JDynaSim AS A SIMULATION TOOL TO STUDY THE DYNAMIC BEHAVIOUR OF SAILPLANES IN THERMALS

D. P. Coirot[†], F. Nicolosi[†], M. Madonna[†]
[†]Dipartimento di Progettazione Aeronautica,
Universita' di Napoli "Federico II"
Via Claudio 21 - 80125 Napoli, Italy
E.Mail: coiro@unina.it
Presented at XXVI OSTIV Congress, Bayreuth, Germany

SUMMARY

A fast and interactive software to simulate aircraft behaviour may be a very useful and helpful tool for aerospace research and design, especially when light aircraft and sailplanes are involved, and when atmospheric wind effects are to be considered. This paper presents an attempt to give interactive and realistic simulation possibility almost to everyone through the use of ***JDynaSim*** code.

JDynaSim is an interactive graphic flight simulation code written in JAVA and VRML (Virtual Reality Modelling Language) languages which practically allows any user with any hardware (PC, workstations, etc.) to *fly* the aeroplane under investigation. This is true due to the fact that JAVA is a language born to work under a generic Internet browser (such as Microsoft Explorer or Netscape) and thus it is independent from the operating system under which it is running. The code uses the VRML to represent the aircraft motion and JAVA classes to control and interactively *fly* the aeroplane, solving runtime the dynamic motion equations in which atmospheric wind terms have been included. A simple thermal model has also been implemented in the code and useful information on dynamic behaviour of sailplane can be drawn when this is flown through the simulated thermals. Parametric variation of sailplane aerodynamic and geometrical characteristics has allowed to investigate the relative importance of aerodynamic coefficients on sailplane performances in thermal soaring. This paper will present the results of such an investigation.

JDynaSim is available to everybody through Internet at the following URL:
http://www.dpa.unina.it/coiro/

NOMENCLATURE

$\bar{c}$ = mean reference chord
C.G. = center of gravity referred to l.e. of m.a.c.
D = drag force
g = gravity
I = moment and/or product of inertia
L = lift force
l = rolling moment
M = pitching moment
m = mass
N = yawing moment
p,q,r = rate of roll, pitch, and yaw, respectively
TM = transformation matrix Earth=>Body axes
t = time
$\overline{V}$ = relative velocity vector
V = absolute value of velocity vector
$\overline{W}$ = wind velocity vector
W = absolute value of wind velocity vector
x = distance along x axis
y = distance along y axis
z = distance along z axis
Y = y component of aerodynamic force
α = angle of attack
β = angle of sideslip
δ_e = elevator angle
ϑ = Euler angle (elevation)
ϕ = Euler angle (bank)
ψ = Euler angle (azimuth)
$\overline{\omega}$ = angular rotation vector
C_i = inertial terms

SUPERSCRIPTS

(.) = time derivative d/dt

SUBSCRIPTS

E = referred to inertial (Earth) coordinates
W = referred to wind coordinates
B = referred to body coordinates
x,y,z = measured in x, y, and z directions

INTRODUCTION

Sailplane and aircraft behaviour are better understood if simulation and visualization tools are available to the designer. In particular when complex and non-linear maneuvers are to be investigated, the necessity of really looking at the *true* aircraft motion, has led to the development of an interactive flight simulator. If students needs have also to be taken in account, portability of such a software and/or its availability through internet become important requirements. *JDynaSim* represents the result of efforts done toward these requirements and it is the arrival point of many researches performed by the authors in recently past years regarding aircraft aerodynamic and dynamic responses prediction.

JDynaSim originates from *DynaSim* (Refs. 1,2,3) which was an interactive graphic code that allowed the user to fly the selected aeroplane using the mouse as stick command. *DynaSim* was presented and shown at the last OSTIV XXV Congress. Originally *DynaSim* was partly written in C language and partly in Fortran language, but, in this case, the code was dependent on the machine it was running on (it was written for Silicon Graphics workstations) limiting in such a way its portability especially to the wider PC's user world. So, to make this code more portable, it has been totally translated in JAVA (Ref. 4) language allowing in this way to be used by virtually any computer with the only requirement that an internet browser, such as Microsoft Explorer or Netscape, should be installed on it along with the shareware software CosmoPlayer. In fact the code uses the Virtual Reality Modelling Language (VRML) (Ref. 5), thanks to which it is possible to create an environment which allows visualization of 3D aircraft motion resulting from the integration of motion equations.

JAVA, once communication with VRML has been established (Ref. 6), allows the user to control and interactively fly the aeroplane.

Big effort has been put in the process of transforming the original C and Fortran subroutines in JAVA language especially because JAVA is not a language oriented to scientific calculations, but the characteristics of the language make the code *user friendly* and machine-independent.

