# HyPSim: a simulation tool for hybrid aircraft performance analysis

Vittorio Cipolla, Fabrizio Oliviero

**Abstract** This work presents performance prediction contribution provided by the research team at the University of Pisa. The results are part of the *HYPSTAIR (Development and validation of hybrid propulsion system components and sub-systems for electrical aircraft)* European project on the development and validation of hybrid propulsion system components and sub-systems for electrical aircraft. The first part of the paper discusses the performance analysis of a serial hybrid general aviation airplane for a reference mission profile. In particular, the best flight performance is evaluated varying the relevant mission parameters (e.g. range, cruise altitude, cruise speed) and the amount of available energy, in terms batteries and fuel. In the second part, a hybrid plane simulator, conceived to implement different mission profiles and to include pilot effects on power management by adopting a human-in-the-loop approach, is presented. Such simulator consists of three main software modules linked to each other in real time: a flight simulator, used to compute the aerodynamic forces and to visualize the airplane in flight, a flight planner, in which the mission profile can be defined, and a performance module, which calculates the instantaneous consumption of energy and provides the endurance prediction.

## 1 Introduction

In the *HYPSTAIR* project, the object of study is a general aviation aircraft in which the propulsion system is made of a propeller driven by an electric motor, that can be fed by both batteries and an Internal Combustion Engine (ICE) used as generator. The architecture of the hybrid system is serial, which means that the ICE is not

V. Cipolla
University of Pisa. Department of Civil and Industrial Engineering, Aerospace Section.
Via G. Caruso 8, 56122 Pisa (Italy).
e-mail: vittorio.cipolla@for.unipi.it

directly connected to the propeller but it is used as a source of electric power (Fig. **??**).

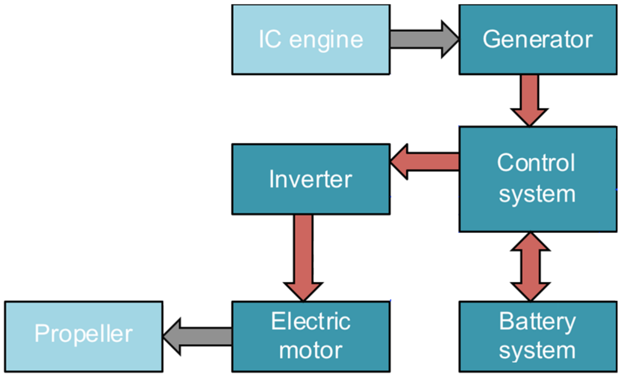


**Fig. 1** Serial hybrid architecture

This architecture gives more flexibility in terms of power management and it can provide a significant reduction of environmental impact, increasing safety at the same time. In fact, the presence of two independent energy sources introduces a redundancy and, in addition, an electric motor is more reliable than a piston engine.

## *1.1 Preliminary analysis*

The design of the hybrid system depends on mission requirements such as flight range, cruise altitude and speed, etc.

Concerning the novelty of the system here considered, a preliminary performance analysis has been performed in order to define the operating requirements limitations and to optimize the use of different energy sources. As detailed in [**?**], such analysis has been carried out implementing a simple but reliable performance model and considering the aerodynamic characteristic of an existing aircraft and the reference mission as shown in Fig. **??**, where the adopted assumptions are indicated in Table **??**.

The operating modes of the entire powertrain have been varied in accordance to the power request along the mission: battery packs provide additional energy during the most power demanding flight segments (take-off and climb), while during cruise the ICE generator gives sufficient power for both flight and battery charge.

The energy consumption has been evaluated through an energy balance. In particular, the efficiencies of all the components of the power architecture have been taken into account to define the mission parameters providing the best performance in terms of range or take-off weight. Therefore, two problems have been studied:

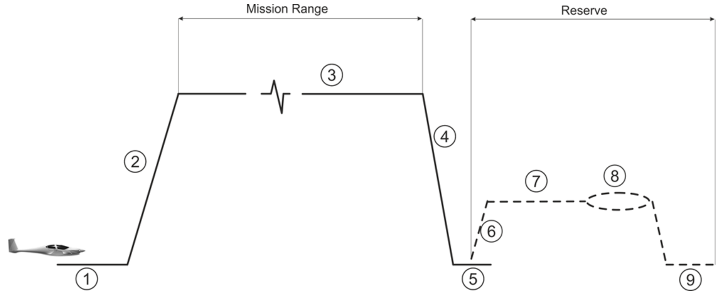


**Fig. 2** Reference mission for the hybrid aircraft

**Table 1** Hypotheses on mission segments

| Segment | Parameters | Hypotheses |
|---|---|---|
| Climb | $H_{in}$, $H_{fin}$, $P_{batt}$ | Fast climb |
| Cruise | $V_{cruise}$ or $P_{cruise}$, $H_{cruise}$, range | Constant Speed (or Power) and Altitude |
| Descent | - | Negligible for energy calculation |
| Landing | - | Negligible for energy calculation |
| Diversion | range, $H_{div}$ | Minimum power |
| Loiter | time , $H_{loi}$ | Max. endurance |

- evaluating the maximum flight range achievable with given amount of available energy (fuel + batteries) at take-off;
- evaluating the minimum energy amount (fuel + batteries) required at take-off in order to fly for a given range.

It has been observed that batteries affect performance mostly in flight segments such as climb or first part of the cruise, when batteries recharge occurs (see Fig. **??**).

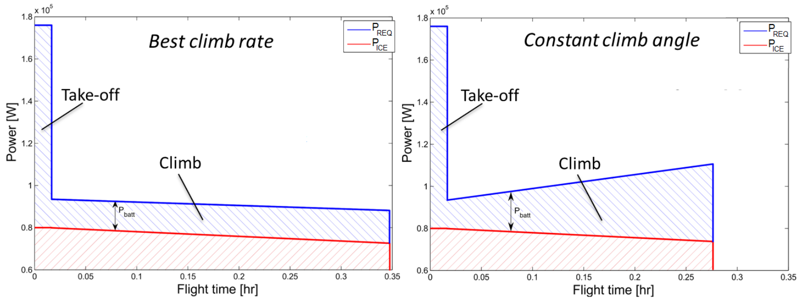


**Fig. 3** Effect of different climb programs on power sources

Therefore, as Fig. **??** suggests, differences between a hybrid and an internal combustion propulsion are more evident when the mission range is smaller. For long range mission, indeed, the energy contribution of batteries is less significant and flight performance are largely dominated by the ICE used as generator.

