

Fast simulation model for control law design and benchmark of high aspect ratio flexible UAVs

Romain. Jan (ISAE-SUPAERO, University of Toulouse, 31400 Toulouse, France);* Jean-Philippe Condomines (ENAC,31400 Toulouse, France),Jean-Marc Moschetta(ISAE-SUPAERO)

ABSTRACT

A low/middle range fidelity model is highlighted in this work through the development of a conservative extension in order to assess a significant part of the UAVs community's current issues. Such as flutter control, gust alleviation, trajectory tracking, energy harvesting strategies, parametric studies... One aims at providing an efficient tool for the quick design of high aspect ratio UAVs coupled with advanced control laws. The work recalls the theoretical background used on the model, the methodology applied to enhanced it, and 3 illustrative examples.

1 INTRODUCTION

Nowadays' the trend regarding UAVs is the improvement of efficiency, endurance, and impact on the environment leading to lighter and flexible aircraft, rising new problems regarding its sensitivity to the surrounding environment.

The last works regarding UAVs rise the need for modular and fast simulation model to capture sophisticated flight dynamics as well as the structure behaviours both coupled with advanced control strategies. Such a tool could assess the different issues risen by UAVs such as flutter damping ([1], [2],[3]), energy harvesting strategies ([4], [5], [6]), flap inversion behaviour due to structural deformation ([7]), flight enveloppee extension ([8]), gust alleviation ([9],[10])

There exists different tools to assess those problems adapted to high aspect ratio UAVs such as ASWING [11], UM/NAST [12], CA^2LM [13] and others used in [14]. Despite the recency of some of them, after an overview of the theoretical background used, the authors choice converged to ASWING for 3 main reasons derived below:

structural part:

Some of them do not consider all the 6 degrees of freedom, inertial coupling, and local damping. Moreover, coupling between local beam variables is not considered (cross-sectional bending/twisting). Few models assume small beam deflections and a linear beam behavior. Elastic, tension, and gravity's center's positions can not be specified. Others use a rectangular beam assumption for bending and stiffness matrices computation.

aerodynamic part:

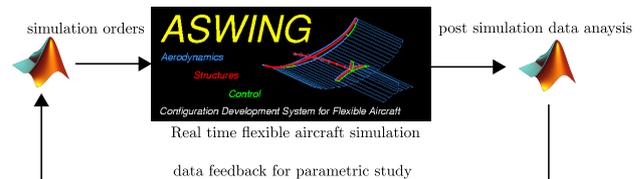


Figure 1: ASWING working with MATLAB

Propeller jet-induced velocities are neglected in most of cases. For some compressibility effects are not taken into account as well as ground effects. Shed vortices are approximated by a temporal Theodorsen function. Other models do not present a post-stall behavior capture.

applied forces:

Added mass effects, unsteady effects on propeller blades, joints, and struts loads are not usually considered as well. Joints degrees of freedom restriction, and concentrated efforts neither.

Furthermore, as most of those low order models are designed on MATLAB, calling external libraries costs a lot of time especially in temporal loops. Memory is dynamically allocated and freed leading to slower simulation. As a compiled program ASWING is very fast and works in real-time with nowadays computers. With an improved temporal solver and a compiled version on a dedicated device, it could show interesting performances.

However, MATLAB remains a very efficient and productive tool for analysis and post-simulation data treatment. Consequently, one's objective is to present an extension of ASWING that the user could run through MATLAB as an "opened" black box as shown in figure 1 The use of a fast compiled simulation software with MATLAB would allow large parametric studies of flexible UAVs for a small computational time cost.

Unfortunately, ASWING provides a very limited control law toolbox which limits its use in the flight control community highlighting the main contribution of this work.

2 PROBLEM STATEMENT

In ASWING [15],[16] the dynamic of a given aircraft is described by a non linear system with x as a states space vector:

*Email address(es): romain.jan@isae-supaero.fr

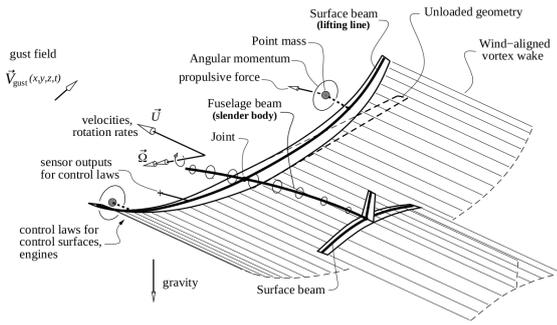


Figure 2: ASWING configuration representation [15]

$x =$

$(\vec{r}_i, \theta_i, M_i, u_i, \omega_i, \dot{r}_i, \dot{\omega}_i)$: **beam node variables** ($18N_B$)

r_J, θ_J, M_J, F_J ; **joints variables** ($12N_J$)

$A_1, \dots, A_K, \dot{A}_1, \dots, \dot{A}_K$: **circulation modes** ($2K$)