The practical use of such a tool is connected to the possibility of analyzing the aircraft response in real time for both interactive and pre-defined time dependent control laws. The necessary steps to perform an interactive session are:

1. Generation of geometry model (this is not necessary to correspond to the real geometry)
2. Generation of virtual reality model to be used by Cosmo Player. This is done through an automatic converter that translate a standard geometry data file in a VRML file.
3. Generation of geometry and mass file input
4. Generation of non linear aerodynamic data base eventually transforming the available forces from one reference system to the appropriate *JDynaSim* reference system
5. Run the simulation

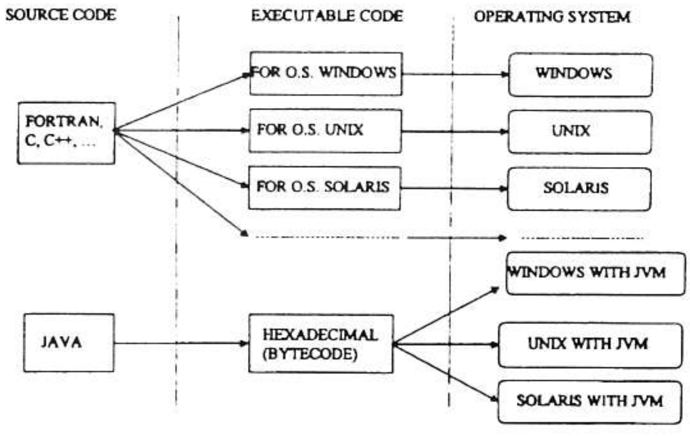
Steps 1 and 2 are necessary only the first time and they can be avoided if the user accepts to see the geometry of the model different from the real one.

Step 4 can rely on either experimentally determined aerodynamic forces or on data deriving from numerical simulation. Obviously data should comprise the whole range of angles of attack including the nonlinear part.

Previous work done in this direction {Refs. 2,3,7} has shown that this procedure can be reliably used to predict aircraft and sailplane performances and behaviour. Part of the current work has also being devoted to the inclusion of atmospheric wind terms as forcing functions of the motion added to the normal control forces forcing functions. In particular a simplified thermal model has been introduced in order to investigate the sailplane performances in thermal soaring. In fact one of the main objective of this paper is studying the influence of geometric parameters and aerodynamic coefficients variation on sailplane performance. We have tried to answer the following question: "Would the sailplane perform better if that coefficient or geometric parameter would be greater or smaller than actual value?" In order to reduce the number of degrees of freedom relative to the motion under investigation, the study has been done trying to obtain the same sailplane path in the thermal **assigning different control laws for** each maneuver!

JAVA LANGUAGE

JAVA is an object oriented programming language similar to C and to C++ that allows to write applets. Applets are applications ready to be transferred in internet and to be executed by a local machine with a simple internet browser like MS Explorer or Netscape Navigator. In other words an applet is like a normal audio or image file and it can be downloaded and executed using the browser. JAVA has then introduced interactivity in internet. To run JAVA applets, the browser must include the Java Virtual Machine (JVM) that is now normally included by default in most common browsers. The portability of JAVA and its characteristic of being platform independent is connected to the way a source code is *compiled*. In fact, after the compilation, the output is in a *bytecode* format that is in a hexadecimal format and not in machine language, see sketch n. 1.



Sketch n. 1: JAVA independence from platform

The interpreter of this type of language is the JVM. In this way portability and flexibility are ensured. In practice JAVA is a *general purpose* and complete language, easily extendable to multimedia and to networks.

Being a new language born having in mind internet, it has rapidly become a standard for applications available to everyone in the net.

Another advantage is that it is free of charge and its compiler is freeware.

It also allows a lot of library for a broad range of applications starting from networking to data base management.

VRML

The Virtual Reality Modeling Language (VRML) is the language used to describe three dimensional objects for Internet. With VRML, authors can construct entire sites, or "worlds," with infinite space and depth. Objects in these worlds can be links to text, audio, or video files, HTML files or sites, or links to other VRML worlds.

A VRML file is an ASCII file containing 3D graphical data and it is read by specific browsers allowing the user to navigate in an interactive way in three-dimensional scenes. It is recently born, it is shareware and it represents a standard in Internet. Some of the possible uses of virtual reality with VRML are: teaching, scientific visualization, dynamic simulation, industrial design, medicine, etc.

Main advantages are its portability, economy, diffusion, versatility

Even if it is a file format and not a real programming language, it allows a dynamic control of the graphical scene through JAVA code and with the shareware library EAI *(External Authoring Interface)*.

DynaSim AND *JDynaSim*

Dynasim (Refs. 1,2,3) is the original interactive graphic code that allows the user to *fly* the selected airplane using the mouse as stick command. It was written using a combination of C and Fortran languages and was based on Open GL graphic libraries running under Unix operating system on a Silicon Graphic workstation. Its use was then limited to people owning Silicon workstation who represent only a very small part of total users. The portability of *DynaSim* to PC world was the main reason to investigate the possibility to write a totally independent platform software. In fact the simple transformation of the graphical part of the code to Open GL running under Windows operating system, would have limited the use of the code to people owning graphic card supporting Open GL. Then only a combination of JAVA and VRML represented a solution of the problem. The communications between JAVA and VRML has not been an easy task. The other difficult part has been translating the numerical part of the code in JAVA language that is a language born for the graphic world and then not really *smooth* for numerical computation! ***JDynaSim*** screen snapshot is reported in figure 1.