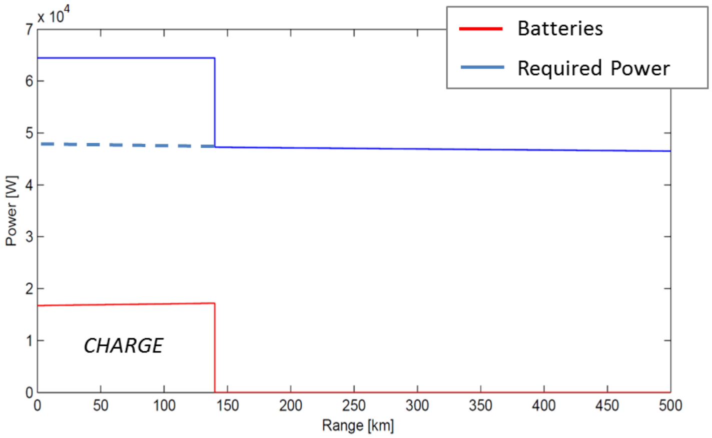


**Fig. 4** Required power VS Range

Finally, Fig. **??** shows the flexibility analysis of the hybrid aircraft compared to a traditional propulsion version, whose Maximum Take-Off Weight has been indicated as MTOW$_{ref}$.

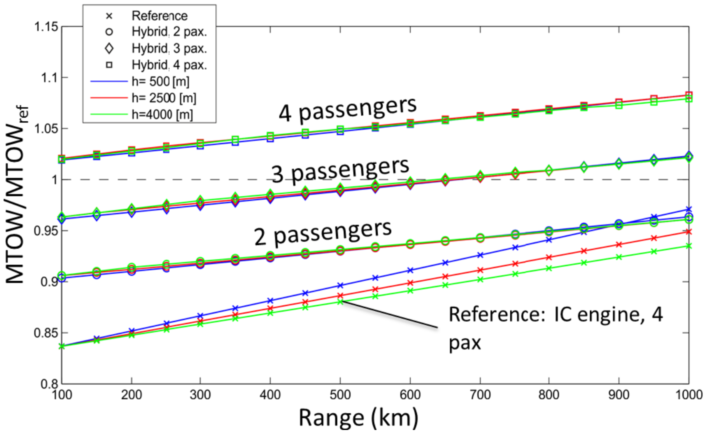


**Fig. 5** Flexibility analysis of the hybrid aircraft and comparison with traditional propulsion

As a preliminary result, the following conclusions can be given:

- if compared to internal combustion propulsion, the hybrid solution has significant influence on climb performance, whereas effects on cruise segment are smaller;
- since hybrid propulsion is less sensitive to altitude and range requirement has a weak influence on MTOW, the hybrid solution is more flexible than the traditional one;
- batteries energy and power densities play a key role.

## 1.2 Overview of the simulator HyPSim

The simulator HyPSim (Hybrid Plane Simulator) has been set up in order to:

- validate the previously achieved results in order to meet the given requirements;
- evaluate the performance for different mission profiles (defined by the user and performed manually or using an autopilot);
- simulate the human-in-the-loop effects on flight performance and, in particular, on power management;
- simulate the instantaneous performance of the aircraft depending on the instantaneous battery State of Charge (SoC);
- be used as a dissemination tool with user-friendly interfaces.

## 1.3 HyPSim architecture

As described in [**?**], HyPSim is composed of three main software: a *Flight Simulator*, in which the airplane is displayed and flight data are calculated (position, angles, speed, forces, etc.); a *Flight Planner* which allows to define the mission profile and the flight mode (manual or automatic); a *Performance Module* in which the hybrid propulsion system is modelled by means of analytical relations and flight data are processed for performance estimations and endurance prediction.

Finally, the main flight parameters, such as flight speed, fuel/battery consumption, etc., are shown on a Human-Machine Interface (HMI) panel. The conceptual arrangements of the simulator is reported in Fig. **??**.

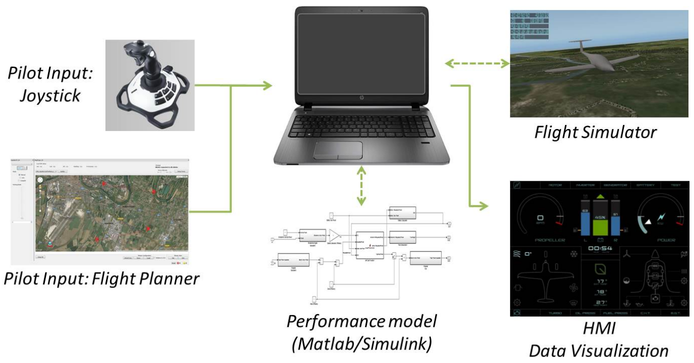


**Fig. 6** Conceptual layout of the Hybrid Plane Simulator

The simulation is performed through the following process:

- the pilot indicates a reference mission at the beginning by indicating a set of waypoints (longitude, latitude, altitude);

- a first estimation of endurance is provided and mission feasibility is evaluated;
- flight can be performed manually (joystick) and/or by means of a an autopilot which helps in performing the waypoint flight;
- the pilot can both use the joystick and modify the waypoints by using the flight planner;
- at each time step, the energy consumption (fuel and batteries) is estimated and the endurance prediction updated;
- instantaneous data (flight parameters, power flows, etc.) and endurance prediction can be displayed via the HMI.

Data are exchanged in real time between the different modules through a set of plugins programmed in C++. Such data can be divided into the following datasets:

- *Aircraft dataset*: it contains information about the initial conditions of the aircraft, such as the amount of embarked fuel, the initial SoC of batteries, the ICE generator characteristics, etc.;
- *Mission dataset*: it contains a description of the mission profile by means of waypoints, which can be modified during the mission;
- *Energy dataset*: it contains the instantaneous values of required flight power and energy consumption of both fuel and battery, thus it is updated continuously.

During the simulation, the main flight data are also recorded directly in a log file, in such a way that part of the calculations performed can be verified in a post processing phase, and the user has a complete overview on mission parameters.

