3° Convegno Tecnico Scientifico
Torino 3, 4, 5 Novembre 2003

Progetto e Sviluppo di Simulatori di Volo per Applicazioni Didattiche

Barbara Pralio
Politecnico di Torino - DIASP
barbara.pralio@polito.it

Fulvia Quagliotti
Politecnico di Torino - DIASP
fulvia.quagliotti@polito.it

Andrea Sanna
Politecnico di Torino - DAUIN
andrea.sanna@polito.it

ABSTRACT
Il presente lavoro, oggetto di collaborazione fra i Dipartimenti di Ingegneria Aeronautica e Spaziale e di Ingegneria Automatica ed Informatica del Politecnico di Torino, ha avuto come obiettivo principale il progetto e lo sviluppo di un simulatore di volo per applicazioni didattiche. L'esperienza in tale ambito ha portato ad evidenziare gli aspetti critici di un sistema che deve essere, innanzitutto, indirizzato ad un utente con scarse conoscenze di pilotaggio ed applicato a strumenti hardware di fascia bassa nonché a software non proprietario disponibile per varie piattaforme. Tale lavoro è stato affrontato prestando particolare attenzione a due aspetti chiave della progettazione: la portabilità del sistema e la semplicità di utilizzo dello strumento sviluppato.

Introduzione

Gli strumenti di simulazione e realtà virtuale hanno incrementato il loro utilizzo ed attirato crescente attenzione in molteplici campi di ricerca e applicazione, quali l'addestramento, l'intrattenimento e, non da ultimo, la didattica. Nell'ambito dell'attività didattica assume fondamentale importanza tenere in considerazione aspetti chiave quali la necessità di impiego di hardware non specializzati di basso costo, spesso equipaggiati con software non proprietario disponibile per varie piattaforme.
Il presente lavoro propone il progetto e lo sviluppo di un simulatore di volo basico, destinato appunto all'impiego in ambito didattico, sfruttando le attrezzature dei laboratori sperimentali dipartimentali. L'obiettivo principale di tale attività consiste nel fornire uno strumento semplice per lo studio della risposta in manovra di velivoli appartenenti a categorie diverse, l'analisi dei modi propri del velivolo e la comprensione dei meccanismi di valutazione delle qualità di volo e delle caratteristiche di manovrabilità dell'aeromobile.
I punti chiave di tale progetto, che ha visto coinvolti due gruppi di ricerca del Politecnico di Torino – l'uno appartenente al Dipartimento di Ingegneria Automatica ed Informatica, l'altro appartenente al Dipartimento di Ingegneria Aeronautica e Spaziale – sono rappresentati dalla modellizzazione della dinamica del velivolo e dell'ambiente circostante e dalla realizzazione di uno scenario virtuale di base. Nell'ambito di tale attività gli obiettivi principali sono stati la portabilità del codice e la semplicità di utilizzo del prodotto finale. La portabilità del codice è stata ottenuta ricorrendo al

linguaggio di programmazione C++, permettendo così una facile ricompilazione in ambienti eterogenei, con minime modifiche del codice sorgente. Da questo punto di vista anche la libreria grafica OpenGL [1] ha rivestito un ruolo chiave, permettendo di raggiungere un buon livello di astrazione rispetto alle primitive geometriche utilizzate. La semplicità di utilizzo è stata invece perseguita limitando al minimo il numero di pacchetti aggiuntivi; in particolare, il solo pacchetto GLUT, peraltro facilmente reperibile ed installabile, richiede di essere aggiunto alle suddette librerie OpenGL. Alla luce dei campi di applicazione del software in oggetto, l'intera architettura della parte grafica è stata progettata nell'ottica di mantenere buone prestazioni anche su calcolatori di fascia bassa; questo aspetto ha, in particolare, condizionato il progetto della pipeline di *rendering*.