$\vec{R}_E, \vec{R}_{E'}, \vec{\Theta}, \vec{\Theta}', \vec{U}, \vec{U}', \vec{\Omega}, \vec{\Omega}'$: **global position and rates** (32)

\vec{a}_0, \vec{a}_0' : **absolute accelerations variables** (6)

$\delta F_1, \dots, \delta F_N, \delta E_1, \delta E_N$: **flaps and engine level** (32)

)^T

y is the output variables measured at each sensor location which are x dependant. The feedback-able variables are the local freestream velocity, angle of attack and sideslip, local positions, angles, velocities, rotations rates and accelerations, internal beam loads and circulations.

u denotes the forcing vector defined by :

$u =$

(y^*) : **desired output behaviour**

$\delta^* F_1, \dots, \delta^* F_N, \delta^* E_1, \delta^* E_N$: **desired flaps, engine levels**

$V_{wind}(\vec{x}, z, y, t)$: **turbulence velocity model**

)^T

The 6 degrees of freedom of every beam node are governed by the non-linear Euler Bernoulli beam equations in its multivariable differential form. By denoting $q = (\vec{r}_i, \vec{\theta}_i)_i \in [1, N_B]$.

$$[M(x)]\ddot{q} + [C(x)]\dot{q} + [K(x)]q = 0 \quad (1)$$

where $[M(x)]$, $[C(x)]$ and $[K(x)]$ are non linear mass, damping and stiffness matrices. They take into account the effect of structural damping, loads and rates, external distributed and concentrated loads. Distributed loads are lift and drag, added mass effect, inertial and gravity loads integrated along the beam. Concentrated loads recover the effect of engine thrust and drag, point mass, struts, and joints loads. Joints variables are introduced to connect and constraint beams. Circulation modes are used in an extended unsteady

lifting line theory taking into account compressibility effect and velocity influence of wing thickness; shed and steady vortices of every lifting beam and propeller jet stream. Circulation modes are constraints in a flow tangency condition with post-stall modelization. Those circulation modes are used to compute lift and drag. Distributed lift is computed with the unsteady form of Kutta-Joukowski theorem for surface beams, and with slender body theory for fuselage beams. Distributed drag sums up the friction and pressure drag and post-stall contributions. For both lift and drag, the knowledge of tangent and orthogonal relative stream velocity is necessary which is a function of infinite freestream, local beam node positions and rates, turbulence, and induced velocities. The global position, rates, and absolute accelerations variables are governed by the "rigid" kinematic and trajectory equations known as :

$$\frac{d\vec{R}_E}{dt} - \vec{T}_E \vec{U} = 0 \quad (2)$$

$$\frac{d\vec{\Theta}}{dt} - \vec{C}_E \vec{\Omega} = 0 \quad (3)$$

$$\frac{d\vec{U}}{dt} + \vec{\Omega} \times \vec{U} - \vec{a}_o = 0 \quad (4)$$

$$\frac{d\vec{\Omega}}{dt} - \vec{a}_o = 0 \quad (5)$$

where \vec{T}_E and \vec{C}_E are global position and orientation dependant transformation matrices. Moreover, absolute accelerations can be constrained for a free flight or anchored configuration.

The flaps and engine-level variables are governed by a control law equation introduced later in the document.

In the end, the behavior of the state space vector x is governed by the non linear system :

$$(\Sigma) : \begin{cases} \dot{x} = f(x, u) \\ y = h(x, u) \end{cases} \quad (6)$$

Depending on the number of beams and their associated mesh, the total number of states variables can rise very fast and reach more than 10000. Using this discrete model as an analytical tool for control law design would be unadapted. Therefore ASWING provides a reduced-order model (ROM) generator function leading to the linear system

$$(ROM) : \begin{cases} \dot{z} = \Lambda z + Bu \\ y = Cz + Du \end{cases} \quad (7)$$

where Λ is a diagonal matrix containing the aircraft modes and z is an alternative state vector equivalent to x defined as:

$$z = Vx \quad (8)$$

Control laws	
Linear	Non Linear
PID	Gain Scheduling
LQG	Slidding mode control
H_∞/H_2	Control-Lyapunov functions
μ -analysis	Backstepping
Guardian mapping	Non linear damping
Fractional order Controller	Feedback Linearization

Table 1: Examples of implementable control laws

In practice, the user can select slower and fast undamped or unstable modes to drastically reduced the dimension of z. Moreover, he can select different outputs to study the small perturbations' effects of each mode. The system (7) is then a useful intel for control laws design.

Unfortunately, ASWING comes with a limited control law known as outputs proportional feedback :

$$\delta = \delta_{ref} + K_y(y - y_{ref}) \quad (9)$$

where K_y is a bi-scheduled matrix. One must note that integral errors are part of the y vector allowing the user to implement PID controllers. However the main problems of this form are the assumptions of perfect knowledge of the outputs and direct impact of actuators (no noise, no sensors/actuators dynamics or saturations), and the lack of the controller's internal dynamic. Consequently, advanced control synthesis and benchmark are limited leading to the contribution of this article.