## 2 The *Flight Simulator*

The commercial flight simulator *X-Plane* ([**?**]) has been implemented in `HyPSim` and used as aerodynamic solver. It provides reliable data on aircraft aerodynamics by means of a panel method that computes the aerodynamic forces at each instant.

*X-Plane* has been chosen since it is easy to interface with other codes and contains a parametric tool for the creation of new aircraft, called *Plane-Maker*, by means of which the user can create all the aircraft components, such as wing, fuselage, blades, control surfaces, landing gears, etc. The resulting model, shown in Fig. **??**, includes the airfoil characteristics which have been added in order to increase the accuracy of the panel method.

Results provided by the aerodynamic model implemented in *X-Plane* have been validated through experimental data provided by the manufacturer. Fig. **??** shows such comparison, which indicate a good accuracy for the model, although some differences are observed at low speed conditions.

During the simulation, aerodynamic forces, speed etc. can be extracted from the *Flight Simulator* and mission parameters can be updated at the same time (Fig. **??**).

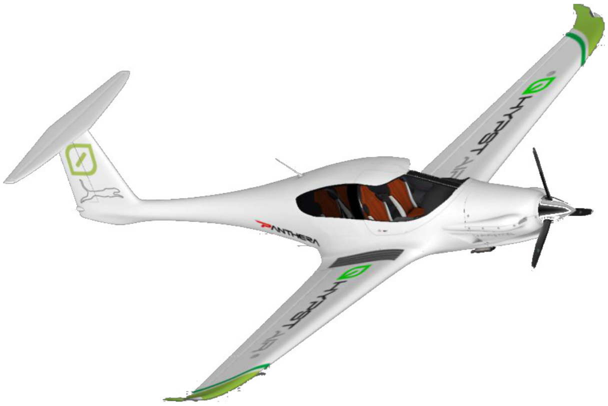


**Fig. 7** X-Plane model of the reference aircraft

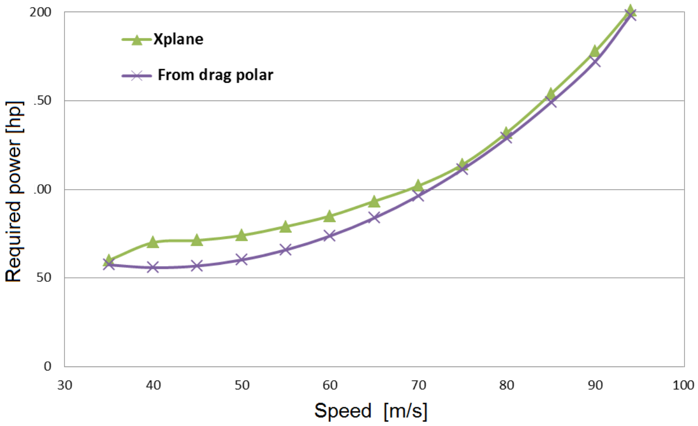


**Fig. 8** Required power VS speed

## 3 The *Performance Module*

Aerodynamic forces and other flight parameters are provided to the *Performance Module* by the *Flight Simulator* and used to compute the available energy in both fuel and batteries, in order to predict the remaining flight endurance.

The *Performance Module* has been developed by means of the Simulink software, implementing two independent blocks, the first one dedicated to the hybrid powertrain modelling and the second one for the endurance prediction.

The *Performance Module* is connected to the other modules as shown in Fig. **??**.

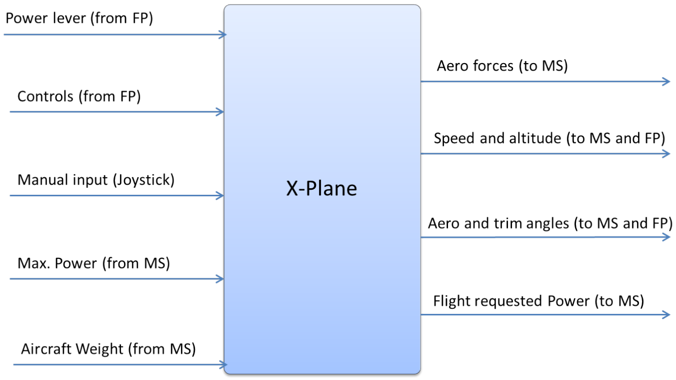

Power lever (from FP)
Controls (from FP)
Manual input (Joystick)
Max. Power (from MS)
Aircraft Weight (from MS)
X-Plane
Aero forces (to MS)
Speed and altitude (to MS and FP)
Aero and trim angles (to MS and FP)
Flight requested Power (to MS)

**Fig. 9** Data exchange in the *Flight Simulator* (XP=Flight Simulator, MS=Performance Module, FP=Flight Planner)

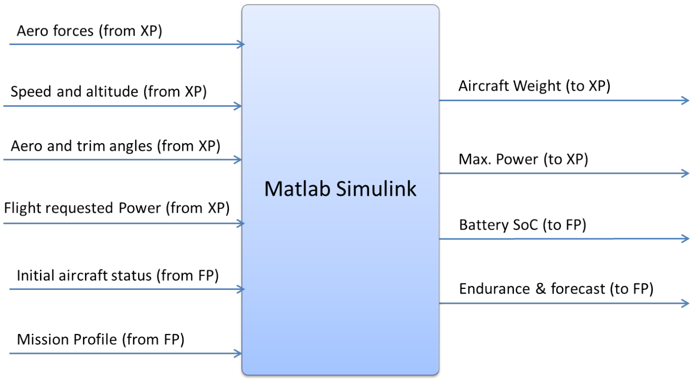

Aero forces (from XP)
Speed and altitude (from XP)
Aero and trim angles (from XP)
Flight requested Power (from XP)
Initial aircraft status (from FP)
Mission Profile (from FP)
Matlab Simulink
Aircraft Weight (to XP)
Max. Power (to XP)
Battery SoC (to FP)
Endurance & forecast (to FP)

**Fig. 10** Data exchange in the *Performance Module* (XP=Flight Simulator, MS=Performance Module, FP=Flight Planner)

## *3.1 Hybrid powertrain model*

The hybrid powertrain, which includes also the ICE and the propeller, has been modelled using the scheme shown in Fig. **??**. Since the maximum power provided by the brushless motor decreases with the batteries SoC, it is assumed that the rotating speed of the propeller is constant during the flight whereas the maximum torque can change.