Struttura del simulatore di volo

Il simulatore di volo in oggetto è costituito, secondo quanto illustrato in Fig. 1, da un modulo rappresentativo della dinamica del sistema velivolo, caratterizzato da estrema versatilità e modularità; tale modulo interagisce con lo scenario virtuale, il cui scopo consiste nel ricreare l'ambiente circostante, fornendo un importante riferimento per le valutazioni dell'utente. Quest'ultimo, a sua volta, si interfaccia con il sistema inserendo i comandi di volo, e quindi gli ingressi del modello matematico, attraverso la tastiera, e scegliendo fra alcune impostazioni di visualizzazione dello scenario. Le informazioni relative al comportamento del velivolo vengono registrate sotto forma di *time-history* di alcuni parametri fondamentali del volo attraverso un'interfaccia grafica, che permette l'analisi in tempo reale. La comprensione della risposta del velivolo alla manovra viene ulteriormente supportata dalla presenza di un panello strumenti basico, il cui scopo è insegnare all'utente a familiarizzare con le informazioni degli strumenti caratteristici di un *cockpit* reale.

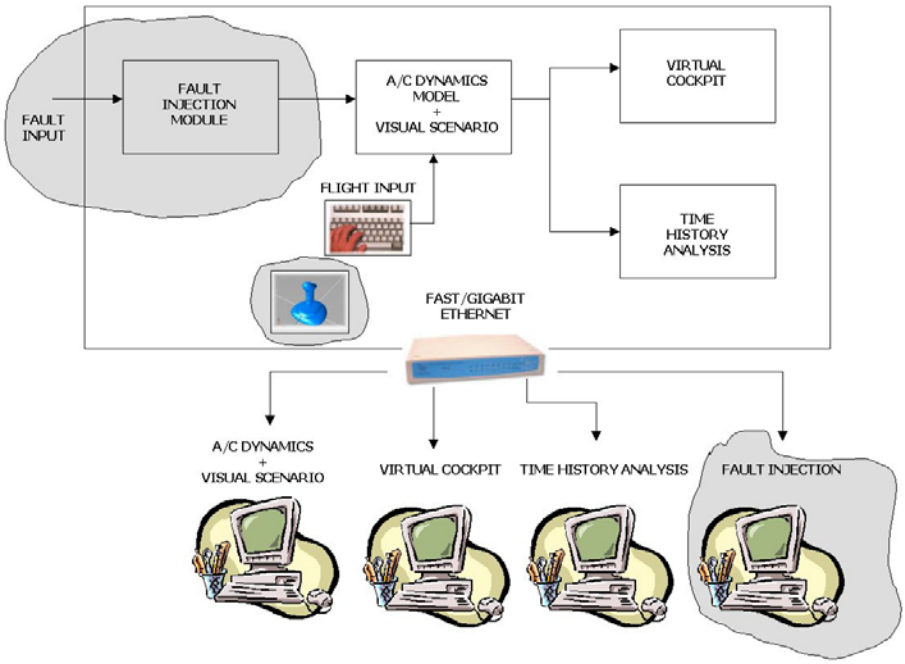

Fig. 1: Struttura del simulatore di volo.

Modello matematico del velivolo
Il modello matematico del velivolo è costituito da un corpo principale in linguaggio FORTRAN che si interfaccia con il codice di modellizzazione dello scenario attraverso un'interfaccia in linguaggio C++. Il modello matematico si suddivide nei seguenti moduli:

-) il modello matematico completo della dinamica del velivolo;
-) il modello matematico del sistema propulsivo;
-) il modello matematico della turbolenza atmosferica.