3 ASWING ADVANCED CONTROL LAWS

One is seeking to implement a multi-inputs multi-outputs (MIMO) non-linear control law as depicted on figure 3. To fulfill this objective, one must introduce a new state vector to consider its internal dynamic. Thus for a set of control states variables vector x_c the control law takes the form:

$$(C) : \begin{cases} \dot{x}_c = f_c(x_c, y, y_{ref}, \delta_{ref}) = f(x_c, u_c) \\ \delta = h_c(x_c, u_c) \end{cases} \quad (10)$$

10 recovers many control law forms used in MIMO non linear and linear theory sum up in table 1. Moreover, this form remains conservative, meaning that the original PID can still be implemented. Thus the next lines aim at giving the critical steps to apply so that such control laws could be used on ASWING.

3.1 Modal analysis and time marching

For the next lines one will prefer the residual form of 6 written as :

$$r(x, \dot{x}, u) = \dot{x} - f(x, u) \quad (11)$$

The state vector temporal evolution can be solved by using a multivariable Newton alorithm depicted in [17] and [18] as :

$$x_n^{i+1} = x_n^i - \left[\frac{\partial r}{\partial x} + k_0 \frac{\partial r}{\partial \dot{x}} \right]_i^{-1} r^i \quad (12)$$

where i is the Newton iteration index and n is the time index such that $t = nTe$. Again for the next lines one will prefer the shorter notation of the matrix:

$$\left[\frac{\partial r}{\partial x, \dot{x}} \right]_i = \left[\frac{\partial r}{\partial x} + k_0 \frac{\partial r}{\partial \dot{x}} \right]_i \quad (13)$$

To solve 12 a Gaussian block elimination [17] is used to invert the matrix (13). Such matrix representation appears suitable for different types of analysis. In fact by forcing k_0 to zero, one force 12 to the steady case and by inspection of 13 eigenvalue one recovers the modal response of the aircraft for a given steady flight condition. Secondly, it provides an accurate time-marching behavior for a relatively small time step choice. To embed 10 in the solver 12 one must split the state vector x such as :

$$x^T = (x_Q^T, x_V^T, x_P^T, x_D^T) \quad (14)$$

with:

$$\begin{aligned} x_Q^T &= (\vec{r}_i \quad \vec{\theta}_i \quad \vec{M}_i \quad \vec{F}_i \quad \vec{u}_i \quad \vec{\omega}_i)_{i \in [1, N_B]} \\ x_V^T &= (\Delta \vec{r}_J \quad \Delta \vec{\theta}_J \quad \vec{M}_J \quad \vec{F}_J \quad A_1 \quad A_2 \\ &\dots \quad A_K \quad e)_{J \in [1, N_J]} \\ x_P^T &= (\vec{R}_E \quad \vec{\Theta} \quad \vec{U} \quad \vec{\Omega} \quad \vec{a}_o \quad \vec{\alpha}_o) \\ x_D^T &= (\delta_{F1}, \dots, \delta_{FN}, \delta_{E1}, \dots, \delta_{EN})^T \end{aligned}$$

where evey variables from x_Q, x_V, x_P, x_D have been described in section 2. At the light of the state vector form, 12 is equivalent to invert the matrix:

$$\left[\frac{\partial r}{\partial x, \dot{x}} \right]_i^{-1} = \begin{bmatrix} \left[\frac{\partial r_Q}{\partial Q, \dot{Q}} \right]_i & \left[\frac{\partial r_Q}{\partial V, \dot{V}} \right]_i & \left[\frac{\partial r_Q}{\partial P, \dot{P}} \right]_i & \left[\frac{\partial r_Q}{\partial D, \dot{D}} \right]_i \\ \left[\frac{\partial r_V}{\partial Q, \dot{Q}} \right]_i & \left[\frac{\partial r_V}{\partial V, \dot{V}} \right]_i & \left[\frac{\partial r_V}{\partial P, \dot{P}} \right]_i & \left[\frac{\partial r_V}{\partial D, \dot{D}} \right]_i \\ \left[\frac{\partial r_P}{\partial Q, \dot{Q}} \right]_i & \left[\frac{\partial r_P}{\partial V, \dot{V}} \right]_i & \left[\frac{\partial r_P}{\partial P, \dot{P}} \right]_i & \left[\frac{\partial r_P}{\partial D, \dot{D}} \right]_i \\ \left[\frac{\partial r_D}{\partial Q, \dot{Q}} \right]_i & \left[\frac{\partial r_D}{\partial V, \dot{V}} \right]_i & \left[\frac{\partial r_D}{\partial P, \dot{P}} \right]_i & \left[\frac{\partial r_D}{\partial D, \dot{D}} \right]_i \end{bmatrix}^{-1} \quad (15)$$

As mentioned before a Gaussian block elimination of order 4 is used to solve 15. Note that, the jacobian matrix $\left[\frac{\partial r_Q}{\partial Q, \dot{Q}} \right]_i$ follows a bi-tridiagonal block pattern. Thus the order of definition of the state vector in Eq. (14) is relevant because the first step of Gaussian Block elimination will not affect the upper left matrix block. For the rest of the inversion, one uses

http://www.imavs.org/

LU factorization to invert the diagonal matrix blocks. Consequently one must not change the order of definition of the state vector, otherwise, a huge slow down effect during simulation will be witnessed.