In the first block, called “IN” in Fig. **??**, the input data coming from both the *Flight Planner* and the *Flight Simulator* are initialized and used as variables for the calculation. The main blocks are briefly described here after:

- *ICE*: in this block the efficiency of the ICE is computed depending on the flight altitude. The relation is based on the interpolation of experimental data provided by the ICE manufacturer ([**?**]).

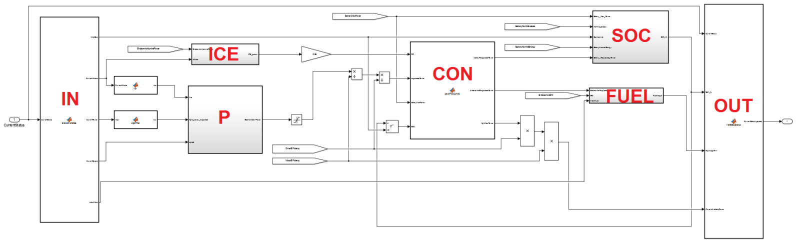


**Fig. 11** Simulink scheme of the Hybrid powertrain

- *P*: here the propeller efficiency is defined and, by applying the actuator disk theory and taking the flight conditions into account, the power demand to the electric motor is calculated.
- *CON*: this block simulates the control system which manages the available power through proper control laws.
- *SOC*: once the instantaneous power request to batteries is known, the SoC of the batteries is calculated in this block thorough an energy balance, in which internal losses are taken into account.
- *FUEL*: given the specific fuel consumption and the power required to the ICE, the fuel consumption is calculated.

At the present stage of development, all the electric components (generator, motor, inverters) are modelled by means of constant gains, representing the efficiencies, which can be modified during the initialisation. In a similar way, the reference values for the batteries, such as maximum energy and initial SoC, are also set at the beginning of the mission.

The control block needs the instantaneous batteries SoC as input, that is computed in the battery block: thus, a closed loop is needed and an anticipator block is applied to the value of the SoC in order to synchronize the calculation.

The Simulink scheme is triggered with a value of $0.1$ seconds in order to properly update the aircraft status.

## 3.2 The predictor

The predictor has been conceived in order to estimate at each time step the remaining flight endurance. The prediction is performed taking the amount of available energy (fuel and batteries) and the reference mission defined during the initialization into account.

Mission parameters and aircraft status are used as input of the predictor, which is made of two blocks, as Fig. **??** shows:

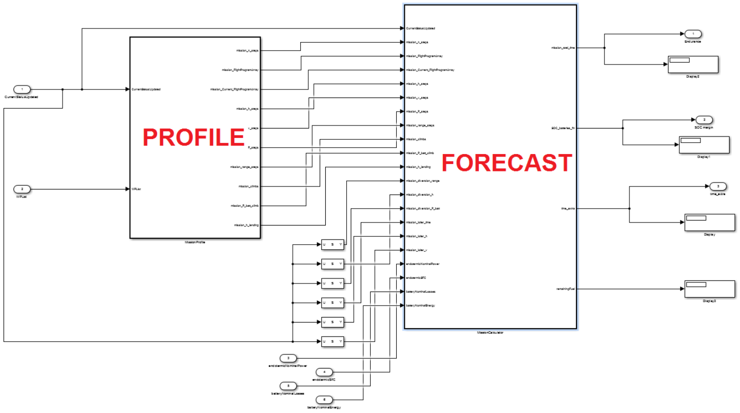


**Fig. 12** Simulink scheme of the performance predictor

- *PROFILE*: here the mission profile is divided into flight segments: climb, cruise, descent, landing, diversion and loiter. Each segment is defined entirely by the parameters listed in Table **??** and extrapolated from the *Flight Planner*.
- *FORECAST*: in this block, the fuel consumption and the battery discharge are calculated for the mission defined in the previous steps, assuming the flight programs reported in Table **??**.

Although the energy required to perform the emergency segments (diversion and loiter) is considered in the mission energy balance, the predicted endurance does not include the time needed to fly over such segments, hence the endurance prediction is always conservative.

In the *FORECAST* block, the amount of fuel required to complete the mission ($W_{f_{req}}$) is calculated and compared with the fuel available on the aircraft at the given time step ($W_f(t)$). Then, two conditions are possible:

a. $W_f(t) > W_{f_{req}}$: the reference mission can be accomplished with some safety margin, which is indicated to the pilot as an additional flight endurance ($t_{extra}$);
b. $W_f(t) < W_{f_{req}}$: the reference mission cannot be accomplished and a negative extra flight time is provided as output to the pilot, together with a warning message.

In addition to these results, the calculation block returns also the final SoC that the batteries are expected to have at the end of the mission.

The prediction block is updated continuously during the mission in order to take possible external interferences during flight (e.g wind or manual input) or possible modifications to the mission during the simulation into account. In this case the trigger frequency is lower because of the low computational speed of this block; since this calculation does not interfere with the other ones, this different frequency is considered acceptable.

## 4 The *Flight Planner*

The flight planner is an in-house developed software which is used for several purposes in `HyPSim`:

- to act as an autopilot, allowing to perform the given mission profile accurately (the pilot can always change the aircraft trajectory manually through the joystick);
- to allow the data exchange between the *Flight Simulator* and the *Performance Module*;
- to initialize the simulation, defining the initial status of the aircraft and the reference mission;
- to extract and visualize the results.

