint` | `getIOTransfer` | `getLoopTransfer` | `hinfstruct` | `looptune` | `slTuner` | `systune (for genss)` | `systuneOptions` | `writeBlockValue`

Topics

Tune Control Systems in Simulink (Control System Toolbox)

Control of a Linear Electric Actuator (Control System Toolbox)

Interpret Numeric Tuning Results

Tuning Goals

Robust Tuning Approaches (Robust Control Toolbox)

Introduced in R2014a

Fig. 5. The successor of `hinfstruct` is `systune`, based on [1,2], published in 2006, and available in the robust control toolbox since 2014. Parametric robustness is published in 2015 [3] and made available through [5] since 2015.

4. The mathematical background

The authors of [1, 2] started their academic cooperation in the late 1990s. We had recognized that a radically new approach to feedback control design was required, which would allow to bridge theoretical results obtained by the robust control community since the early 1990s with the challenging constraints of engineering practice. This ultimately led to the powerful tool `hinfstruct`, which for the first time enables engineers to apply the H_∞-paradigm in practice.

This strongly hinges on the use of sophisticated non-differential non-convex optimization methods, which we developed since the early 2000s. Our idea to create such a structured H_∞-theory was first presented at a Workshop in Toulouse in 2001, and even though it had to face strong opposition by adherents of the LMI-theory, the resistance was quickly overcome due to the evident superiority of our techniques. Today non-smooth H_∞-synthesis is undisputedly the standard way to solve difficult synthesis problems.

References

[1] P. Apkarian, D. Noll. Nonsmooth H_∞ synthesis. *IEEE Trans. Automat. Control*, 51 (2006), no. 1, 71 – 86.

[2] P. Apkarian, D. Noll. Nonsmooth optimization for multi-disk H_∞-synthesis. *European Journal of Control*, 12 (2006), no. 3, 229 – 244.

[3] P. Apkarian, M.N. Dao, D. Noll. Parametric robust structured control design. *IEEE Transactions on Automatic Control*, vol. 60, issue 7, 2015, pp. 1857 – 1869.

[4] D. Noll. Cutting plane oracles to minimize non-smooth non-convex functions. *Set-Valued and Variational Analysis*, 18 (2010), no. 3-4, 531 – 568.

[5] The MathWorks, Inc. *Robust Control Toolbox.*

[6] A. Falcoz, Ch. Pittet, S. Bennami, A. Guignard, C. Bayart, B. Frapard. Systematic design methods of robust and structured controllers for satellites. *CEAS Space Journal.* DOI 10.1007/s12567-015-0099-8