3.2 Extension to a control law with an internal dynamic

As one rises the state vector x with x_c , the problem becomes:

$$(16) \quad \begin{bmatrix} \left[\frac{\partial r}{\partial x, \dot{x}} \right]_i & \left[\frac{\partial r}{\partial x_c, \dot{x}_c} \right]_i & \vdots & R_x \\ \left[\frac{\partial r_c}{\partial x, \dot{x}} \right]_i & \left[\frac{\partial r_c}{\partial x_c, \dot{x}_c} \right]_i & \vdots & R_{x_c} \end{bmatrix}$$

which leads again to the seek of:

$$(17) \quad \left[\frac{\partial r}{\partial x, \dot{x}} \right]_i^{-1} = \begin{bmatrix} \left[\frac{\partial r}{\partial x, \dot{x}} \right]_i & \left[\frac{\partial r}{\partial x_c, \dot{x}_c} \right]_i \\ \left[\frac{\partial r_c}{\partial x, \dot{x}} \right]_i & \left[\frac{\partial r_c}{\partial x_c, \dot{x}_c} \right]_i \end{bmatrix}^{-1}$$

As one was seeking to modify as less as possible ASWING code and as it implements a 4th order Gaussian Block Elimination and not a generic one, it is consequently not possible to just add the Jacobian matrix associated to X_c and R_{X_c} (bottom right block in (17)). Consequently, the code has been modified so that the new set of control law x_c is a part of the x_D state vector.

Note that it's not necessary to give all the jacobian matrices. Considering the definition of the control law from Eq. (10) and by having in mind that $\frac{\partial y}{\partial x}$ is automatically provided by ASWING code [16], the user only needs to provide $f_c(x_c, u, y)$, $h_c(x_c, u, y)$, $\left[\frac{\partial f_c}{\partial X_c} \right]$, $\left[\frac{\partial f_c}{\partial y} \right]$, $\left[\frac{\partial h_c}{\partial X_c} \right]$ and $\left[\frac{\partial h_c}{\partial y} \right]$. With those intels, one can recover both residual forms of the controller defined in Eq. (10)

$$(18) \quad \begin{aligned} r_D &= \delta - h_c(x_c, u, y) \\ r_{X_c} &= \dot{x}_c - f_c(x_c, u, y) \end{aligned}$$

and its associated Jacobian matrices:

$$\begin{aligned} & \left[\left[\frac{\partial r_{X_c}}{\partial Q, \dot{Q}} \right]_i \quad \left[\frac{\partial r_{X_c}}{\partial V, \dot{V}} \right]_i \quad \left[\frac{\partial r_{X_c}}{\partial P, \dot{P}} \right]_i \quad \left[\frac{\partial r_{X_c}}{\partial D, \dot{D}} \right]_i \right] = \\ & - \frac{\partial y}{\partial x} \frac{\partial f_c}{\partial y} \\ & \left[\frac{\partial r_{X_c}}{\partial X_c, \dot{X}_c} \right]_i = \left[k_0 I - \frac{\partial f_c}{\partial X_c} \right] \\ & \left[\left[\frac{\partial r_D}{\partial Q, \dot{Q}} \right]_i \quad \left[\frac{\partial r_D}{\partial V, \dot{V}} \right]_i \quad \left[\frac{\partial r_D}{\partial P, \dot{P}} \right]_i \quad \left[\frac{\partial r_D}{\partial D, \dot{D}} \right]_i \right] = \\ & - \left[\frac{\partial y}{\partial x} \frac{\partial h_c}{\partial y} \right] + [0 \quad 0 \quad 0 \quad I_{n_D}] \end{aligned}$$

$$(19) \quad \begin{aligned} & \left[\frac{\partial r_D}{\partial X_c, \dot{X}_c} \right]_i = \left[\frac{\partial h_c}{\partial X_c} \right] \\ & \left[\left[\frac{\partial r_Q}{\partial X_c, \dot{X}_c} \right]_i \quad \left[\frac{\partial r_V}{\partial X_c, \dot{X}_c} \right]_i \quad \left[\frac{\partial r_P}{\partial X_c, \dot{X}_c} \right]_i \right]^T = \\ & [0 \quad 0 \quad 0]^T \end{aligned}$$