The main input/output data managed in the *Flight Planner* are reported in Fig. **??**

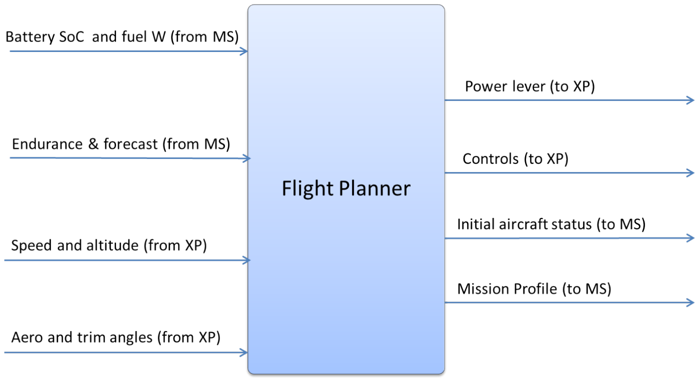

Battery SoC and fuel W (from MS)
Endurance & forecast (from MS)
Speed and altitude (from XP)
Aero and trim angles (from XP)
Flight Planner
Power lever (to XP)
Controls (to XP)
Initial aircraft status (to MS)
Mission Profile (to MS)

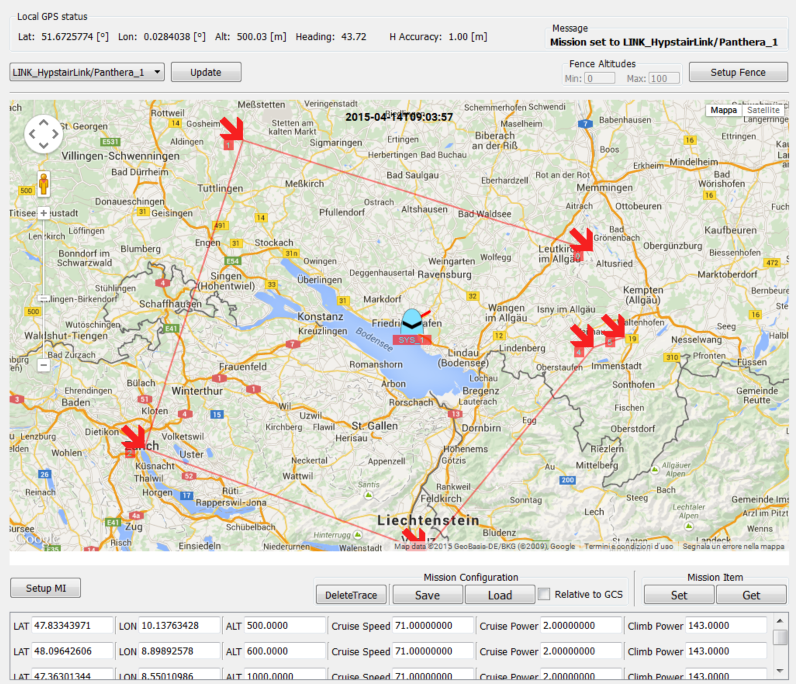
**Fig. 13** Data exchange in the *Flight Planner*

The *Flight Planner* interface consists of a plugin manager which allows to launch different software modules in a customizable layout. The main modules are:

- a link module, which allows the communication between the different software of `HyPSim`;
- the map plugin, which is used to display the position, the direction and the trajectory of the aircraft on a map as well as to define the mission profile (Fig. **??**);
- the *Aircraft Management Module*, shown in Fig. **??**, which allows to manage the aircraft during the flight simulation.

The mission profile can be defined by providing the waypoint list shown in the bottom part of Fig. **??**. The waypoints are defined through following values:

- latitude (*LAT*) and longitude (*LON*), whose values can be written in the related fields or provided by clicking on the map;



Lat: 51.6725774 [°] Lon: 0.0284038 [°] Alt: 500.03 [m] Heading: 43.72 H Accuracy: 1.00 [m]

Mission set to LINK_HypstairLink/Panthera_1

LINK_HypstairLink/Panthera_1

Min: 0 Max: 100

2015-04-14T09:03:57

☐ Relative to GCS

| LAT | LON | ALT | Cruise Speed | Cruise Power | Climb Power |
|---|---|---|---|---|---|
| 47.83343971 | 10.13763428 | 500.0000 | 71.00000000 | 2.00000000 | 143.0000 |
| 48.09642606 | 8.89892578 | 600.0000 | 71.00000000 | 2.00000000 | 143.0000 |
| 47.36301344 | 8.55010986 | 1000.0000 | 71.00000000 | 2.00000000 | 143.0000 |

**Fig. 14** Mission profile definition in the Map Plugin

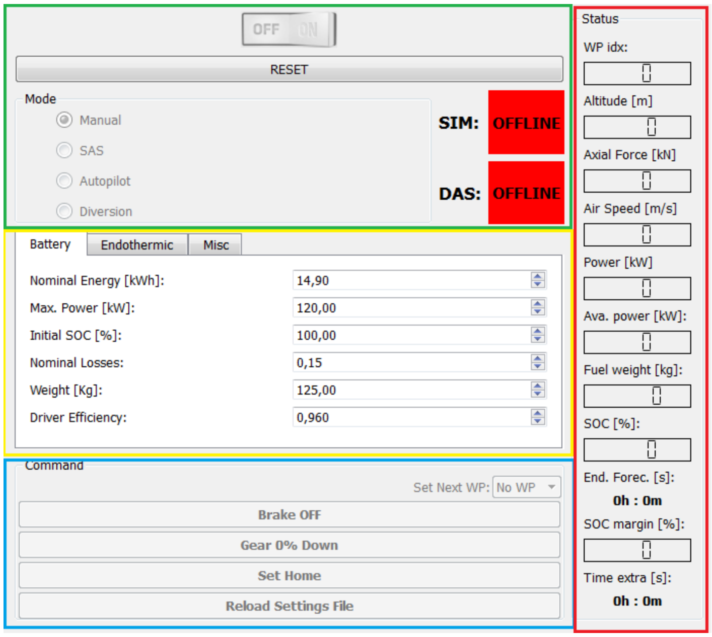
- altitude [m] (*ALT*);
- cruise speed [m/s] or cruise power [kW], which is neglected if cruise speed is assigned;
- climb power [kW], which is used when the altitude of the following waypoint is higher than the previous one.

The mission can be modified during the flight simulation by moving the waypoints on the map or modifying the parameters in the list; after any modification, the "Set" button has to be clicked to make them active.

Some comments on both the initial and the last parts of the mission are remarked in the following points:

- the take-off point is not included in the waypoint list; the simulator recognizes whether the aircraft is on the ground and an automatic take-off procedure is performed in order to reach the first waypoint.
- two waypoints are required in the end of the list in order to perform an automatic landing: the first one is used to define the landing point and the second one provides the runway direction;
- diversion and loiter segments are defined through the *Aircraft Management Module*, hence waypoints are not required.