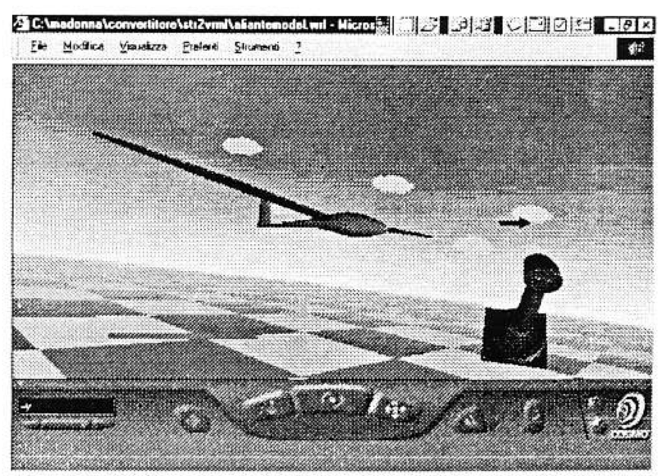


Figure 1. Example of screen shot of JDYNASIM flight simulator code: ASW-24 sailplane.

Aerodynamic prediction

Longitudinal and lateral-directional aerodynamic coefficients, static and dynamic stability derivatives and sailplane performances are obtained through the use of ***AEREO*** code. Without going in detail about the features of *AEREO* code (they can be found in Refs. 2,3,7) we just remind here that *AEREO* predictions are based on a mix of semi empirical methods (see references 8-12) and of an extension of Prandtl lifting line theory to the non linear range of angles of attack. *AEREO* predicts also power effect on aerodynamic coefficients together with static and dynamic stability characteristics (characteristic modes, frequencies, root locus, etc.). Aircraft and sailplane performances along with load- and gust-envelopes are also generated. *AEREO* code needs two dimensional airfoil aerodynamic coefficients assigned as curves in function of angle of attack and of Reynolds number. These data are necessary for each lifting surface and they can be either experimental or numerical data. In this last case we use aerodynamic coefficients as obtained from TBVOR code (Ref. 13). Examples of aerodynamic predictions as predicted by AEREO can be found in reference 7.

Equations of motion

The translation equations of motion are written in the flight path axis system and the rotational equations of motion are written in a fixed body axis system. The equilibrium values of the variables corresponding to the trimmed input condition are first found and then the integration of the differential equations starts. The code can interactively read mouse and keyboard inputs as well as files with command laws assigned in function of time. There is the possibility to record the interactive session performed and then to repeat the maneuver.

The solution is obtained solving 12 ordinary non linear differential equations using a 4th order Runge-Kutta integration scheme. Aerodynamics is input as non-linear forces and moments expressed in function of angle of attack, sideslip angle and control surfaces deflections, thus they are input in multidimensional matrix form and are interpolated at each instant of time according to the instantaneous values assumed by independent parameters during the maneuver.

Wind terms in the motion equations

The inertial reference frame is the Earth axis system, based on the assumption that the Earth is a stationary plane in an inertial space. The coordinate systems used in this work have been: flight path axis system (origin at the c.g. of the airplane and X_w axis directed along the velocity vector of the vehicle relative to the atmosphere), atmosphere fixed reference frame (useful to evaluate the inertial wind velocity $\overline{W}$), and body axis system (vehicle fixed reference frame, based on the assumption of rigid and symmetrical aircraft). In order to incorporate the variation of the wind in the motion equation, the inertial velocity $\overline{V}_E$ (that is the aircraft speed vector relative to the Earth reference frame) must be expressed in terms of the wind vector in this way: $\overline{V}=\overline{V}_E-\overline{W}$ where the atmosphere motion relative to the Earth is indicated with $\overline{W}$ and $\overline{V}$ is the aircraft speed relative to the atmosphere. This means that the translational equation of motion becomes $\dot{\overline{V}}+\overline{\omega}\times\overline{V}=\overline{T}/m+\overline{A}/m+\overline{g}-(\dot{\overline{W}}+\overline{\omega}\times\overline{W})$. Of course if the wind speed is zero or constant, the equation will return to its original form. When vectors are projected on the path reference frame, translational equations will be modified consequently (Refs. 14,15,16). The other obvious consequence of the wind variation is that all aerodynamic forces and moments are expressed in terms of the aircraft speed relative to the atmosphere (for istance $D=C_D\rho V^2S/2$, where V is different from V_E).