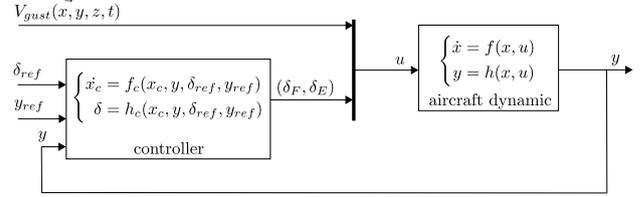


Figure 3: Proposed closed loop in Aswing including advanced control laws

3.3 Linear Pattern

Equations (19) and (10) are the milestone of the ASWING control laws extension but they still need to be implemented in the source code. Therefore one proposes a linear pattern depicted in figure 4. The user only needs to provide every A,B,C,D matrices of each block in a MATLAB format. If a block is not used, it will be automatically removed from the program. Regarding the pattern itself, one can implements sensors and actuators dynamics and saturations, a full MIMO linear controller, and an anti-windup system. Moreover, every block can be bi-scheduled by every measurable output. Furthermore, noise can be added to the simulation for temporal robustness studies in a stochastic environment. This pattern is motivated by the fact that UAVs are using small embedded sensors and actuators leading to a noisy environment, saturations delays, etc. The authors consequently thought it would be relevant to consider them directly in a numerical tool. Let $x_c = (x_A, x_S, x_{AW}, x_K)^T$ be the control states vector, following Eq.(18), In its full configuration (every block used) one has :

$$(20) \quad h(x_c, y, u) = \text{sat}(C_A x_A)$$

where $\text{sat}(C_A x_A)$ $\text{sat}(C_S x_S)$ are the saturation function associated to actuators and sensors

Moreover, one has

$$(21) \quad \frac{\partial h(x_c, y, u)}{\partial x_c} = \begin{pmatrix} \frac{\partial \text{sat}(C_A x_A)}{\partial x_A} & 0 & 0 & 0 \end{pmatrix}$$

The jacobian matrix becomes :

$$(22) \quad \frac{\partial h(x_c, u, y)}{\partial y} = 0_{n_\delta, n_y}$$

At the light of figure 4 f_c expression comes:

$$f(x_c, y, u) = \begin{pmatrix} A_a x_A + B_A C_K x_K + B_A D_{K,Q} \text{sat}(C_S x_S) \\ A_S x_S + B_S (y + w_y) \\ A_{AW} x_{AW} + B_{AW} (C_A x_A - \text{sat}(C_A x_A)) \\ A_{KxK} + B_{K,Q} \text{sat}(C_S x_S) - B_{K,Q} Q^* \\ + B_A \delta^* \\ \dots \\ \dots \\ B_{K,AW} (C_{AW} x_{AW} + D_{AW} (C_A x_A - \text{sat}(C_A x_A))) \end{pmatrix} \quad (23)$$

With its associated jacobians:

$$\frac{\partial f}{\partial x_c} = \begin{pmatrix} A_a & B_A D_{K,Q} \frac{\partial \text{sat}(C_S x_S)}{\partial x_S} \\ 0_{n_s, n_A} & A_S \\ B_{AW} C_A - B_{AW} \frac{\partial \text{sat}(C_A x_A)}{\partial x_A} & 0_{n_{AW}, n_S} \\ D_{AW} \left(C_A - \frac{\partial \text{sat}(C_A x_A)}{\partial x_A} \right) & B_{K,Q} \frac{\partial \text{sat}(C_S x_S)}{\partial x_S} \\ 0_{n_A, n_{AW}} & B_A C_K \\ 0_{n_S, n_{AW}} & 0_{n_S, n_K} \\ A_{AW} & 0_{n_{AW}, n_K} \\ B_{K,AW} C_{AW} & A_K \end{pmatrix} \quad (24)$$

And

$$\frac{\partial f}{\partial y} = \begin{pmatrix} 0_{n_A, n_y} \\ B_S \\ 0_{n_{AW}, n_y} \\ 0_{n_K, n_y} \end{pmatrix} \quad (25)$$

Adding 23, 24, 25, 20, 21 and 22 to 19 recovers 10 for the linear pattern case.

Interpolation methods:

Interpolation methods have been implemented to set up the A,B,C and D matrices of each block regarding their discrete scheduled values. The first method is the nearest neighbor, the second a bilinear interpolation and the third is a 2D polynomial interpolation. Interpolation methods are often used as gain scheduling methods, however, a stability study must be made to ensure that the flight envelope is not too coarse to threaten the stability of the overall closed-loop system during working point switches. ASWING modification does not provide such an analytical tool. The user will consequently make sure that the aircraft flight envelope has enough points to avoid such stability problems.

4 EXAMPLES

The authors must notify, that the change in ASWING code lets the previous one unchanged meaning that the examples presented in [11] are still valid and usable. The lecturer must see this work as a conservative extension.