## *4.1 The Aircraft Management Module*



Mode
● Manual
○ SAS
○ Autopilot
○ Diversion

SIM: OFFLINE
DAS: OFFLINE

Battery | Endothermic | Misc

Nominal Energy [kWh]: 14,90
Max. Power [kW]: 120,00
Initial SOC [%]: 100,00
Nominal Losses: 0,15
Weight [Kg]: 125,00
Driver Efficiency: 0,960

Set Next WP: No WP

Status
WP idx: 0
Altitude [m] 0
Axial Force [kN] 0
Air Speed [m/s] 0
Power [kW] 0
Ava. power [kW]: 0
Fuel weight [kg]: 0
SOC [%]: 0
End. Forec. [s]: 0h : 0m
SOC margin [%]: 0
Time extra [s]: 0h : 0m

**Fig. 15** The *Aircraft Management Module* window

The *Aircraft Management Module*, shown in Fig. **??** is compose of the following parts:

- the *Simulation Control* section (green box);
- the *Power Control* section (yellow box);
- the *Commands* section (blue box);
- the *Status* section (red box);

### 4.1.1 Simulation control section

Once the connections of the *Flight Planner* with the *Flight Simulator* and the *Performance Module* are active (green color in *SIM* and *DAS* boxes respectively), the *On/Off* button can be turned on in order to control the simulation. The following options can be activated:

- *AUTO*: the flight is controlled directly by the *Flight Planner* according to the mission profile defined in the map plugin;
- *MANUAL*: stick and throttle are manually controlled by means of a joystick;

- *DIVERSION*: the flight is automatically controlled and in addition the aircraft follows the path defined for the diversion and loiter.

When the autopilot mode is on, the *Flight Planner* manages the flight simulation directly. Most of the climb and cruise parameters can be inserted directly in the map plugin or in the aircraft management module, whereas the other flight segments (take-off, descent, diversion, loiter) are managed by means of a *setting file* that must be loaded before starting the simulation control.

### 4.1.2 Power control section

The power sources can be defined through the interactive yellow box in the center of the *Aircraft Management Module*. The data that must be provided are divided into three panels:

- *Battery panel*: required inputs concern the batteries characteristics (weight, maximum storable energy, initial SoC, etc.) and the internal losses of the electric propulsion components (propeller is not included);
- *Endothermic panel*: required inputs concern the nominal power, the efficiency and the Specific Fuel Consumption of the ICE, as well as data on the initial embarked fuel;
- *Misc panel*: required inputs are options about the power management during take-off (e.g.: power provided by batteries, etc.).

### 4.1.3 Commands section

When the autopilot is active, some commands can be managed using the *Commands* section in the bottom part of the interface.

Defined the reference mission as a waypoint list, the pilot can change the flight plan by selecting which waypoint has to be reached first (*Set next WP* button) or flying manually. In this latter case, by enabling again the autopilot mode, the *Flight Planner* automatically recognizes the nearest waypoint as the first one to be reached and the mission is then performed from that point ahead following the list.

Finally, the *Reload Setting File* button allows to modify the aircraft flight parameters and the control laws of moveable surfaces. Such file provides the following settings:

- take-off is performed at the maximum nominal power, with a given gain for rudder control in order to compensate the propeller torque effect;
- descent is performed with a given power throttle level;
- diversion and loiter parameters refer to the minimum power and maximum endurance conditions respectively;
- landing is performed with given speed, flap deflection and providing the runway altitude.
- manoeuvring limitations (e.g.: maximum bank angle)

### 4.1.4 Status section

The *Status* section is dedicated to the real-time visualization of the main flight parameters; the first field (*WP idx*) refers the identification number of the waypoint which the aircraft is heading to, whereas the other fields provide instantaneous data on the hybrid aircraft performance.

Finally, data resulting from the prediction model are visualized: the predicted endurance (*End. Forecast*), the estimated SoC at the end of the mission (*SoC margin*) and the difference, positive or negative, between the expected flight time and the time required to complete the mission (*Time extra*).

## 5 The Human-Machine Interface (HMI)

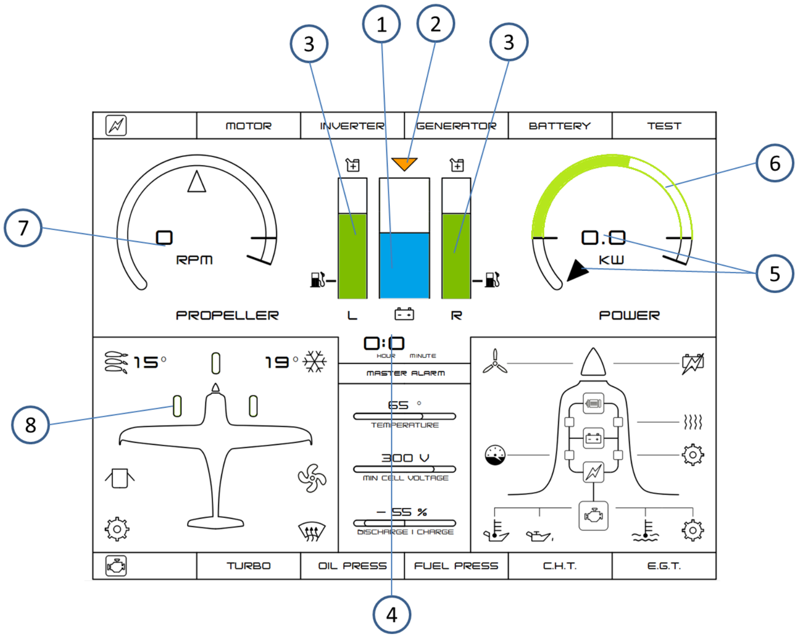


**Fig. 16** The HMI panel

Since part of the HYPSTAIR project has been dedicated to the development of a dedicated Human-Machine Interface ([**?**]), the simplified HMI panel shown in Fig. **??** has been implemented in `HyPSim` in order to display the following information:

1. battery SoC;

2. discharging/charging state: the triangle is green and rotated upward during charge or yellow and rotated downward in discharge;
3. fuel amount in left and right tanks;
4. remaining flight time in hours and minutes, as the sum of mission time and extra time (if this latter is negative, the time is visualized in orange colour in order to create a warning for the pilot);
5. power consumption: the instantaneous required power is displayed through both numbers and a pointer which moves along the green arch;
6. available power, represented through the empty green arch, whose length changes if batteries are fully discharged or the ICE is switched off;
7. propeller revolutions per minute (RPM);
8. landing gear position: green if extracted, empty otherwise.