Angular rates relative to body axes will be affected as well because it is necessary to add terms following the trasformation of angular wind velocity $\overline{\omega}_W=\frac{1}{2}\overline{\nabla}\times\overline{W}$ in body axis sytem. Due to the presence of $\overline{\nabla}$ operator, spatial derivatives of $\overline{W}$ components will be used in evaluating the aerodynamic coefficients. In fact the roll, pitch and yaw moments are dependent on the values of the angular rotation relative to the atmosphere. So, since the aircraft angular velocity is evaluated in the Earth reference frame - and its components p, q and r are expressed on the body reference frame the angular wind velocity $\overline{\omega}_W$ will not appear in the rotational equations, but it will affect the calculation by its direct influence on the aerodynamic moments. To evaluate this influence the angular rates must be modified as follows: $\overline{\omega}=\overline{\omega}_B-\overline{\omega}_{WB}$, where $\overline{\omega}_{WB}$ is the angular wind velocity projected on body axis and $\bar{\omega}$ is the final angular velocity ([pr, qr, rr]^T) vector to be used in computing the aerodynamic moments.

To obtain ω_{WB} a coordinate transformation matrix will be used, since the wind angular velocity is usually evaluated in the Earth axis system, as

indicated here $\bar{\omega}_W = \begin{bmatrix} (\partial W_z/\partial y - \partial W_y/\partial z) \\ (\partial W_x/\partial z - \partial W_z/\partial x) \\ (\partial W_y/\partial x - \partial W_x/\partial y) \end{bmatrix}$. The transformation matrix and the complete set of non linear ordinary differential equations are reported in appendix.

Thermal modelling

Up to our knowledge, no complete thermal models are available in current literature. In this case one possible way of proceeding is to choose a relatively simple model that can be quite representative of a real thermal. A complete thermal model would include a distribution of wind velocity components in space and in time.

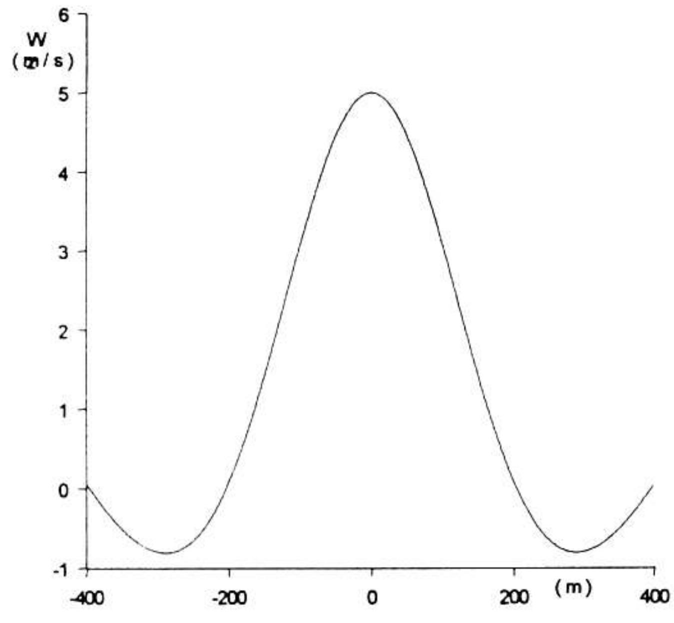


Figure 2: W_z velocity component inside the thermal

Our simplified model not only "freezes" the velocity distribution in space but the velocity components are also reduced to only one component (normal component to the earth) that varies only horizontally (in X-Y plane) according to a specified function. Furthermore the thermal has a circular shape (it develops in an axial-symmetric body of revolution) and the total mass is conserved inside the thermal itself. Figure 3 shows vertical wind distribution in a section obtained cutting the thermal with a plane normal to the diameter and passing through the centre. The thermal diameter is about 800 meters with maximum upward velocity varying from 5 to 10 m/s and maximum downward velocity form 1 to 2 m/s.

Results

All calculations have been performed for the ASW24 sailplane. We did not have the accurate geometry of this sailplane and most of geometrical data were deduced from available pictures. Mass and inertia data were available to the authors. As far as it concerns sailplane aerodynamic predictions, they were already presented in Ref. 2. Once aerodynamic coefficients and stability derivatives are available, simulation has mainly regarded two types of maneuver:

1) Straight crossing the thermal

2) Entering and climbing the thermal keeping the trajectory approximately the same.

Sensitivity analysis has been performed to establish those parameters and aerodynamic coefficients to which the sailplane would have been more sensitive.

This analysis has been performed in the following way: for each maneuver, as specified above, the sailplane has been trimmed at same initial speed and height outside the thermal. Then for each parameter or coefficient variation the control surface deflections have been assigned (and, where necessary, modified) in order to obtain approximately the same path inside the thermal.

Altitude gain at the end of the maneuver has been considered as the performance index.

Aerodynamic coefficient curves have been modified changing their slopes. For example, regarding the lift coefficient, keeping the same Cl_{max}, curve slope has been changed increasing and decreasing it by an amount of 10%.

Same procedure has been applied to all coefficients and only a reduced set of them has resulted having a significant influence on performance index.

Straight crossing maneuver

In the first part of table 1, main characteristics of original sailplane maneuver are reported. With reference to these data, altutude gain or loss are reported in the remaining part of the table.