Span	4.4m	Time range	36h
Weight	14kg	Range	3000km
Cruise speed	24m/s	nominal altitude	100m

Table 2: Mermoz main specs

4.1 open loop control behavior with actuator dynamic:

The first example aims at showing the effect of an added dynamic and saturations to an aircraft actuator such as an engine. The aircraft used in the next examples is the Mermoz UAV hydrogen prototype [4] shown on figure 5 whose main characteristic are depicted on table 2. The engine limits are arbitrarily set to 0 and 7N for a non negative thrust behavior and limited engine. A first order linear system with a time response of 3s is used to recover its dynamic. To ensure that the simulation would not crashed, the aircraft is cantilevered to the ground. The figure 6 shows the benefit of added pattern to take into account actuators dynamic and saturations. The black line accounts for the previous way, and the blue the new one.

4.2 Output feedback Longitudinal controller for gust harvesting: effect of actuators or sensors dynamics

As Mermoz is supposed to fly around 100m over the atlantic ocean, one plans to harvest or alleviate the turbulence created by the sea waves. Futhermore as the altitude is quite low, the aircraft trajectory must be holden in a given range rising a control law problem. [6] [5] have shown that there exists control law strategies for small UAVs which lead to power saving using the surrounding aircraft environment. They firstly shown that for a single pulsation sinewave vertical gust, one could harvest energy and witness a positive gain in altitude. The proposed control law was a PD control using the encountered vertical gust velocity measurement as input. In this example, one proposes a simple propotionnal feedback of the aircraft pitch angle as a horizontal stab control law, leading to a gain in altitude. However one presents the effect of actuator or sensor dynamic on the performance. 3 different first order dynamics have been studied with a time response of : 0.1 s 1s and 10s (respectively blue, orange and yellow on figure 7). The vertical gust is a 1Hz sinewave with hyperbolic tangent activation function, the amplitude is 1.5m/s. The feedback gain is set to -1.8. The figure 7 shows that for slow actuators one witness a delayed flap answer with smaller amplitude. However the performance in gain of altitude seems to be better with a slow actuator. This means that the optimum feedback gain used for the simulation should be smaller. Consequently if the actuator dynamic response is slower or near the aircraft one, it is highly possible that the control law will witness a change in its performance.

4.3 Performance robustness to a full bandwidth gust model:

The last example follows the same protocole as the previous example. However the authors has extended the gust