Il primo modulo implementa un sistema di equazioni complete della dinamica del velivolo, inteso come corpo rigido a sei gradi di libertà, basato sulla trattazione classica della meccanica del volo atmosferico [2]. Il codice di calcolo permette inoltre di considerare il modello matematico non lineare completo, secondo quanto teorizzato da Tobak e Schiff [3], rendendo possibile lo studio della dinamica di velivoli ad elevata manovrabilità. La caratterizzazione del velivolo in oggetto avviene fornendo i dati geometrico-inerziali e il database aerodinamico. Il codice di calcolo in questione permette, inoltre, l'applicazione di un modello basato sui quaternioni, che permette di svincolarsi dai problemi che sorgono in manovre ad elevati angoli di assetto, evitando le singolarità che si manifestano applicando il modello basato sugli angoli di Eulero.
Il codice è strutturato in modo tale da effettuare la ricerca delle condizioni di equilibrio come primo passo della procedura di simulazione. L'utente ha il compito di impostare quota, velocità, angolo di rampa e condizioni di manovra; il codice effettuerà innanzitutto la ricerca delle condizioni di equilibrio attraverso l'applicazione di un metodo iterativo Netwon-Raphson. L'integrazione delle equazioni del moto è invece affidata ad un integratore esplicito Runge-Kutta del quarto ordine, basato sul metodo di Gill; la scelta di un integratore esplicito è stata dettata della necessità di ridurre i tempi di calcolo, rendendoli il più possibile compatibili con una simulazione in tempo reale.
Per quanto riguarda, invece, la modellizzazione del sistema propulsivo, il codice di calcolo è strutturato in modo tale da permettere la simulazione sia di velivoli motoelica sia di velivoli turbogetto. Nel caso di velivolo motoelica, il codice effettua il calcolo della spinta fornita dall'elica e dei momenti di beccheggio e di imbardata dovuti alla non uniforme distribuzione di trazione sulle pale. La valutazione delle caratteristiche dell'elica viene affidata alla teoria dell'elemento di pala.
La modellizzazione della turbolenza atmosferica è stata introdotta al fine di aumentare il grado di attendibilità e di realismo della simulazione e di permettere all'utente una valutazione, seppure indicativa, degli effetti del volo in aria non calma. Tale modulo si basa sul modello di turbolenza atmosferica di Dryden [4] [5], in uso nella normativa MIL STD-1797A. L'utente è quindi in grado di scegliere se effettuare o meno il volo in aria calma e di selezionare l'intensità dell'eventuale raffica da introdurre, oltre che la sua durata nell'ambito della simulazione.

Scenario virtuale
L'intera architettura della parte grafica è stata progettata nell'ottica di mantenere buone prestazioni anche su calcolatori di fascia bassa, al cui utilizzo è finalizzata l'applicazione in ambito didattico. Per raggiungere tale obiettivo si è reso necessario adottare alcuni accorgimenti nella fase di progetto della "pipeline" di *rendering*. La parte più onerosa del motore di *rendering* è rappresentata dalla creazione e dalla gestione dello scenario virtuale, che, seppur elementare, è costituito da milioni di poligoni. L'individuazione, quindi, della parte di scenario immediatamente visibile dall'utente

permette di ridurre il numero di poligoni utilizzati e di non penalizzare così in modo e eccessivo le prestazioni. A questo proposito è stato introdotto un efficiente algoritmo di *frustum culling*, in grado di stabilire istante per istante quali vertici devono essere processati e quali possono essere trascurati. La forma del campo visivo della telecamera virtuale costituisce la base dell'algoritmo di *culling*, poiché, note le equazioni dei piani di taglio principali, permette, grazie a pochi controlli, di stabilire la "visibilità" di un determinato punto. Affidarsi semplicemente a tale algoritmo non permette comunque di ottenere prestazioni soddisfacenti, in relazione agli strumenti hardware a disposizione in ambito didattico. In fase di progetto si è quindi reso indispensabile organizzare la mappa in diversi settori, di grandezza fissa pari ad un quinto della dimensione totale. Ciascun settore è stato, a sua volta, suddiviso in aree più piccole, organizzate poi in un albero gerarchico (*octree*). Grazie al livello di organizzazione garantito da questa architettura, è stato possibile stabilire in maniera euristica quali parti della mappa dovessero essere processate per il *rendering*. L'intera fase di riproduzione dello scenario virtuale si riduce così ad una verifica in profondità dell'albero, controllandone di volta in volta la visibilità dei nodi ed eliminandone opportunamente intere sezioni.

Lo scenario virtuale, la cui rappresentazione grafica, come detto in precedenza, è stata appositamente semplificata per le esigenze di "leggerezza" del codice, consente di visualizzare paesaggi definiti tramite mappe di altezze, su cui vengono aggiunte apposite *texture*. Nell'ottica di incrementare il realismo del "mondo virtuale" e di creare alcuni punti di riferimento durante l'esecuzione del volo, sono state introdotte strutture aggiuntive, quali edifici ed una pista aeroportuale (Fig. 2); tale scelta è stata portata avanti guardando, ancora una volta alla semplicità dell'architettura ed al compromesso fra realismo e prestazioni.

