Modeling and Analysis of a Turbojet Fuel System

C. Le Brun, E. Godoy, D. Beauvois, G. Le Pache, and R. Noguera

***Abstract*—The aim of this paper is to study a new concept of fuel system of a turbojet, to design its control system and to validate its behavior using simulation. AMESim is used to represent the physical system and Simulink has been adopted for the implementation of the control laws. A multi-loop PID strategy is then selected. Simulation tools and different methods of their couplings, including exchange, cosimulation and FMI are benchmarked to find which strategy is the best suitable.**

***Index Terms*—Analysis, control, coupling methods, modeling, simulation.**

I. Introduction

In aeronautics, two of the main key design drivers are the reduction of the fuel consumption and the reduction of its environmental impact. These two considerations are taken into account by both aircraft and engine manufacturers. On the engine regulation system level, the first objective is reached by reducing its mass. The second one can be achieved with a more advanced fuel system. This paper deals with the control of a new fuel system of a turbojet. In comparison with classical fuel systems, a hydraulic equipment of the original system has been removed to reduce the mass, which means that standard control of the turbojet is no more effective. After modeling the new system, its couplings are studied and a controller is designed.

This paper is organized as follows: Section II introduces the system and its functioning; the dynamic model is described in Section III and Section IV presents the design of the control system and its performances. Section V compares different methods of simulation, in order to find the most effective one for the system. At last, conclusions are presented in Section VI.

II. Modeling

A. Overview of the Fuel System

The main goal of the turbojet engine is to ensure a requested thrust. To provide it, a fuel mass flow is injected into the combustion chamber. Moreover, some mechanical elements actuated byhydraulic cylinders (described in greater detail later) need to be in a particular configuration to ensure maximal efficiency and to redirect the airflow in order to prevent stall. These goals are reached thanks to the fuel system which is depicted in Fig. 1.

It includes the following elements: a centrifugal pump increases the pressure of the fuel coming from the tank of the aircraft and feeds that fuel to a gear pump. Because of the characteristics of the gear pump, the delivered fuel flow is most of the time more important than the one needed in the combustion chamber. A bypass valve is used to transport the fluid surplus back to the output of the centrifugal pump. The way to control the fuel mass flow rate going into the combustion chamber is to adjust the section of this valve to bypass more or less fuel. In this study, we consider only one hydraulic cylinder but it can be generalized. The one taken into account here is a VSV (Variable Stator Valve) which aims at ensuring that the airflow reaches the high pressure compressor with a right angle to increase its efficiency. A check-valve located upstream the injection line allows to quickly stop the injection of fuel for security purposes. The check-valve is also used to pressurize the fuel flowing into the hydraulic cylinders, enabling the control of the pistons even when no fuel is injected in the combustion chamber.

B. Modeling Approach

As explained above, the system we need to control is mainly hydraulic but can be linked afterwards to other systems, which belong to other physical fields. In order to make the modeling of the system easier, it seems logical to use software adapted to these domains. Moreover, we do not need to take into account 3D phenomena so a 0D/1D tool may be sufficient. Those considerations will point towards multi-physics software. Several studies have dealt with multi-physics tools [1], and [2]. LMS Imagine.Lab AMESim seems to be adapted to the simulation of multi-physics phenomena: it provides a graphical modeling approach which fits the Bond Graph theory, a lot of libraries of components, particularly in hydraulics and solvers which are known to be appropriate for stiff systems.

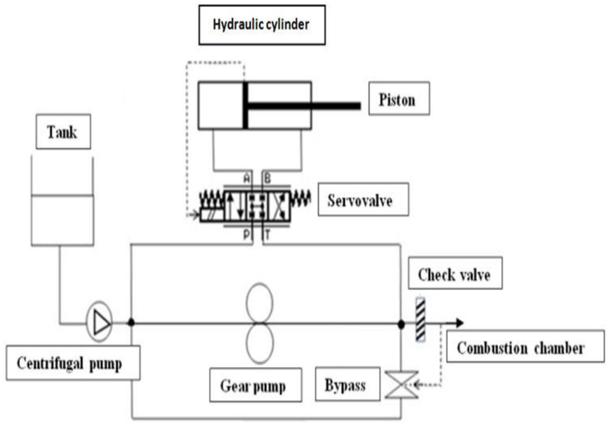


Fig. 1. Drawing of the fuel system.

Unfortunately, despite the availability of a signal/control library, AMESim is not well adapted for the design of controllers. In this domain MATLAB/Simulink remains the

Manuscript received November 4, 2013; revised January 9, 2014.

C. Le Brun, E. Godoy and D. Beauvois are with the Automatic Control Department of Supelec, Gif-sur-Yvette, France (e-mail: {christophe.lebrun, emmanuel.godoy,dominique.beauvois}@supelec.fr).