Table n. 1

Original sailplane (weight = 500 Kg, C.G = .25 m)	
Max height(m)	1049.8
Min height (m)	988.
Angle of attack(deg.)	$2 < \alpha < 2.7$

Modified sailplanes	Max height variation	Min height variation	Angle of attack
+10% CL_α	Null	Null	$2 < \alpha < 2.7$
+10% CD_α	Null	Null	$2 < \alpha < 2.7$
+10%Cm_α	Null	Null	$2.1 < \alpha < 2.$
Weight = 350 Kg	+ 4. m	Null	$-0.5 < \alpha < 0.5$
C.G. .19 m	Null	Null	$2.2 < \alpha < 2.8$
C.G. .32 m	Null	Null	$2 < \alpha < 2.7$
Wing span: (+13%)	Null	- 1.5 m	$1.9 < \alpha < 2.4$
Wing surface(+15 %)	+2.5 m	-3. m	$0.6 < \alpha < 1.9$
Horizontal plane(+10%)	~ -1. m	-2.5 m	$1.95 < \alpha < 2.6$

It can be seen that only wing surface variation has a small impact on performance index.

Climbing maneuver

A typical trajectory in the X-Y plane is shown in figure 3. In the same figure dashed lines represent the marks of the thermal with outside and inside rings representing respectively downward and upward velocities.

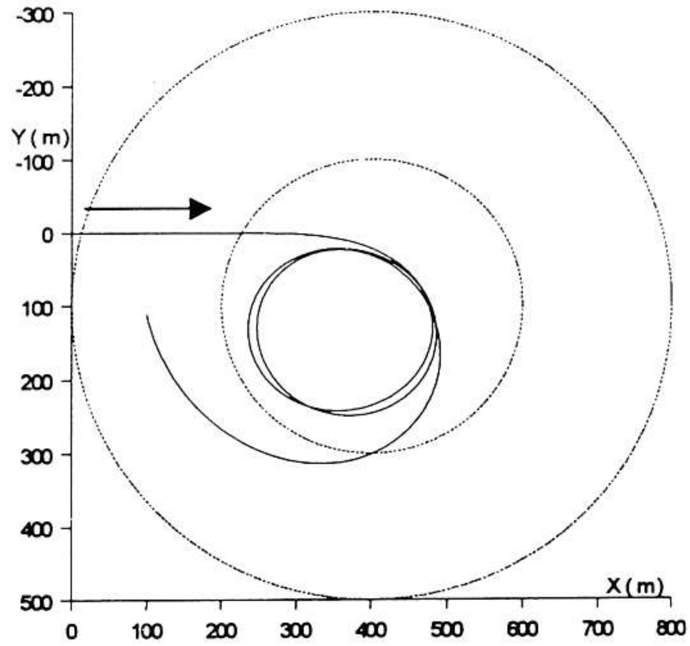


Figure 3. Path of sailplane inside the thermal for climbing maneuver.

Altitude in function of time is reported in figure 4. It should be noted the oscillatory character of the sailplane path: this is due to the low damping of the phugoid motion.

We remind that sailplane path has been held approximatley the same for all cases investigated and that appropriate control laws have been assigned for each case studied. Typical elevator and aileron control laws are reported in figure 5.

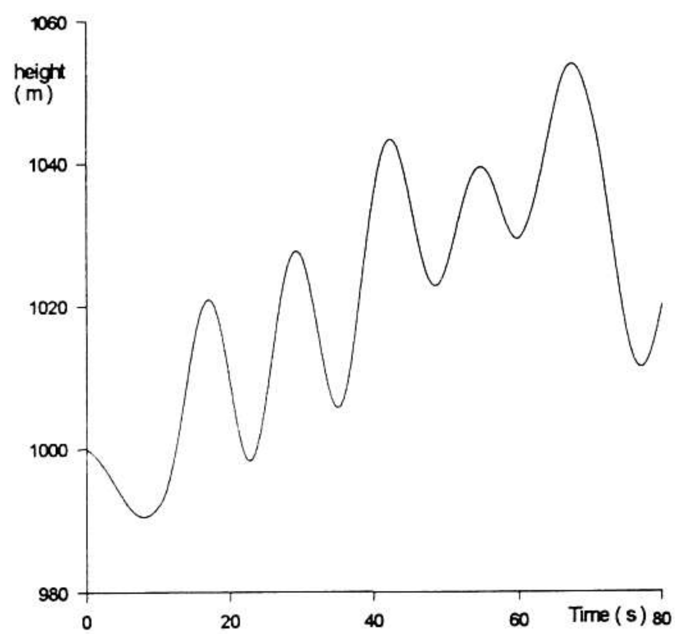


Figure 4: Sailplane altitude in function of time.