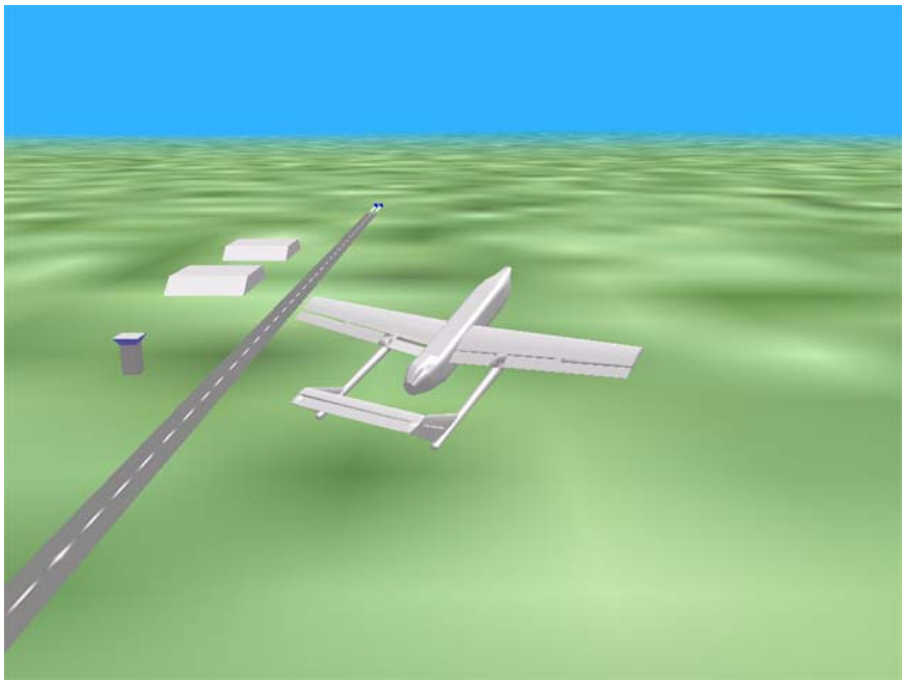

Fig. 2: Particolari dello scenario virtuale.

Ai fini di una miglior comprensione delle variabili in gioco e del comportamento del velivolo durante le manovre, è prevista la visualizzazione di un *head-up display* semplificato (Fig. 3), contenente le sole informazioni ritenute essenziali per il volo. In particolare, allo stato attuale, vengono visualizzati i valori istantanei di quota, velocità, assetti di volo e comandi impartiti al velivolo.

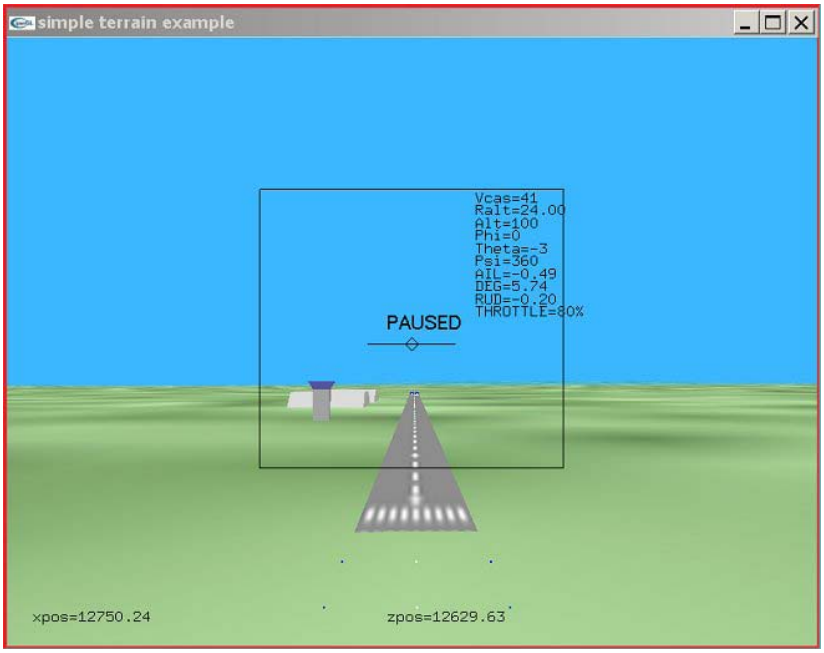

Fig. 3: Particolare dell'HUD.