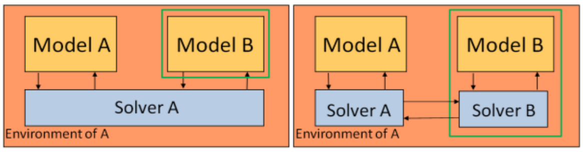
G. Le Pacheis with the System Division of SNECMA (SAFRAN group), Villaroche, France (e-mail: gwenole.lepache@snecma.fr).

R. Noguera is with the DynFluid Laboratory of Arts & M étiers ParisTech, Paris, France (e-mail: ricardo.noguera@ensam.eu).

DOI: 10.7763/IJCTE.2014.V6.872

most commonly used software. For those reasons, these two tools are regularly used together like in [3]-[5]. To benefit from the advantages of these two tools, a coupling approach has been selected.

There are different ways to couple AMESim and Simulink. The different methods have some advantages and drawbacks that will be explained below and the choice of one of these methods depends on its intended use. Fig. 2 shows the difference between import/export and cosimulation strategies: in exchange mode, a model B is imported in an environment A,the solver of which simulates both models while in cosimulation mode, model B and its solver are imported in an environment A and the two solvers work together to simulate the whole model. Both Simulink and AMESim can work with import and export exchange or cosimulation models. [6] gives more information on this subject.



(a) (b)

Fig. 2. Model exchange (a) and cosimulation (b).

Application software has been recently developed to make the coupling of multiple simulation tools easier [7]-[9]. In order to generalize these exchange and cosimulation strategies, research institutes and companies have developed together an independent standard called FMI (Functional Mock-up Interface) [10], [11]. This standard comes from a European project under the name MODELISAR; it is today developed as a Modelica Association Project.

In this paper, we will use the standard coupling between AMESim and Simulink and we will have a brief look at the FMI for export.

C. Time Behavior of the Process

The AMESim implementation of the model of the hydraulic system is shown in Fig. 3. As we have explained above, AMESim uses a graphical modeling approach: each component contains a set of equations which is linked through a system of ports with the equations of the other components. In this model, hydraulic and mechanical libraries are used to model conventional elements. However, the pumps used in this system have been modeled from scratch.

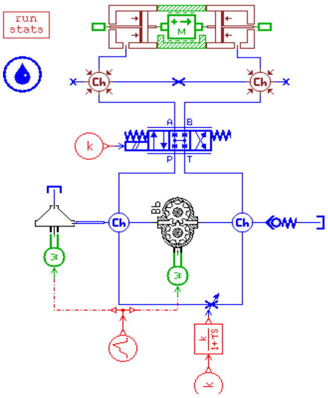


Fig. 3. AMESim model of the system.

1) For the centrifugal pump, the pressure difference on both sides of the pump is linked to its flow rate by:

$$\Delta P = \rho \times \omega^2 \times (\alpha_1 Q^2 + \alpha_2 Q + \alpha_3) \quad (1)$$

ω is the rotation speed of the pump and α_1,α_2,and α_3 are coefficients that depend on the characteristics of the pump.

2) For the gear pump, the flow rate is linked to the pressure difference on both sides of the pump by :

$$Q = cyl \times \omega \times f(\omega, \Delta P, T) \quad (2)$$

cyl is the pump displacement, T the temperature of the fluid and f is a non linear function.

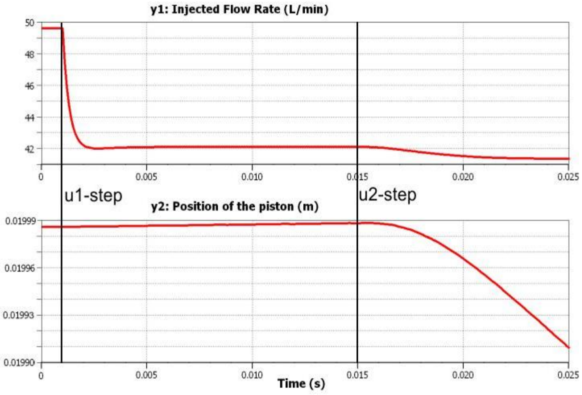


Fig. 4. Responses of the system in open-loop.

The inputs of the system are u_1 and u_2, the current in the bypass valve and in the servo-valve respectively. The outputs of the system are$y_1$the injected flow rate and y_2 the position of the hydraulic cylinder piston.

Fig. 4 presents results of a transient analysis around the operating point y_{1o} = 49.6 l/min, y_{2o} = 0.02 m. $y_1$and $y_2$are the step responses for a u_1-step of 2 mA at 0.001 s and a u_2-step of 4.5 mA at 0.015 s.