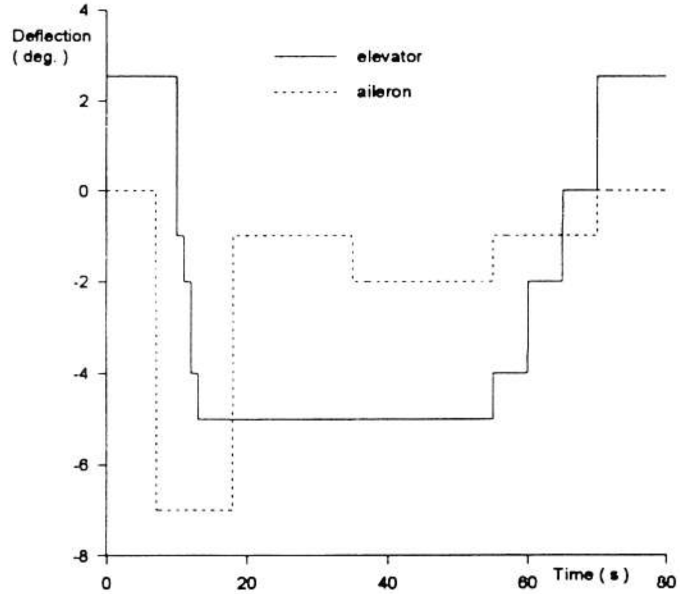


Figure 5: Control laws for climbing maneuver.

Variation of angle attack and bank angle in function of time for a typical maneuver are reported in figures 6 and 7.

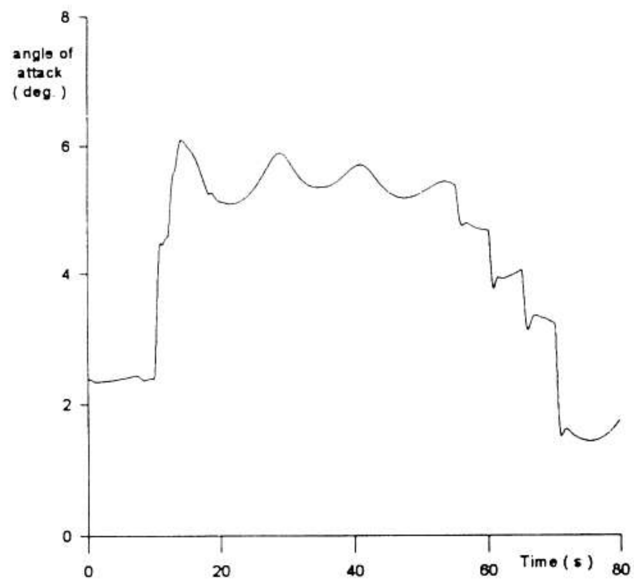


Figure 6: Angle of attack variation during the climbing maneuver.

Again oscillations are present in both graphs. Nevertheless average bank angle is around 42 degrees. In table n. 2 results of parameters variation are reported and they are relative to the climbing maneuver. It can be seen that significant effect on performance index are due only to wing surface modification, and to lesser extent, to CL_x

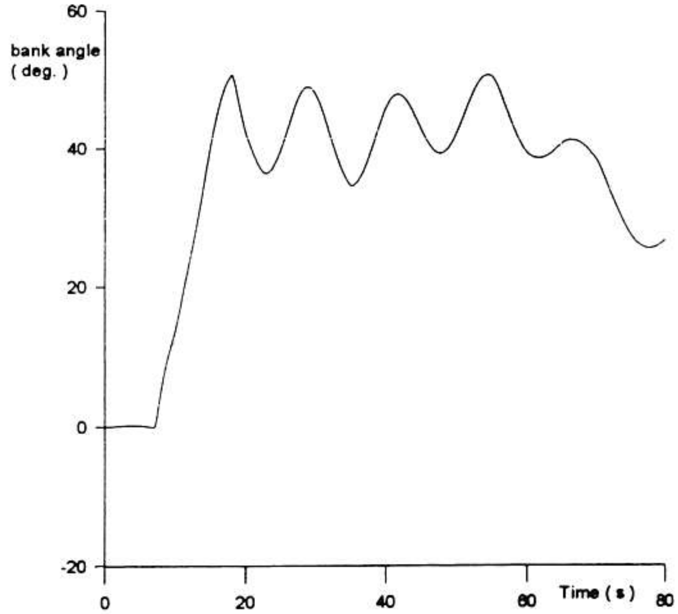


Figure 7: Sailplane bank angle during climbing maneuver.

The effect of wing surface variation on resulting motion is shown in figure 8. Figure 9 shows the altitude gain and loss due to increasing and decreasing lift coefficient slope.

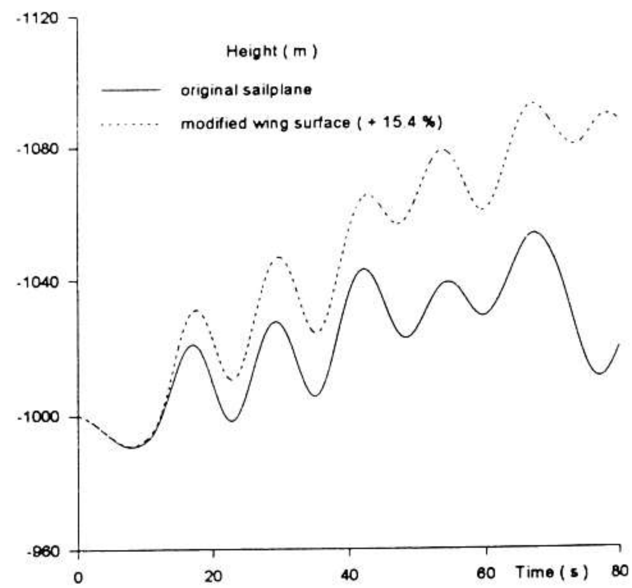


Figure 8: Altitude gain due to wing surface variation.

