Joint Strike Fighter

Flight Control Law Development for the F-35 Joint Strike Fighter

David W. Nixon
Lockheed-Martin Aeronautics

5 October 2004
F-35 Variants

**STOVL**
Integrated STOVL Propulsion System, Flying Qualities and Performance From Hover Through Supersonic Flight

**CTOL**
Flying Qualities, Engine-Inlet Compatibility, and Flight Performance at Representative Mission Points

**CV**
Carrier Suitable Flying and Handling Qualities and Flight Performance at Representative Mission Points
X-35A/B Features

Conventional Configuration

- Engine Bay Vent Static Inlet (Typ.)
- APU Inlet
- APU Exhaust
- LiftFan Inlet Doors (Activated - Commanded Closed)
- Air Refuel Receptacle
- Air Data Sensors
- ECS Ram Air Inlet
- ECS Ram Air Exhaust
- Roll Nozzle Aperture (Sealed)
- Aux. Inlet Doors (Activated - Commanded Closed)
- Cockpit Emergency Vent Inlet
- 3BSD Nozzle Doors (Activated - Commanded Closed)
- LiftFan Nozzle Doors (Activated - Commanded Closed)
- Engine Bay Vent Ram Inlet
X-35A/B Features

STOVL Configuration

- LiftFan Nozzle & Doors
- Roll Nozzle
- 3BSD Doors
- 3BSD
- Air Refuel Receptacle
- LiftFan Inlet & Doors
- Aux Inlet “Rabbit Ear” Doors & Louver Mechanism
CV Configuration

- LiftFan Nozzle Doors (Activated - Commanded Closed)
- Roll Nozzle Aperture (Sealed)
- 3BSN Nozzle Doors (Activated - Commanded Closed)
- AOA Approach Lights
- Simulated Air Refuel Probe
- Emergency Tail Hook
- LiftFan Inlet Doors (Activated - Commanded Closed)
- Aux Inlet Doors (Activated - Commanded Closed)
- Air Refuel Receptacle
- Ailerons

X-35C Features
Flight Control Objectives

• Leverage Advanced Control Design Methodology
  – Maximize Commonality in Control Laws Across the Variants
  – Enable Design-to-Flying Qualities Philosophy
  – Facilitate Rapid Updates to the Control Laws Throughout the Design Cycle

• Exploit Model-Based Software Development and Automatic Code Generation Technology
  – Singular Design Reference
  – Reduce Software Defects
  – Improve Cycle Time
Dynamic Inversion Control Law Structure

Flying Qualities Dependent
(How it should Fly)

Airframe/Engine Dependent
(Aero, Engine, Mass)

Isolate

Common Control Law Structure for All Aircraft Variants

Lockheed Martin Aeronautics Company
What is Dynamic Inversion?

- **Background**
  - *Initial Methodology Developed by Dr. Dale Enns (Honeywell Technology Center)*
  - *Honeywell/Lockheed Teamed on Multi-variable Control Research Program That Applied Methodology to F-16, YF-22, and F-117*
  - *Early STOVL Application During ASTOVL Program*

**Linear Aircraft Equations of Motion**

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
,cv &= Cx
\end{align*}
\]

- \(x\) - states
- \(u\) - effectors
- \(cv\) - control variable
- \(A\) - Aircraft Dynamics Matrix
- \(B\) - Control Effectiveness Matrix
- \(C\) - Control Variable Matrix

**Dynamic Inversion Formulation**

\[
\begin{align*}
\dot{cv}_{des} &= Cx = CAx + CBu \\
u &= (CB)^{-1}(cv_{des} - CAx)
\end{align*}
\]
• Map the Pilot Commands and Feedbacks into the Desired Aircraft Accelerations, not Aircraft Surface Commands

Roll Regulator Example

Pilot’s Roll Command

\[ \text{Cmd}_{\text{roll}} \]

\[ + \]

\[ 1/\tau_{\text{roll}} \]

\[ - \]