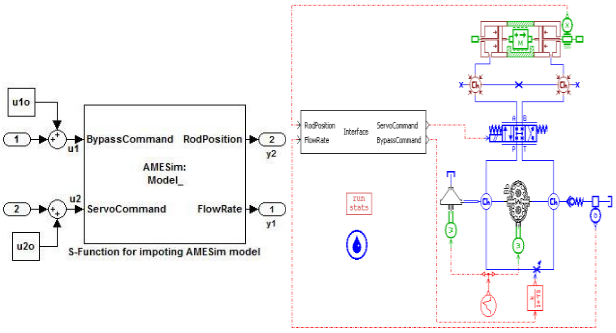
We can see that the injected flow rate$y_1$is mainly impacted by $u_1$and the position of the hydraulic cylinder piston$y_2$byu_2, which is physically logical. This also illustrates the influence of the two inputs on the two outputs. Let us note that the bypass valve input does not really affect the piston position because pressures in each chamber of the hydraulic cylinder are almost affected in the same way.On the contrary,the servo-valve input affects the injected flow rate.

III. Control Law Design

A. Analysis

Frequency analysis is well adapted to the characterization of a system and to the synthesis of a controller. To use this concept, linearization of the system around an operating point is used at first. Results presented here deal with the behavior of the process around the previously considered operating point. Other operating points have been tried and can be used but they do not strongly affect the frequency responses compared to the one used here.

Linearization can be done analytically from the equations of the model: this burdensome and time consuming task leads to a rigorous state-space representation function of the physical variables. Linearization can also be performed using tool-provided numerical methods.



(a) (b)

Fig. 5. AMESim model (b) imported into Simulink thanks to a S-function (a).

The one used here consists in using the linearizing tool of Simulink on the imported AMESim model as shown in Fig. 5.It works in exchange mode but it will not work in cosimulation one. This comes from the fact that in cosimulation, Simulink receives only outputs from AMESim whereas in exchange mode it receives state derivatives as well, which are necessary for the linearization.

Due to numerical errors, the integrating behavior of our system is not perfectly modeled. Despite that, this numerical method almost gives the same results as the analytical solution. It can thus be trusted for this system and will therefore be used in the following of this study.

Table I gives the poles of the system. The system is stable as real parts of the poles are negative or almost null. The first pole μ_1 can be identified as corresponding to the integrating behavior of y_2: even if it is positive, it is extremely weak. The first pair of conjugate poles $\{\mu_{4,}\mu_5\}$ characterizes the servo-valve dynamics; the other pair $\{\mu_{8,}\mu_9\}$ matches with the behavior of the piston and we can see that it is almost not damped. μ_3 characterizes the behavior of the bypass valve, which has beenmodeled as a first order system.

TABLE I: POLES AND NATURAL FREQUENCIES

Poles	Damping	Frequency (rad/s)
μ_1= +5.16×10^{-9}		5.16×10^{-9}
μ_2 = -1.97×10^{1}		1.97×10^{1}
μ_3 = -1.00×10^{2}		1.00×10^{2}
μ_4 = -4.02×10^{2} + 3.02×10^{2}i	0.8	5.03×10^{2}
μ_5 = -4.02×10^{2}– 3.02×10^{2}i	0.8	5.03×10^{2}
μ_6 = -8.97×10^{2}		8.97×10^{2}
μ_7 = -3.17×10^{3}		3.17×10^{3}
μ_8 = -3.87×10^{1} + 6.09×10^{3}i	6.36×10^{-3}	6.09×10^{3}
μ_9 = -3.87×10^{1}– 6.09×10^{3}i	6.36×10^{-3}	6.09×10^{3}

Fig. 6 shows the Bode diagrams of the multivariable system. Low frequency phases of the transfer functions y_1/u_1 and y_1/u_2 are positive, corresponding to the fact that opening the bypass valve and the servo-valve will make the injected flow rate decrease. The low frequency phase of the transfer functions y_2/u_1 and y_2/u_2 are ±90° because of the integrating behavior, the sign of the phase being function of the orientation convention of the piston stroke.

A brief look at the magnitude of the Bode diagrams allows assessing the impact of the inputs on the different outputs in open-loop as in (3). The system is apparently row diagonally dominant at all frequency ranges [12].

$$\left|\frac{y_1}{u_1}\right| \geq \left|\frac{y_1}{u_2}\right|, \quad \left|\frac{y_2}{u_2}\right| \geq \left|\frac{y_2}{u_1}\right| \tag{3}$$

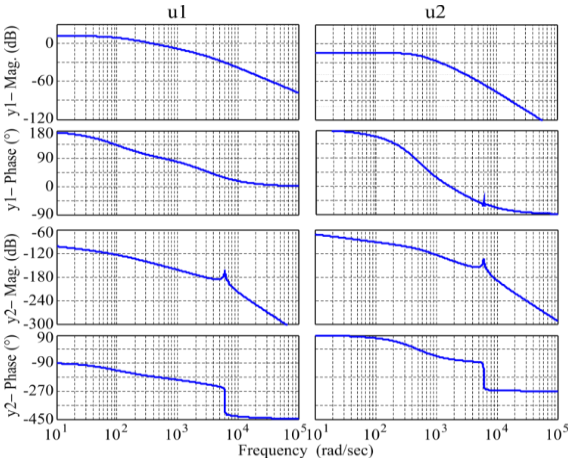


Fig. 6. Bode diagrams of the system in open-loop.