Il simulatore di volo in oggetto annovera fra le sue caratteristiche la possibilità di fornire una visuale interna al velivolo (in cui la posizione del pilota risulta specificabile in ingresso) oppure una visuale esterna, a totale discrezione dell'utente; quest'ultima ha l'importante vantaggio di permettere una migliore comprensione degli assetti del velivolo in risposta ai comandi ad esso impartiti. Poiché la libreria grafica OpenGL non mette a disposizione primitive grafiche in grado di descrivere un solido complesso come un velivolo completo e il realismo del sistema dipende anche dalla qualità del modello inserito, il modello tridimensionale che descrive il velivolo deve essere fornito in ingresso dall'utente (fig. 4); a questo proposito è stata opportunamente creata una classe in grado di caricare il file con il formato adeguato. Il modellatore scelto a riferimento nell'ambito di questo progetto è 3D Studio Max.

Interfaccia grafica per analisi dati simulazione

Al fine di rendere pienamente comprensibile la risposta alla manovra del velivolo e di effettuare un'analisi dei dati di volo pressoché in tempo reale rispetto alla simulazione stessa, è stata realizzata un'interfaccia grafica in ambiente LabVIEW (Fig. 5), in grado di visualizzare mediante grafici l'evoluzione di una serie di parametri caratteristici durante la simulazione del volo. Lo scambio dati fra il PC responsabile della simulazione ed il PC utilizzato per la visualizzazione dell'interfaccia grafica utente è stato affidato al protocollo di comunicazione TCP/IP, che ha dimostrato, nella suddetta applicazione, prestazioni soddisfacenti.

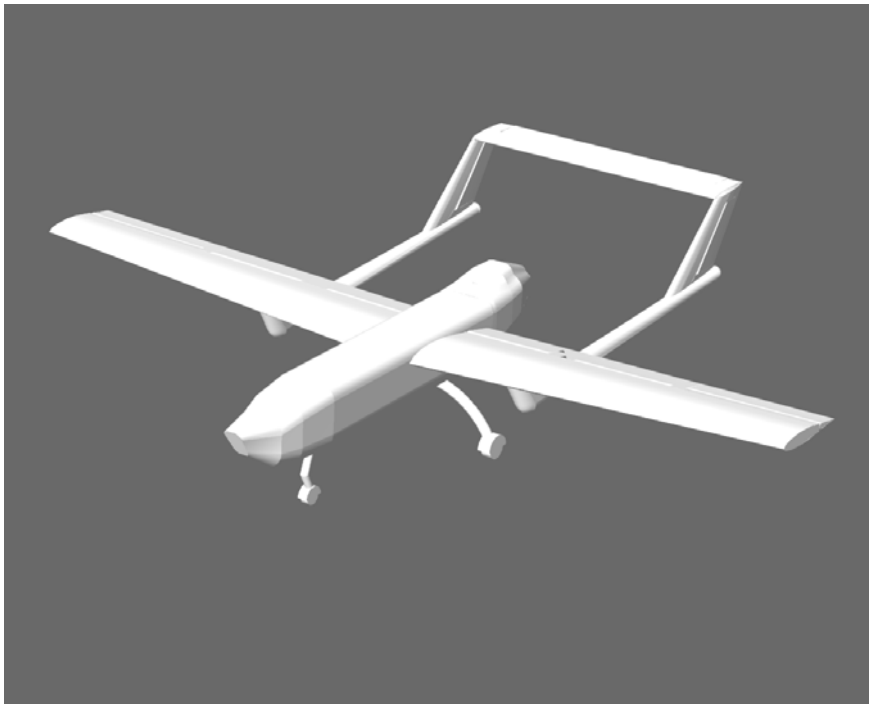

Fig. 4: Esempio di modello 3D caricato nel simulatore di volo.