Roll Rate Feedback

\[ P_s \]

\[ \dot{P}_{s \text{ desired}} = \frac{1}{\tau_{\text{roll}}} \times (\text{Cmd}_{\text{roll}} - P_s) \]

\[ P_s = \frac{1}{\tau_{\text{roll}}} \]

\[ \text{Cmd}_{\text{roll}} \times (s + \frac{1}{\tau_{\text{roll}}}) \]

Simple Dynamic Inversion Roll Control Law Provides a Classical First Order Roll Response

Design goal embedded in control law

Pilot's Roll Command

Roll Rate Feedback

\[ \dot{P}_{s \text{ desired}} \]

\[ P_s \]

\[ 1/\tau_{\text{roll}} \]

\[ \text{Ps des} \]

63% Max

Time (sec)

Roll Rate (deg/sec)
Model-Based S/W Development Philosophy

- Single Electronic Source for All Software Requirements, Design, and Implementation
  - Graphical Representation of Software Design - No Paper Diagrams or Separate Block Diagrams
  - All Textual Documentation Embedded in Model

- Automatic Code Generation Process to Eliminate Coding Defects
  - Eliminate Errors Normally Incurred From Translating Requirements Into Design and Code

- Model Thoroughly Evaluated in Analytical and Simulation Environment
  - Code Supplied to Six DOF Simulation (ATLAS) for Dynamic Analysis and Piloted Simulator
  - Prototype Design Changes Rigorously Tested in Simulator with Test Pilots

Not Just A Higher Level Language for Programming – A Different Software Development Paradigm
Model-Based Development Process

Central Model
Simulink/Stateflow

MATLAB
Linear Analysis/Design

ATLAS
Non-linear Sim

CLAW
Linear Models

Gain Data

Flowdown Reqts (SRS)

Control Laws

Mode Logic

DOORS
Air System
Air Vehicle
Vehicle Systems
FCS

SIMS
Interface

Design Guides
• Flying Qual.
• Air Data Perf.

Embedded Software (OFP)

VMX OS
Built-In Test App
FCRM App
CLAW App (RTW)
Air Data App (RTW)

Actuators
Aero
Air Data
CLAW
Sensors
Engine

RTW/ERT C

Simulators

Simulink/Stateflow

Embedded Software (OFP)

Formal S/W Test

Lockheed Martin Aeronautics Company
Model-Based Software Products

- **Model-Based Process Requires a Re-interpretation of Traditional Software Products**
  - *Software Requirements are Combination of SRS Text & Diagrams*
  - *Software Design is Combination of SDD Text & Diagrams*
  - *Verification is Performed with SRS Text & Graphical Model*
  - *Requirements-to-Design Linkage is Inherent*
  - *SPEs are Performed on Graphical Model Instead of Code*
Where We Are

- Model-Based Design proven in CDA phase
  - Successful flight test of all variants with one OFP
  - Reduced Software Defects (Early Checkout in Engineering Simulations)
  - Overall Reduction in Manhours/SLOC of ~40%

- Fully functional UA control laws and Air Data in Simulink
  - CLAW model is very large
    - consists of root model + 266 library files
    - Root model has 421 inputs and 337 outputs
    - 16,143 blocks in 871 subsystems
    - 998 instances of reused utility subsystems
    - Real-Time Workshop® ERT code is ~47,000 logical lines of code in 750 files
  - CLAW and Air Data code is running in offline simulation, handling qualities simulator, and on target hardware on test stations

- MathWorks support has been a key element in overcoming obstacles
  - R13SP1
  - R14SP1
Challenges

- Automated testing to meet Safety-critical test requirements
  - T-VEC
  - Running ATLAS check cases in target simulator
  - LDRA static/dynamic analysis
- Design with a Large-Scale Mode
  - Configuration Management
  - Time and memory required to simulate and code
What’s Next

• R14
  – Model Reference is important new technology
    • Incremental code generation
  – EML could be very useful for utility development
  – Improvements in code generation
    • Better MISRA compliance
    • More efficient code
  – Improved code customization capabilities

• R15
  – More improvement needed in code efficiency
  – Mapping of function interfaces from model to code
  – Improvements to reusable function code
    • Work toward the goal of producing a single function
Flight Test Video

- X-35A Highlights
- X-35B Highlights
- X-35C Highlights