B. Multi-Loop Control Design

To manipulate the system, a structure involving a MIMO (Multiple Inputs Multiple Outputs) controller or two SISO (Single Input Single Output) controllers can be considered. According to [12], the efficiency of the latter depends on the degree of coupling: if the interactions between the loops are weak, a decentralized control can work properly. Different methods allow assessing the degree of coupling [13]. The Mu-Interaction measure has been chosen because of its easy implementation in the 2×2 case [14]. Its maximum value is 0.04, which means interactions are weak in open-loop. Thus, a decentralized control will reach good performances.

Technical specifications are given for the step responses of the two loops:

1) First maximum time less than 0.0025 s for the flow rate loop and less than 0.1 s for the position loop.
2) Zero steady-state error.
3) Overshoot less than 10 %.

An order of magnitude for the specifications of the open-loop frequency responses can be deduced from these technical specifications by making some assumptions [15]:

1) Cutoff frequency greater than 1200 rad/s for the flow rate loop and greater than 30 rad/s for the position loop.
2) Controller with integral action.
3) Phase margin greater than 60 °.

PID controllers reach adequate performances for most applications while having a simple, failure tolerant and easy to understand structure. We will thus design a PID controller for each diagonal input/output pairing.

Several methods allow setting parameters of a PID controller in SISO systems. Some of these methods have been extended for MIMO systems. Reference [16] identifies detuning methods, sequential loop closing methods, iterative or trial-and errors methods and optimization methods.

In this part, we will use the "PID Tuner" of Simulink/MATLAB to adjust the parameters of the two PIDs using the imported AMESim model. Automatic tuning works in the same way whether it is with a Simulink system or with an imported model because it uses the closed-loop linearized system.TheAMESim model has been shown Fig. 5 (b) and the Simulink controllers can be found Fig. 7.

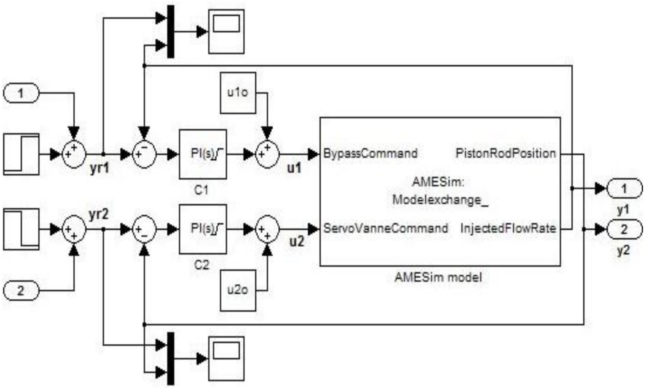


Fig. 7. AMESim model imported into Simulink.

In the first place, one needs to choose the type of the controller: P/I/PI/PD/PID. Simulink then suggests a tuned response but one can adjust some parameters in order to obtain better performances. In the « extended mode » which has been chosen, the bandwidth and the phase margin can be set. The advantage of this method is that the parameters can be adjusted while visualizing directly the step response, disturbance rejection, the open-loop Bode diagram and having access to performances (rise time, settling time, overshoot, peak and margins).The corresponding gain parameters of the controller can also be watched but they cannot be parameterized directly.

Because of the desired cutoff frequency and the open loop phase characteristic shown on Fig. 6, PI controllers will be sufficient to obtain the desired phase margins.Bandwidth and phase margin with safety factors have been directly inputted in the tuner and responses proposed with the linear model seem acceptable. Moreover, to improve step disturbance rejection, the chosen bandwidths have been increased.

If there were no coupling at all in the system, it would be possible to use successively the Tuner for both PI controllers (and even to open one loop when tuning the other) to obtain the desired control. In case of coupling, it could be necessary to reiterate successively the tuning of the two PI controllers to take into account the new set of parameters. Having noted on the one hand the weak coupling and on the other hand that the impact of u_2 on $y_1$is more important than that of u_1 ony_2, tuning of the controller C2 before C1 seems to be an appropriate methodology to get performances in a straightforward way. Doing so, reiterating once this tuning of C2 and C1leads to controller parameters close the initial ones.

When running the simulation with the nonlinear model, control is not effective anymore due to the saturation of the current inputs. Adding an anti-windup mechanism discharges the integrator when the PI block is saturated, preventing the integrator output from growing. Clamping [17] has been selected: it stops the integration when the output of the PI exceeds its bounds and when the integrator output and PI output have the same sign.

With this adjustment, control of the non linear process is effective. The parallel form parameters of the PI controllers used in the following are given in Table II(PI parameters are negative because of the negative static gain of the process).