Cm_α and CD_α derivatives have only a little effect on performance index.

As weight concerns, lighter glider will obviously climb better.

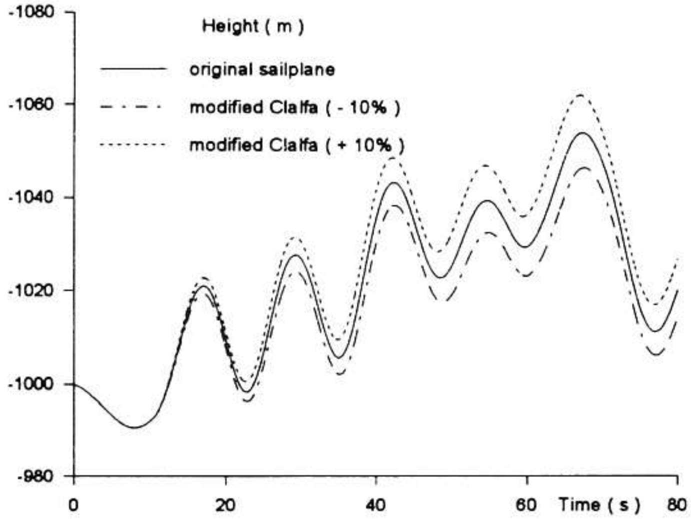


Figure 9. Altitude gain/loss due to lift coefficient slope variation.

Table n. 2

Original sailplane	
Max height (m)	1053.8
Min height (m)	990.5
Mean angle of attack (deg.)	< 5.5
Mean angle of bank (deg.)	~45
Mean angle of sideslip (deg.)	~3

Modified sailplane	Max height variation (m)	Mean angle of attack variation	Mean angle of bank variation	Mean angle of sideslip variation
+10% CL_α	+ 8.1	Null	Null	Null
-10% CL_α	- 7.5	Null	Null	Null
+10% CD_α	- 3	Null	Null	Null
-10% CD_α	+ 2.5	Null	Null	Null
+10% Cm_α	+ 2.5	Null	Null	Null
-10% Cm_α	- 2.5	Null	Null	Null
+10% $Cn_{\delta a}$	Null	Null	Null	Null
-10% $Cn_{\delta a}$	Null	Null	Null	Null
+10% Cy_β	Null	Null	Null	Null
-10% Cy_β	Null	Null	Null	Null
Wing surf.(+15%)	+ 20.	+ 0.5	- 10.	+ 0.5
Weight: 350 Kg.	+ 40.	- 2.0	- 10.	+ 0.5

All other longitudinal and lateral-directional aerodynamic coefficients and derivatives have shown one or little influence on performance index. No specific indications came out from this study but the possibility of changing the performance index and the ease of performing the sensitivity analysis, will allow further investigations in the direction indicated in this paper.

Conclusion

JDynaSim has shown to be a valuable tool to investigate and visualize sailplane maneuvers. Used in combination with *AEREO* code it allows to study the effect of geometrical and aerodynamic parameter variation on sailplane performances. An example of this type of study has been presented in this paper. Despite the simplicity of thermal model, useful information on relative importance of aerodynamic and geometric parameters on straight and climbing flights have been obtained and discussed. Through this type of investigations a designer can be helped during the designing phase where decisions have to be taken to optimise a particular performance. The possibility of ***JDynaSim*** use over internet will allow a wide number of people to become closer to the gliding world.

References

1) Coiro D.P, P. Leoncini, *"Dynamic Stability Evaluation and Graphic Visualization of Aircraft Motion at High Angles of Attack,"*XIII AIDAA Congress, Roma, September 1995.

2) Coiro D.P., Nicolosi F., *"Aerodynamics,Dynamics and Performance prediction of Sailplanes and Light Aircraft," XXV* OSTIV Congress, St. Auban, France, July 1997. It will be published on Technical Soaring Journal

3) Coiro D.P., Nicolosi F., Avino M., *"Light Aircraft Aerodynamics, Dynamics and Performances Prediction: AEREO and DYNASIM codes," XIV* AIDAA National Congress, Napoli, Italy, 20-24 October 1997

4) Naughton, P., Schildt, H.; *"Java - The complete guide,* "- McGraw - Hill 1997

5) *"The Virtual Reality Modeling Language Specification, Version* 2.0," ISO/IEC CD 14772, 4th August 1996, also available at **httl)://vrml.sgi.com/movin-worldspec**

6) Scott, A.; *"The Marriage of Java and VRML", also* available at **http://vrml.sgi.com/developer/index/index.html**