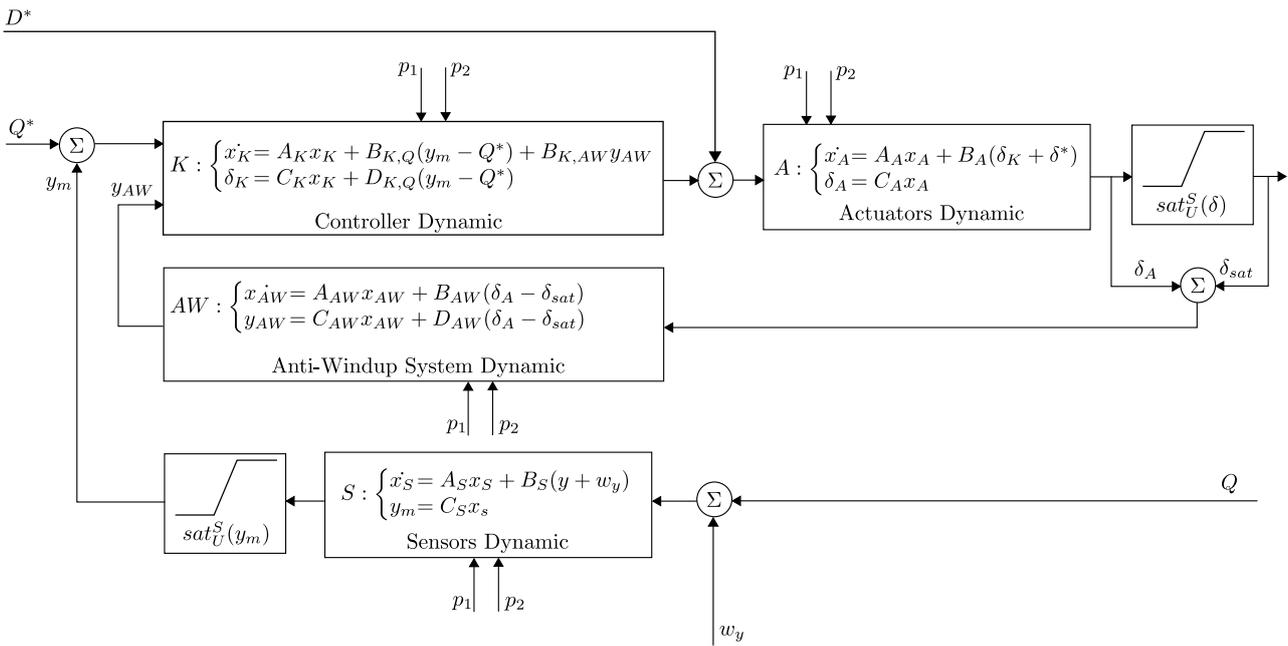


Figure 4: Linear control pattern proposed

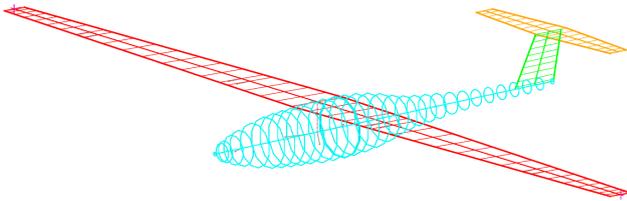


Figure 5: Mermoz aircraft: ASWING configuration

model function of ASWING with a full 3D spatio-temporal Von Karman turbulence model based on the adaptation to UAV work of [1]. The figure 8 clearly show that the control does not satisfy the energy harvesting objectives no matter the dynamic of the actuator. One must consequently seek for a better and more robust control law to fulfill the objective for an extended gust bandwidth.

5 CONCLUSION

This article has presented a methodology to modify ASWING source code so that control laws with internal dynamics commonly used by the control theory community can be implemented. One has presented the different algorithms used in ASWING and the proper modifications that have to be made to fulfill this objective. Some theoretical results have been recalled for a better understanding of the methodology. Moreover, one has to make sure that the modifications brought do not lead to an added stiffness on the differential equations leading to the solver divergence. Finally,

the modification has been illustrated with 2 examples to have a quick partial view of the modification. From this modification, the users can use this methodology to quickly implement and benchmark sophisticated bi-scheduled controls law such for example Linear quadratic Gaussian, H_∞ / H_2 , Guardian mapping, Fractional order controller, μ analysis. He can also investigate sensors and actuators' saturations effects and implement anti-wind-up and or observers. Sensors noises can also be taken into account for robustness study to stochastic environment. Future works aims at providing a "craftable flyable aircraft" design tool which respects ASWING theoretical background for automatic and parametric design of UAVs based on modern crafting technics

REFERENCES

- [1] Julian Theis, Harald Pfifer, and Peter Seiler. Robust Modal Damping Control for Active Flutter Suppression. *Journal of Guidance, Control, and Dynamics*, 43, March 2020.
- [2] David K. Schmidt, Brian P. Danowsky, Aditya Kotikalpudi, Julian Theis, Christopher D. Regan, Peter J. Seiler, and Rakesh K. Kapania. Modeling, Design, and Flight Testing of Three Flutter Controllers for a Flying-Wing Drone. *Journal of Aircraft*, 57(4), July 2020.
- [3] David Schmidt, Brian Danowsky, Peter Seiler, and Rakesh Kapania. *Flight-Dynamics and Flutter Analysis and Control of an MDAO-Designed Flying-Wing Research Drone*. January 2019.

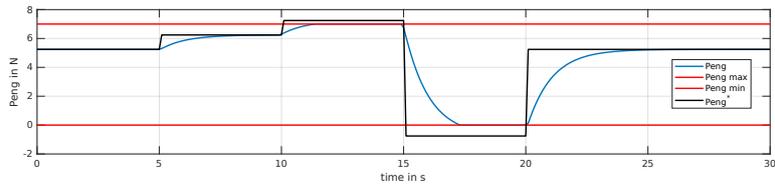


Figure 6: Effect of first order dynamic on actuators

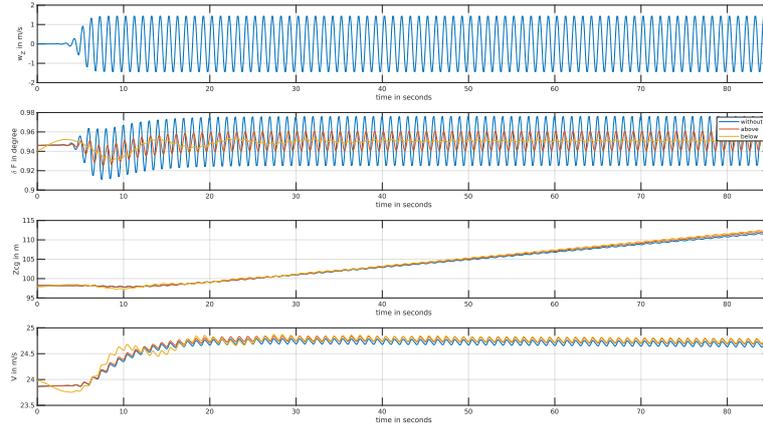


Figure 7: Single pulsation gust soaring and effect of the actuator dynamic on performance