TABLE II: PI CONTROLLERS PARAMETERS

Loop	P	I	Anti-windup	Saturation limit (mA)
PI C1	-5	-840	clamping	± 30
PI C2	-30000	-450000	clamping	± 120

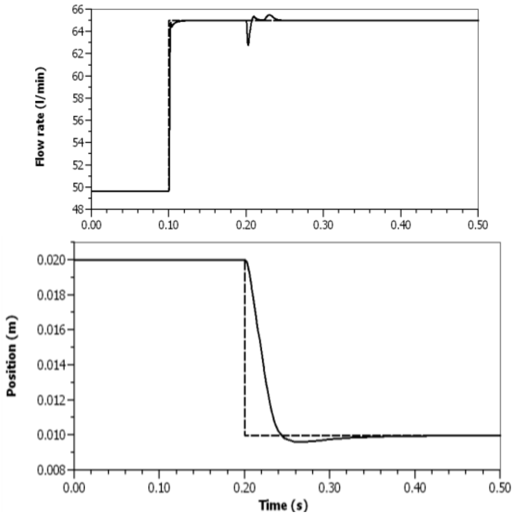


Fig. 8. Results of the system in closed-loop (dashed line: step demand, solid line: response)·

In closed-loop, the temporalresponsesy_1 and y_2 to input steps on flow-setpointy_{r1} and position-setpointy_{r2} are shown in Fig. 8. The first maximum times of y_1 and y_2 are respectively near 0.002 s and 0.06 s. There is no overshoot on y_1 because of the saturations of the command $u_{1.}$ For the same reason, there is an overshoot of only 5% on y_2(instead of 10%). Moreover, there are no steady-state errors. The technical specifications are thus respected.

We can observe that the injected flow rate is perturbed by the moving of the piston rod but this discrepancy of 3.5 % is quickly corrected. Moreover, we have to take into account that the regulation system must stay efficient with a digital electronic control unit which runs at a time step of 10^{-4} s. Discrete PI controllers have then been implemented using a sample time of 10^{-4} s. The discrepancies between temporal responses obtained using continuous and discrete controllers are 0.16%. The controller will thus remain effective after digital implementation.

To conclude, the controller design method is effective, quick to implement and intuitive to adjust the parameters to respect the technical specifications and to obtain the desired responses.

IV. EXCHANGE AND COSIMULATION STRATEGIES WITH CONTINUOUS MODELS

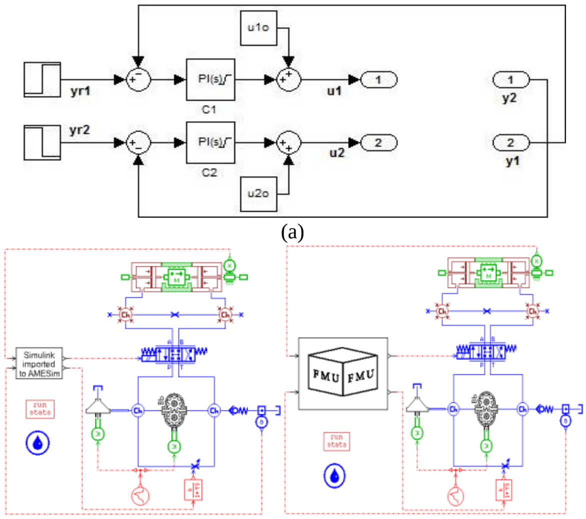
Once controllers have been designed, the goal is to have the fastest possiblesimulation,while keeping up an acceptable level of precision. The practicality of modeling and simulation has to be considered as well.In this part, PI controllers are still continuous. The goal is to find the best coupling method with continuous models analyzing temporal behavior over a 0.12 s time horizon.

Results are compared to a reference obtained using cosimulation with a cosimulation step of 10^{-8} s and both solvers with very restrictive tolerance and maximum step size. In the simulations, solvers parameters have been set to obtain similar relative errors when possible in order to compare the simulation times.

A. Exchange Solutions

1) “AMESimimported into Simulink”: in this mode, Simulink solver simulates the closed-loop system including the AMESim nonlinear model as an S function. The worst and the best results obtained with the different solvers of Simulink are given in Table III. Calculation times are very small and precision does not decrease much compared to the reference solution. Moreover, the model made for the control design can be used as is. Indeed, linearization and PID tuning have been done with the AMESim imported model. This solution is effective, accurate and convenient.

2) “Simulink imported into AMESim”:Another solution is to export the Simulink controllers and loop structure into AMESim. In this case, AMESim simulates the whole closed-loop system using an adapted variable step solver that is automatically defined depending on the system. Calculation time and relative error are shown in Table III. To complete this export, some minor changes on models have to be made in Simulink and AMESim as shown in Fig. 9 (a) and Fig. 9 (b). We also have to use Simulink Coder (formerly RTW) to generate binary code corresponding to the model. This solution is thus efficient and accurate but makes it necessary to modify the models.