Fra i dati di simulazione si è reso necessario scegliere alcuni parametri di importanza chiave per la corretta interpretazione del comportamento del velivolo. L'interfaccia grafica riporta, quindi, allo stato attuale, l'andamento temporale degli angoli aerodinamici α e β, degli angoli di Eulero Φ, Θ e Ψ e delle velocità angolari p, q ed r.
Diagrammi a parte sono dedicati alle variazioni delle deflessioni delle superfici mobili, la cui posizione ricopre così un ruolo di primo piano nell'attenzione dell'utente. Quest'ultima scelta è stata dettata principalmente dalla periferica dedicata al comando delle suddette superfici; in questa prima fase di implementazione del simulatore di volo, infatti, i comandi vengono impartiti esclusivamente tramite tastiera. La scarsa istintività del comando così effettuato ha quindi supportato la scelta di fornire un ausilio alla corretta operatività del sistema e nella fattispecie nel visualizzare su grafico la posizione delle superfici e l'entità della loro variazione. Fra le attività attualmente in fase di sviluppo va comunque annoverato l'inserimento nel simulatore di un sistema joystick ed eventualmente pedaliera, in grado di aumentare il livello di interattività dell'esperienza di simulazione di volo.

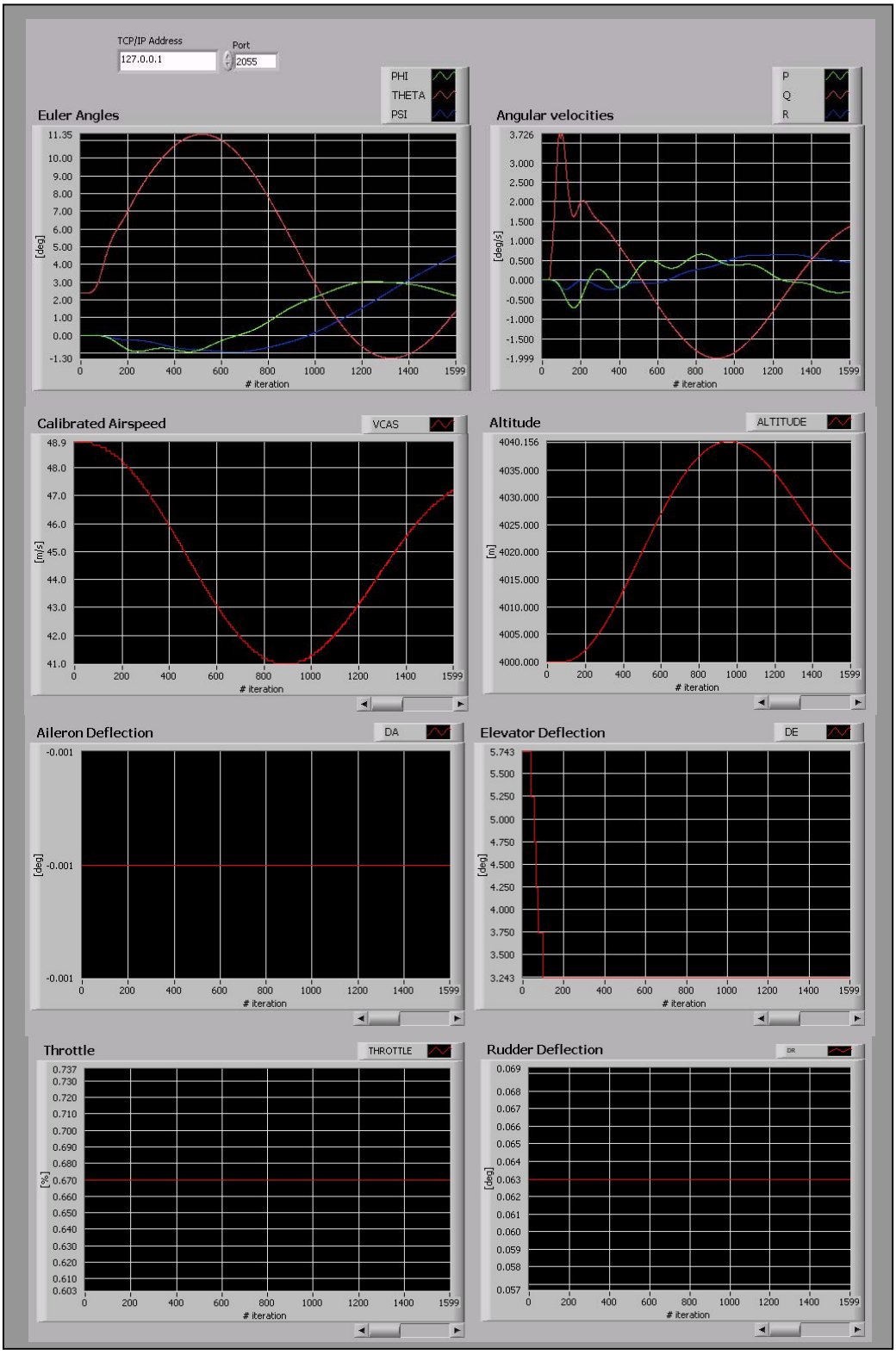

Fig. 5: Interfaccia di analisi dati real-time.

Pannello strumenti