## 6 Simulator testing

Several simulations have been conducted in order to assess the accuracy of the performance models implemented in `HyPsim`: the first test campaign has been focused on required power evaluation, whereas the second one has been carried out in order to study a critical condition in which batteries are fully discharged.

### *6.1 Required power evaluation*

The simulator has been first tested by assigning the mission profile shown in Fig. **??**, in which two level flight phases, at 200 m and 1000 m, have been performed varying the speed from 55 m/s to 85 m/s with a step input given to the throttle.

Fig. **??** shows the required power calculated by the simulator, whose positive and negative peaks are due to the accelerations and decelerations of the aircraft. In fact, according to Equation **??**, the required power can be decomposed in three contributions: the first one associated to aerodynamic drag $D$ (speed and altitude are considered constant), the second due to altitude variation on a constant slope ($\gamma$) trajectory and the third one due to speed variations ($dV/dt$).

$$P_{req} = V \cdot D + V \cdot W \cdot \sin\gamma + V \cdot \frac{W}{g}\frac{dV}{dt} \tag{1}$$

In this case, positive peaks are due to the accelerations introduced at each step of the $V(t)$ input function, whereas negative peaks indicate that the aircraft is decelerated, hence $P_{req}$ is set to 0.

The required power is multiplied by the efficiencies of all the powertrain components in order to calculate the power demanded to both ICE generator and batteries.

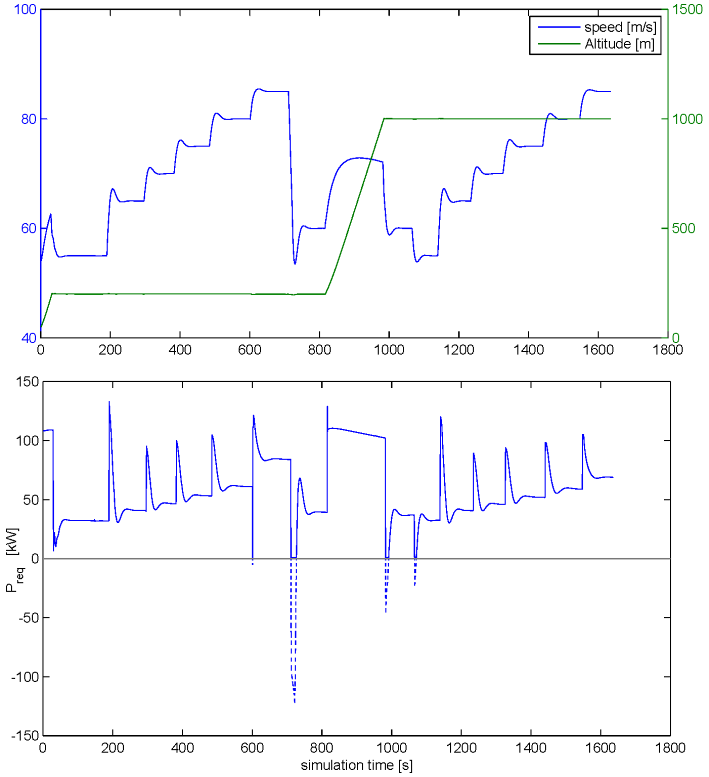


**Fig. 17** Flight speed and altitude of the input mission (top) and required power output (bottom)

Fig. **??** shows the comparison between required power for flight and the power demand to energy sources as provided by the *Flight Simulator*.

The dashed grey line in Fig. **??** is the maximum power provided by ICE generator, hence this chart allows to define the batteries charge and discharge phases. Therefore the SoC chart has been obtained, observing that a big discharge (about 30%) is needed to perform the climb from 200 m to 1000 m, whereas during level flight charge and discharge phases alternate depending on speed variations (after taek-off SoC has been limited to 90%).

### *6.2 Fully discharged batteries*

During this simulation campaign, the plane has been set in level flight conditions with a constant speed of 80 m/s, in such a way the batteries are continuously discharged until the minimum SoC threshold, set to 4%, is reached. In such condition

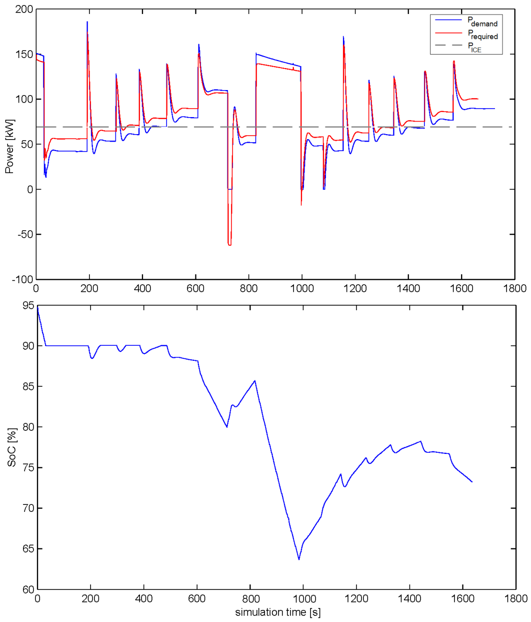


**Fig. 18** Required power for flight and power demand to energy sources (top) and batteries SoC (bottom)

the only available energy source is the ICE generator, which is assumed to provide a constant power of 80 kW.

With the aim of evaluating the flight performance when the available power is limited, a simulation has been performed using the *Flight Planner* in automatic mode in order to force the aircraft to fly at 80 m/s although the available power is not sufficient.

As Fig. **??** shows, when SoC reaches its lower limit the available power is instantaneously reduced to 80 kW and the aircraft speed decreases until the required power becomes lower than the available one. When this happens, the batteries begin charging and as soon as the SoC becomes higher than 4% the available power is restored to the maximum value, which brings the autopilot to increase aircraft speed up to 80 m/s. Hence, batteries are discharged again and such cycle is repeated creating an oscillating behaviour which can have negative consequences on both batteries health and flight dynamics.