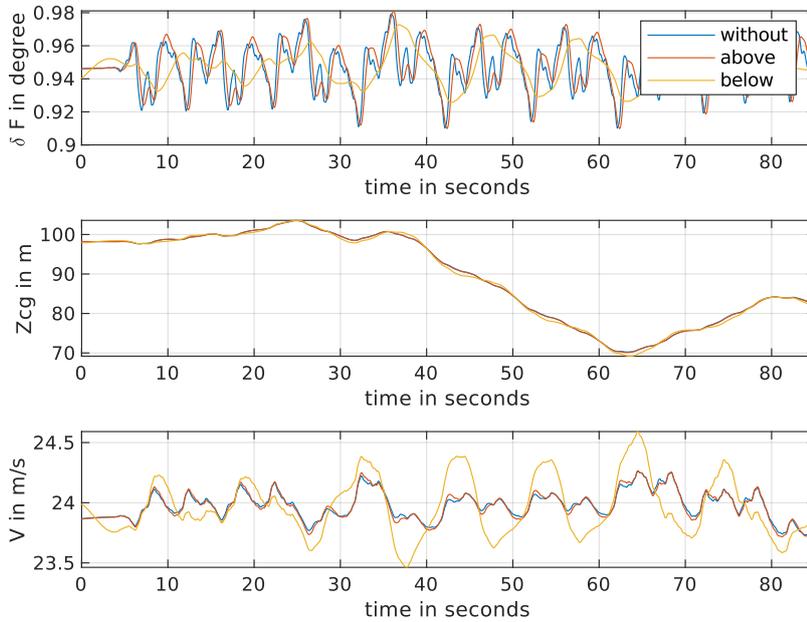


Figure 8: Robustness of gust soaring strategy for an extended bandwidth

- [4] N Gavrilovic, D Vincekovic, and J-M Moschetta. A Long Range Fuel Cell/Soaring UAV System for Crossing the Atlantic Ocean.
- [5] Nikola Gavrilovic, Emmanuel Benard, Philippe Pastor, and Jean-Marc Moschetta. Performance Improvement of Small Unmanned Aerial Vehicles Through Gust Energy Harvesting. *Journal of Aircraft*, 55(2), March 2018.
- [6] N Gavrilovic, A Mohamed, M Marino, S Watkins, J-M Moschetta, and E Benard. Avian-inspired energy-harvesting from atmospheric phenomena for small UAVs. *Bioinspiration & Biomimetics*, 14(1):016006, November 2018.
- [7] Ravi Jaiswal, Abhishek Shastri, Swati Swarnkar, and Mangal Kothari. Adaptive Longitudinal Control of UAVs with Direct Lift Control. *IFAC-PapersOnLine*, 49(1), January 2016.
- [8] R. F. Latif, M. K. A. Khan, A. Javed, S. I. A. Shah, and S. T. I. Rizvi. A semi-analytical approach for flutter analysis of a high-aspect-ratio wing. *The Aeronautical Journal*, 125(1284), February 2021. Publisher: Cambridge University Press.
- [9] Wang Rui, Zhu Xiaoping, and Zhou Zhou. Design Gust Alleviation Controller for Highly Flexible Solar UAV. In *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, volume 1, pages 930–933, January 2011. ISSN: 2157-1481.
- [10] Ya Wang and Daniel J. Inman. Autonomous Gust Alleviation in UAVs. In *Advanced UAV Aerodynamics, Flight Stability and Control*. John Wiley & Sons, Ltd, 2017.
- [11] Mark Drela. Integrated simulation model for preliminary aerodynamic, structural, and control-law design of aircraft. In *40th Structures, Structural Dynamics, and Materials Conference and Exhibit*, St. Louis, MO, U.S.A., April 1999. American Institute of Aeronautics and Astronautics.
- [12] Mateus d Virgilio Pereira, Ilya Kolmanovsky, Carlos E. Cesnik, and Fabio Vetrano. Model Predictive Control Architectures for Maneuver Load Alleviation in Very Flexible Aircraft. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2019-1591>.
- [13] Stuart P. Andrews. Modelling and simulation of flexible aircraft : handling qualities with active load control. March 2011. Accepted: 2012-12-13T11:36:23Z Publisher: Cranfield University.
- [14] Xuerui Wang, Tigran Mkhoyan, and Roeland De Breuker. Nonlinear Incremental Control for Flexible Aircraft Trajectory Tracking and Load Alleviation. *Journal of Guidance, Control, and Dynamics*, May 2021. arXiv: 2101.00594.
- [15] Mark Drela. ASWING 5.86 Technical Description — Steady Formulation.
- [16] Mark Drela. ASWING 5.81 Technical Description — Unsteady Extension.
- [17] Mark Drela. *2D transonic aerodynamic design and analysis using euler equations*. PhD thesis, MIT, 1983.
- [18] Eugene Isaacson and Herbert B. Keller. *Analysis of numerical methods*. Dover, New York, NY, 1994. OCLC: 30032279.

ACKNOWLEDGEMENTS

This work has not been fulfill in partnership with the main developer of ASWING namely Prof M.Drela from MIT, as it is not a open source and free to use software, one can not let the source code accessible. One strongly aims at proposing this modification as an official ASWING toolbox.