La comprensione dei dati di volo viene poi supportata dalla presenza di un pannello strumenti basico, realizzato in ambiente LabView, che include alcuni degli strumenti classici di un vero cockpit (Fig. 6). Lo scambio dati con la simulazione si basa sul medesimo protocollo di comunicazione che permette la visualizzazione degli strumenti di analisi dei dati. In questo stadio preliminare di sviluppo, il simulatore di volo è corredato di un pannello strumenti costituito da altimetro, anemometro, variometro, orizzonte artificiale e coordinatore di virata. Poiché il loro sviluppo non rientra negli interessi dei gruppi di ricerca coinvolti nel presente progetto, gli strumenti più complessi (di realizzazione più complessa), quali orizzonte artificiale e coordinatore di virata, sono stati inseriti quali oggetti preesistenti.

Alla luce della finalizzazione del sistema allo studio degli aspetti correlati alla meccanica del volo, non è stata introdotta alcuna modellizzazione dell'impiantistica di bordo e dei sistemi correlati. Il *"cockpit" virtuale* riporta, quindi, i soli strumenti essenziali alla comprensione dell'interazione fra i comandi impartiti al velivolo e la sua risposta alla manovra, trascurando gli aspetti legati all'avionica ed alle comunicazioni.

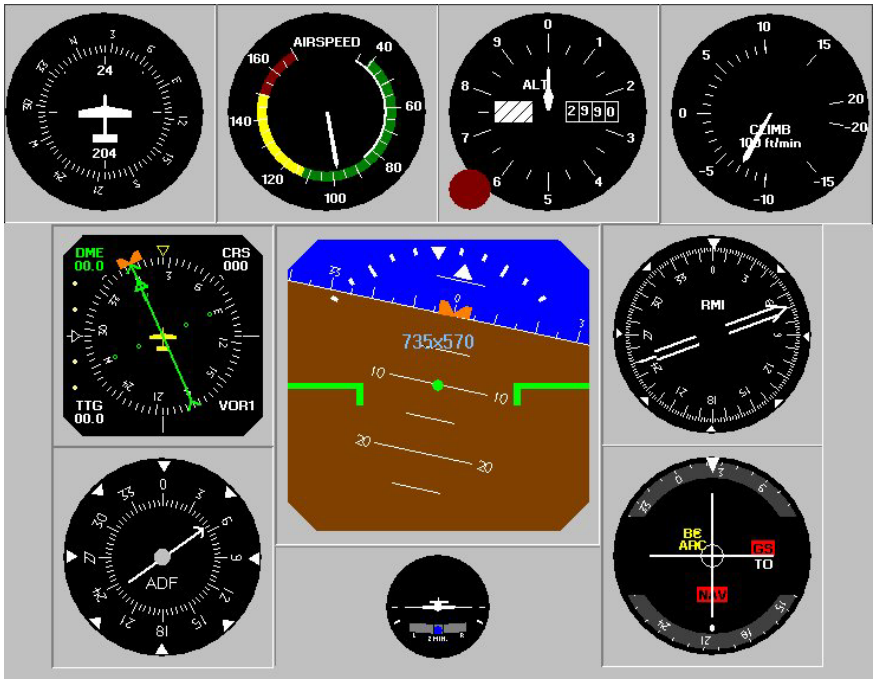

Fig. 6: Interfaccia grafica del pannello strumenti.