It has been observed that such oscillations can be avoided by adding a second SoC threshold of 7%, below which the battery charge is not activated. The introduction of this additional threshold changes the power profile as illustrated in Fig. **??**, in which oscillations can be still observed but the frequency is much lower and the effects on flight dynamics are reduced.

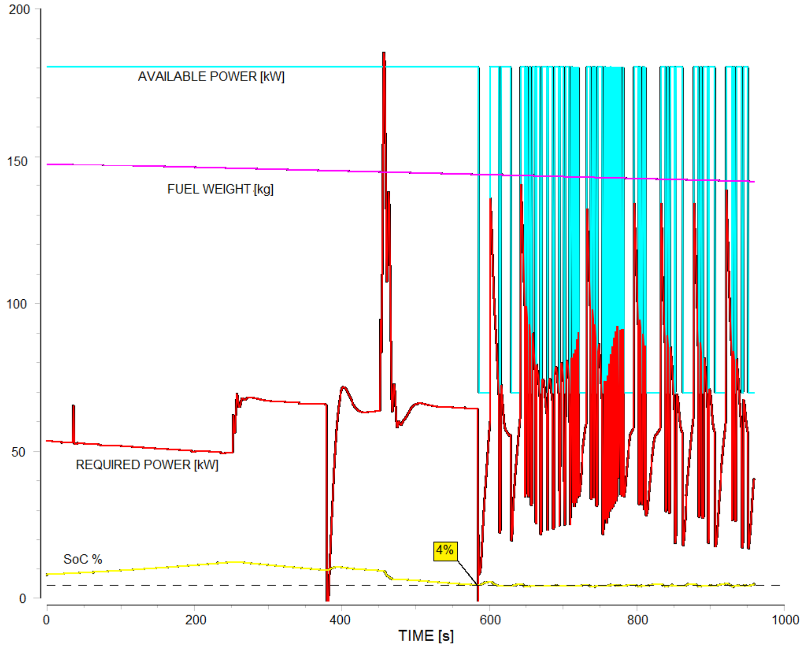


**Fig. 19** Simulation of fully discharged batteries

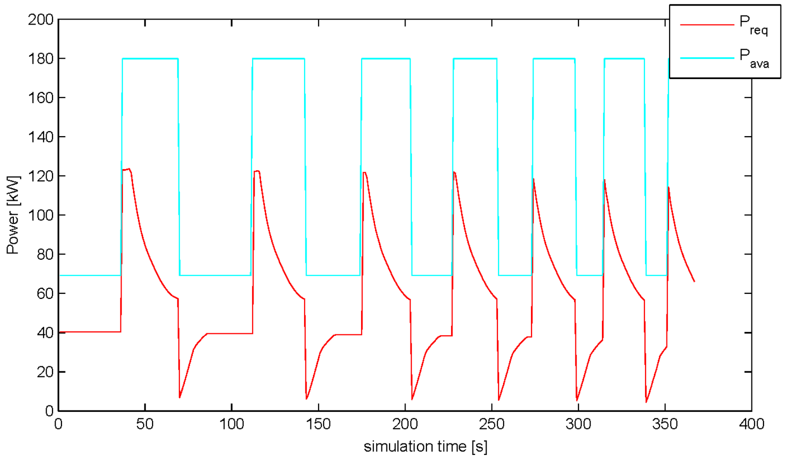


**Fig. 20** Power function after the introduction of a second threshold on SoC

## 7 Conclusions

The activities presented in this paper have been part of the European project called *HYPSTAIR*, concerning the development and validation of hybrid propulsion system components and sub-systems for electrical aircraft.

In particular, the development of a hybrid plane simulator, called `HyPSim`, have been described focusing on the software architecture and the functionality of such a simulation tool.

The main modules which compose the simulator are a commercial *Flight Simulator* (X-Plane), with which the aircraft geometry and aerodynamics have been modelled, a *Performance Module* developed in Simulink and used to simulate the hybrid powertrain, calculate the energy consumption and predict the flight endurance, and a in-house developed *Flight Planner* which allows to define the mission profile, select the flight mode (manual, autopilot, etc.) and allow the data exchange between all the modules. The simulator, in addition, can provide the main output using the Human-Machine Interface developed in the *HYPSTAIR* project.

The accuracy of the simulator in evaluating the energy consumption has been verified by comparing the required power for flight with experimental results provided by the aircraft manufacturer. In addition, specific mission profiles have been given as input and positive results on the reliability of the power demand evaluation have been achieved.

Finally, it has been observed that for some peculiar conditions, such as the case of fully discharged batteries, additional control logics must be implemented in order to avoid divergence phenomena.

As a general conclusion, `HyPSim` is a simulation tool able to achieve the several purposes for which it has been conceived, allowing to simulate any kind of mission profile taking also the human-in-the-loop factor into account. Moreover, the simulator is a practical tool for dissemination purposes.

Further development can be focused on the following aspects:

- implementation of additional control logics for off-design conditions;
- deeper and more complete implementation of the HMI module in the simulator;
- integration with haptic input devices developed within the *HYPSTAIR* project [**?**];
- development of more detailed models for the powertrain simulation.

**Acknowledgements** The research has been carried out within project "Development and validation of hybrid propulsion system components and sub-systems for electrical aircraft" (HYPSTAIR), which was co-funded by the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 605305.

## References

1. Oliviero F., Cipolla V., *HYPSTAIR Project Deliverable D2.1: Preliminary design of a serial hybrid aircraft*. Internal report of the Hypstair project. University of Pisa, March 2014
2. Oliviero F., Cipolla V., *HYPSTAIR Project Deliverable D2.2: Preliminary design of a serial hybrid aircraft*. Internal report, University of Pisa, May 2015
3. Laminar Research ind. *Plane maker for X-Plane 10 App Manual*, March 2013.
4. Rotax Aircraft Engines *Operators Manual for Rotax 912*, April 2014.

5. Ferracci M., Barlocchetti S., *HYPSTAIR Project Deliverable D3.1: HMI design*. Internal report, MB Vision, February 2014
6. Hace A., Golob M., *HYPSTAIR Project Deliverable D3.3: Haptic Interface*. Internal report, University of Maribor, June 2015