(a)

(b)(c)

Fig. 9. Simulink (a) imported into AMESim (b) and through FMI (c).

3) “Exchange with FMI”:it is also possible to use the FMI standard to import Simulink models into AMESim. Modelon has developed a Simulink additive called “FMI Toolbox for MATLAB” which allows among others things to export a Simulink model into a FMU (Functional Mock-up Unit). One needs to use code generation with a “FMU target” to obtain the FMU. Then, it can be imported into AMESim which will convert the FMU into a full-featured AMESim submodel that can be connected to other submodels as we can see in Fig. 9 (c). With this solution, performances similar to the previous ones are reached as can be seen in Table III. However, export into FMU and subsequent import into the AMESim environment takes more time than the previous solution.To conclude, there is no real advantage in using FMI in this case. The FMU model can be exported to other software but this feature is not used here.

B. Cosimulation Solutions

The second class of solutions is cosimulation. In this context, each piece of software solves its subpart of the system. One of the software is the master of the cosimulation and the process of exchanging information is entirely controlled by its solver.

One of the most important parameter is the cosimulation step: it represents the rate of exchange of data. The exchange will cause numerical discontinuities in the behavior of the model as if signals were sampled. Thus, phenomena encountered with sampling are the same as those we can have in cosimulation.By exchanging only input and output variables at a certain sample time, there is a loss on information. In order to reduce it, we have to respect Shannon’s theorem which gives an upper limit of the sample step h according the higher frequency of the signals generated in the system [15]:

$$h \leq \frac{1}{2 \times f_{max}} \tag{4}$$

In fact, an exchange frequency at least ten times higher than the natural frequencies of the system is often chosen in order to observe the fastest system dynamics. The highest natural frequency of the system reaches 6090 rad/s (Table I), which means that the cosimulation rate has to be smaller than 10^{-4} s.

Attention also has to be paid to the fact that the sampling generated by the cosimulation doesn’t cause instability. In fact, a delay in the loop will reduce the phase margin and can make the system instable. The delay margin will quantify the maximal time delay to delete the phase margin. Thus, cosimulation step h has to be lower than the delay margin of the system (with $\Delta\varphi$ the phase margin andω_c the cutoff pulsation):

$$h \leq \tau_m = \frac{\Delta\varphi}{\omega_c} \tag{5}$$

In the multivariable system considered, the lowest delay margin is 1.4×10^{-3} s.Taking into account these two limits, the cosimulation step must be smaller than 10^{-4} s. Moreover, it has to be the smallest possible because simulation time increases tremendously with the exchange frequency. This is easily understandable: the integration stepcannot be bigger than the cosimulation step, meaning that the simulation will take integration steps equal to the cosimulation step during steady state behaviors, slowing the computation.

In this situation in which the physical system and its controller are continuous, cosimulation cannot be handled by AMESim (a current restriction imposed to Simulink is to work with a fixed step). This is why cosimulation handled by Simulink has been chosen in this study.Variable-step solvers have been tried and Ode23tb has been chosen for the following tests because of its good performances, and the same solvers parameters as those taken in exchange mode have been chosen. A number of cosimulation steps have been tried in order to determine which one gives accuracy results with a reasonable CPU time. Relative error and calculation time for those tests are shown Fig. 10.

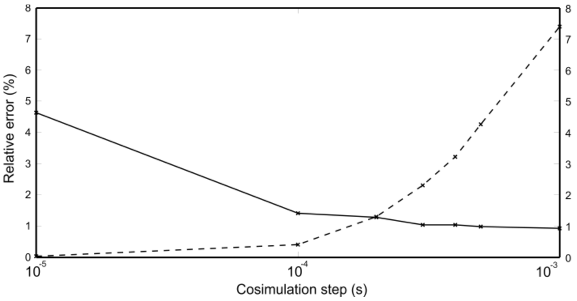


Fig. 10. CPU time (solid line) and relative error (dashed line) in cosimulation.

A cosimulation step of 10^{-4} s as calculated above gives a relative error of 0.4 % and a simulation time of 1.40 s.Taking larger cosimulation steps leads to faster simulations but accuracy is highly impacted andmaylead tothe presence of oscillations which are not representative of the physical behavior of the system.

The setup of this method is quite easy. However, looking at Table III, we can see that cosimulation is more time consuming and precision during transient phases is lower than with the previous methods. Cosimulation can however be beneficial in case of a system with several dynamics which can be decoupled, for instance if we consider the same system with thermal phenomena. Moreover, cosimulation can be necessary if a particular solver on a specific platform is needed, for instance one which deals with large non linearities and discontinuities.