Valutazione delle prestazioni

Al fine di valutare l'efficacia dello strumento di simulazione in oggetto, si riportano i dati di alcune prove [6] eseguite su un calcolatore di fascia bassa, avente le seguenti caratteristiche:

Processore:	AMD K7 Thunderbird (1 GHz)
RAM:	512 MB (su bus da 133 MHz)
Scheda grafica:	Hercules 3D Prophet II MX

Processore grafico : NVIDIA GeForce2 MX – 32 MB DDR memoria video
Bus grafico: conforme allo standard AGP4x

Nella fase di sperimentazione sono stati utilizzati i sistemi operativi Linux – nella versione Red Hat 7.3 con kernel 2.4.18-3 e driver specifici per l'accelerazione grafica forniti dalla NVIDIA (versione 1.0-2123) – e Microsoft Windows XP Professional. Lo spazio occupato su disco è di circa 15 MB. Lo scopo principale delle prove eseguite sul codice è stato il confronto delle prestazioni sui vari sistemi operativi, rivolgendo particolare attenzione all'impatto della parte grafica sulle prestazioni complessive. A questo proposito alcuni risultati sono riportati in Tab. 1:

	Linux		Microsoft Windows	
	Vista Interna	Vista Esterna	Vista Interna	Vista Esterna
Frame per sec	25	15	32	22
Parte grafica	71%	81%	75%	83%
Parte matematica	28%	18%	23%	17%

Tabella 1: Valutazione delle prestazioni del codice di simulazione.

Queste prove hanno messo in evidenza il non trascurabile impatto del modello del velivolo e della sua visualizzazione tramite vista esterna sulle prestazioni finali del simulatore. Va, inoltre, osservato che durante la fase di prova sono stati visualizzati in media 30370 triangoli sui 130000 complessivi e questo ha avuto un'influenza notevole sul tempo di elaborazione. A tal proposito, al fine di migliorare ulteriormente le prestazioni su PC di fascia bassa, sarebbe utile aggiungere all'algoritmo di *frustum culling* e all'albero gerarchico (*octree*) un meccanismo in grado di adattare il livello di dettaglio in maniera dinamica, riducendo il carico di lavoro relativo al calcolo delle superfici tridimensionali.

Conclusioni ed attività future

Lo strumento di simulazione realizzato nell'ambito di tale attività presenta prestazioni soddisfacenti in relazione ai requisiti dell'applicazione finale. Lo scenario virtuale e gli strumenti di analisi dati sono stati interfacciati con il modello matematico del velivolo e la loro interazione è stata collaudata con una serie di prove di funzionamento che hanno dato buoni risultati in diverse condizioni di volo. Il sistema così ottenuto annovera fra le sue caratteristiche:

- la possibilità di intervenire sul modello matematico senza pesanti ingerenze sull'architettura;
- l'applicabilità a calcolatori di fascia bassa così come a piattaforme grafiche dedicate;
- l'estrema portabilità del codice;
- la facilità di installazione;
- l'intuitività dell'interfaccia.

Fra le attività future sono da annoverare:

- l'inserimento di alcuni comandi attraverso interfacce maggiormente intuitive, quali joystick ed, eventualmente, pedaliera;
- l'inserimento nel ciclo di simulazione di un sistema di *"fault injection"*, che permetta di valutare gli effetti di avarie, principalmente legate alle superfici mobili.

Bibliografia

[1] M. Woo, J. Neider, T. Davies, D. Shreiner, “OpenGL Programmino Guide, Third Edition – The Official Guide to Learning OpenGL, Version 1.2”, Addison Wesley Longman Inc.
[2] T. G. Gainer, S. Hoffman, “Summary of Transformation Equations and Equations of Motion Used in Free Flight and Wind Tunnel Data Reduction Analysis”, NASA SP-3070.
[3] M. Tobak, G. T. Chapman, and L. B. Schiff, "Mathematical Modeling of the Aerodynamic Characteristics in Flight Dynamics" NASA TM 85880, 1984.
[4] J. C. Yeager, “Implementation and Testing of Turbulence Models for the F-18 HARV Simulation”, NASA CR-1998-206937.
[5] MIL-STD 1797
[6] Borrione E., “Sviluppo di un’interfaccia grafica per un simulatore di volo didattico”, Tesi di Laurea, Luglio 2003.