7) Coiro D.P., Marulo F., Nicolosi F., Ricci F., *"Numerical, Wind-Tunnel and Flight Tests for P92J and P96 Light Aircraft,"* ICAS 21 - Melbourne, Australia, September 1998

8) Wolowicz, H. C. and Yancey, B. *"Longitudinal Aerodynamic Characteristics of Light, Twin-Engine, Propeller-Driven Airplanes,"* NASA TN D-6800, 1972

9) Wolowicz, H. C. and Yancey, B. R., *"Lateral Directional Aerodynamic Characteristics of Light, Twin-Engine, Propeller-Driven Airplanes,"* NASA TN D-6946, 1972

10) Roskam, J. *"Airplane Design. Part VI:* **Preliminary Calculation of** *Aerodynamic, Thrust and Power Characterisffc,"* Ed. Roskam Aviation 1987

11) DATCOM Data Sheets

12) ESDU Data Sheet

13) Coiro D.P, P. Dini, *"The Prediction of Aerodynamic Stall Over Wind-Turbine Airfoils,"* International Conference on fluid and thermal energy conversion '97 Yogyakarta, Indonesia, July 1997

14) Etkin, B., *"Dynamics of Atmospheric Flight, "*John Wiley & Sons, Inc., New York, 1972

15) Etkin, B., "Turbulent Wind and Its Effect on Flight," J. of Aircraft,Vol. 34, No. 5, May 1981

16) Coiro D.P., Madonna M. *"Investigation of Turbulence Effect on Dynamic Behaviour of Aircraft Through Use of JDYNASIM: a Platform Independent Simulation Software"* **First International** Conference on Advanced Engineering Design, Prague, June 1999

APPENDIX

Translation equations are written in path axes.; rotational equations and angular rates are written in body axes. Trajectory coordinates are written in earth axes. Terms due to turbulence are written in bold. It should also be noted that angular rates derived by gust components must be added to p,q,r values resulting from the equations solution.

$$\dot\alpha = q - p\cos\alpha\tan\beta - r\sin\alpha\tan\beta + \{mg(\sin\theta\sin\alpha + \cos\theta\cos\phi\cos\alpha) - T\sin\alpha - L\}/(mV\cos\beta) + \boldsymbol{[\dot W_x\sin\alpha - \dot W_z\cos\alpha + q W_x\cos\alpha - W_y(p\cos\alpha + r\sin\alpha) + q W_z\sin\alpha]/(V\cos\beta)}$$

$$\dot\beta = p\sin\alpha - r\cos\alpha + \{mg(\cos\theta\sin\phi\cos\beta + \sin\theta\cos\alpha\sin\beta - \cos\theta\cos\phi\sin\alpha\sin\beta) - T\sin\beta\cos\alpha + Y\cos\beta + D\sin\beta\}/(mV) + \boldsymbol{[\dot W_x\cos\alpha\sin\beta - \dot W_y\cos\beta + \dot W_z\sin\alpha\sin\beta - W_x(r\cos\beta + q\sin\alpha\sin\beta) + W_y(p\sin\alpha\sin\beta - r\cos\alpha\sin\beta) + W_z(q\cos\alpha\sin\beta + p\cos\beta)]/V}$$

$$\dot V = \{T\cos\beta\cos\alpha + mg[\cos\theta\cos\phi\sin\alpha\cos\beta - \sin\theta\cos\alpha\cos\beta + \cos\theta\sin\phi\sin\beta] + Y\sin\beta - D\cos\beta\}/m \boldsymbol{- (\dot W_y\sin\beta + \dot W_x\cos\beta\cos\alpha + \dot W_z\sin\alpha\cos\beta) - W_x(r\sin\beta - q\sin\alpha\cos\beta) - W_y(p\sin\alpha\cos\beta + - r\cos\beta\cos\alpha) - W_z(q\cos\beta\cos\alpha - p\sin\beta)}$$

$$\dot p = (C_1 r + C_2 p)\, q + C_3 l + C_4 N$$

$$\dot q = C_5 r p + C_6 (r^2 - p^2) + C_7 M$$

$$\dot r = (C_8 p + C_9 r)\, q + C_4 l + C_{10} N$$

$$\dot\vartheta = q\cos\phi - r\sin\phi$$

$$\dot\phi = p + \psi\sin\theta$$

$$\dot\psi = (r\cos\phi + q\sin\phi)/\cos\theta$$

$$\dot x = V[\cos\alpha\cos\beta\cos\theta\cos\psi + (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)\sin\beta + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\sin\alpha\cos\beta] + \boldsymbol{W_{xE}}$$

$$\dot y = V[\cos\alpha\cos\beta\cos\theta\sin\psi + (\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)\sin\beta + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\sin\alpha\cos\beta] + \boldsymbol{W_{yE}}$$

$$\dot z = V(-\cos\alpha\cos\beta\sin\theta + \sin\beta\sin\phi\cos\theta + \sin\alpha\cos\beta\cos\phi\cos\theta) + \boldsymbol{W_{zE}}$$

Matrix to transform terms from Earth reference frame to Body reference frame:

$$TM = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}$$