TABLE III: SIMULATION TIMES FOR CONTINUOUS TIME MODELS

Solution	CPU Time (s)	Relative error (%)
AMESim imported into Simulink	[0.80,4.50]	[0.045,0.2]
Simulink imported into AMESim	1.10	0.31
FMU imported into AMESim	1.15	0.31
Cosimulation handled by Simulink	1.40	0.40
Cosimulation handled by AMESim		

Variable cosimulation step could highly increase the CPU time [18], [19]. A tested solution that can accelerate the simulation consists in putting the AMESim model in a triggered Simulink subsystem. The trigger (and thus the cosimulation step with inherited sample time) can be controlled to get the required precision for each phase of the simulation. If inputs of the system are known in advance, tolerant and restrictive cosimulation steps can be imposed for respectively steady state and transient phases. This solution works with simple models but is not robust and fails for our case study.

We have seen that modeling the hydraulic system from scratch in Simulink is tedious due to the complexity of the system. However, it is possible to implement the designed control with AMESim blocks, avoiding using two tools in simulation, which is more convenient and better in terms of license costs. Available PI with clamping blocks have thus been added in AMESim model but whatever the solver parameters, relative error was larger than in the other simulation strategies.

To conclude, when no digital electronic unit is taken into account, the models in both tools are continuous. Exchange solutions seem then better than cosimulation solutions. Between exchange solutions, import of an AMESim model in Simulink seems to be the easiest solution to implement while reaching good performances.

V. EXCHANGE AND COSIMULATION STRATEGIESTAKING INTO ACCOUNT THE DIGITAL ELECTRONIC UNIT

In order to simulate the behavior of the real electronic (and thus discrete) unit, discrete PIs can be implemented in Simulink as we have seen before. A new reference was made using the same solvers parameters as before.

A. *Exchange Solutions*

As we can see in Table IV, CPU times of the exchange solutions with discrete controllers are longer than those previously obtained in Table III. This can be explained by the fact that the integration step has to be smaller than the discrete step used by the digital controllers. It is thus more time consuming during steady behaviors. The AMESim solver seems less adapted to discrete blocks than the Simulink solver, which explains the bigger discrepancy of CPU time between these solutions.

TABLE IV: SIMULATION TIMES WITH DIGITAL CONTROL UNIT

Solution	CPU Time (s)	Relative error (%)
AMESim imported into Simulink	[1.9,2.4]	[0.15,1.0]
Simulink imported into AMESim	3.4	0.23
FMU imported into AMESim	3.5	0.23
Cosimulation handled by Simulink	2.0	0.15
Cosimulation handled by AMESim	1.85	0.15

B. *Cosimulation Solutions*

Cosimulation is very suitable for discrete controllers implemented in Simulink controlling AMESim continuous model because it can be made consistent with the physical behavior. A cosimulation step of 10^{-4} shas obviously been used.

1) "Cosimulation handled by Simulink": As the Simulink model is purely discrete, the fixed-step discrete solver of Simulink has been used. CPU times and relative errors are given in Table IV. Compared with Table III, CPU time has slightly increased in cosimulation with the use of discrete controllers. Nevertheless, unlike in the previous situation as reported in Table III, cosimulation reaches same CPU times as those using exchange solutions. Indeed, exchange solutions have to take integration steps at least equal to the discrete sampling time. They then present the same drawbacks as cosimulation which explains the low discrepancy between CPU times in this case.

2) "Cosimulation handled by AMESim":Another recent possibility is to use cosimulation with AMESim being the master. As seen in Table IV, results are slightly better in terms of CPU time with Simulink being the master. This seems strange because the two solvers used are the same for the two cosimulations. To analyze these differences, the number of function evaluations and CPU time per function evaluation have been compared. The number of function evaluations is quite similar but the cosimulation handled by Simulink is slightly more time consuming per

evaluation. Moreover, initialization seems faster when AMESim handles the cosimulation.

To conclude, when taking into account the digital electronic unit, cosimulation reaches equivalent or better CPU times than exchanges solutions. Accuracy of the different methods are similar. Either simulation strategies can then be used.

VI. Conclusions

Based on coupling methods between AMESim and Simulink,the approach described in this paper allows modeling the fuel system, designing its controller and simulating the whole closed-loop system. Without tedious adaptation,an AMESim model can be imported in Simulink, allowing a sophisticated analysis, a fast tuning of controller and efficient simulations.

After assessing the couplings of the multivariable system, a decentralized control approach has been chosen: the system has been handled as two monovariable systems, which has allowed easy implementation and tuning of the control law parameters. This strategy respects the technical specifications but is not optimal because of the neglect of the couplings.Control performances can still be improved: for example, adding a pre-compensator at the input of the plant can transform the open-loop characteristics to reduce couplings.

Different methods for coupling simulation tools have been benchmarked.For coupling continuous systems, a model exchange strategy will be preferred because of the accuracy of the results obtained with reasonable simulation times. However, cosimulation seems to be perfectly adapted for the association of a discrete time controller and a continuous system. Considering the FMI, even if exchange and cosimulation with FMI can raise some issues for the moment, this standard is still in process and will probably be increasingly used because of its capacity to replace the other point-to-point solutions.

References

[1] J. Marques, “Plate-forme d’aide à l’écho-conception de systèmes multiphysiques,” Ph.D. Dissertation, University of Orleans, pp. 17-48, 2010.

[2] H. Kadima, “Méthodes et outils d’ingénierie des systèmes mécatroniques fiables,”EISTI, pp. 6-15, 2010.

[3] K. K. Ahn and D.Q. Truong, “Online tuning fuzzy PID controller using robust extended Kalmanfilter,” *Journal of Process Control*, 2009.

[4] M. Xu, J. Ni, and G. Chen, “Co-simulation of energy regulation based variable-speed electrohydraulic drive,” *Procedia Engineering*, 2011.

[5] Y. Shtessel, M. Taleb, and F. Plestan, “A novel adaptative-gain supertwisting sliding mode controller: methodology and application,” *Automatica*, 2012.

[6] *Simulink Interface Rev12*, 2013.

[7] R. Koseel, W. Tegethoff, M. Bodmann and N. Lemke, “Simulation of complex systems using Modelica and tool coupling,” in *Proc. 5th International Modelica Conference*, 2006.

[8] T. Roudier and P. Leblanc, “Co-Simulation de systèmes multi-physique piloté par Rhapsody et CosiMate,”2011.

[9] Z. B. Touini, M.B. Gaid *et al.*, “From physical modeling to real-time simulation: feed back on the use of Modelica in the engine control development toolchain,” in *Proc. 8th International Modelica Conference*, 2011.

[10] Association Modelica.[Online]. Available: https://www.fmi-standard.org/

[11] J. Bastian, C. Clauss, S. Wolf, and P. Schneider, “Master for co-simulation using FMI,”in *Proc. 8th International Modelica Conference*, 2011.

[12] P. Albertos and A. Sala, *Multivariable Control Systems*, Springer, pp.125-136, 2004.

[13] J.M. Maciejowski, *Multivariable Feedback Design*, Adison-Wesley, ch. 7, 1989.

[14] D. Vaes, “Optimal static decoupling for multivariable control design,” Ph.D. Dissertation, Catholic University of Leuven, Belgium, pp. 110-111, 2005.

[15] E. Godoy, “Régulation industrielle, ”*Dunod*, 2007.

[16] H. P. Huang, J.C. Jeng, C.H. Chiang, and W. Pan, “A direct method for multi-loop PI/PID controller design,”*Journal of Process Control*, 2003.

[17] A. Bemporad, “Anti-wind-up techniques,” *Automatic Control*, 2011.

[18] M. Arnold, C. Clauss, and T. Schierz, “Error analysis and error estimates for co-simulation in FMI model exchange and co-simulation 2.0,” *The Archive of Mechanical Engineering*, 2013.

[19] T. Schierz, M. Arnold, and C. Clauss, “Co-Simulation with communication step size control in an FMI compatible master algorithm,” in *Proc. 9th International Modelica Conference*, 2012.

Christophe Le Brun was born in 1988 in Paris, France. He received the Diploma in engineering from Arts & Métiers ParisTech, France and MS in Advanced Systems and Robotics from Pierre and Marie Curie University, France in 2011.He is currently in Ph.D. (second year) in Automatic Control Department at Supelec in cooperation with the DynFluid Laboratory of Arts & Métiers ParisTech and the System Divisions of SNECMA (SAFRAN group). His research interests include multivariable control and coupling methods.

Emmanuel Godoy graduated from from Supélec, Gif sur Yvette, France in 1984 and HdR (research habilitation) in Automatic control from Paris-Sud University, Orsay, France in 2008. He is currently a professor in the Automatic Control Department of Supélec. His area research is about modelling and robust control of mechatronic systems and common control approaches for industrial problems.

Dominique Beauvois graduated from Supélec, Gif sur Yvette, France in 1977. He is currently a professor in the Automatic Control Department of Supélec. His current interests include estimation, identification techniques and robust control approaches for industrial problems.

Gwenolé Le Pache was born in 1987 in Brest, France. He receivedthe Diploma in engineering from Arts & Métiers ParisTech, France and a MS in mechatronics from Loughborough University, UK in 2010.He is currently working as a method &tools engineer in the System Division of SNECMA (SAFRAN group), dealing with large-scale modeling methods and tools.

Ricardo Noguera was born in 1952 in Palmira, Venezuela. He receivedin 1976 the Diploma in Mechanical Engineering from the University of Carabobo, Venezuela. He receivedin 1978 the Diploma of Advanced Studies – DEA in Mechanical Engineering from ENSAM, France. He was awarded Ph.D. with “Right Honorable” from Pierre and Marie Curie University, France in 1987.His research activities in the Laboratory of Fluid Dynamics include: analysis of internal flows, design and performance of turbomachinery, cavitation and pumping two-phasemixtures.