Joint Strike Fighter

LOCKHEED MARTIN

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

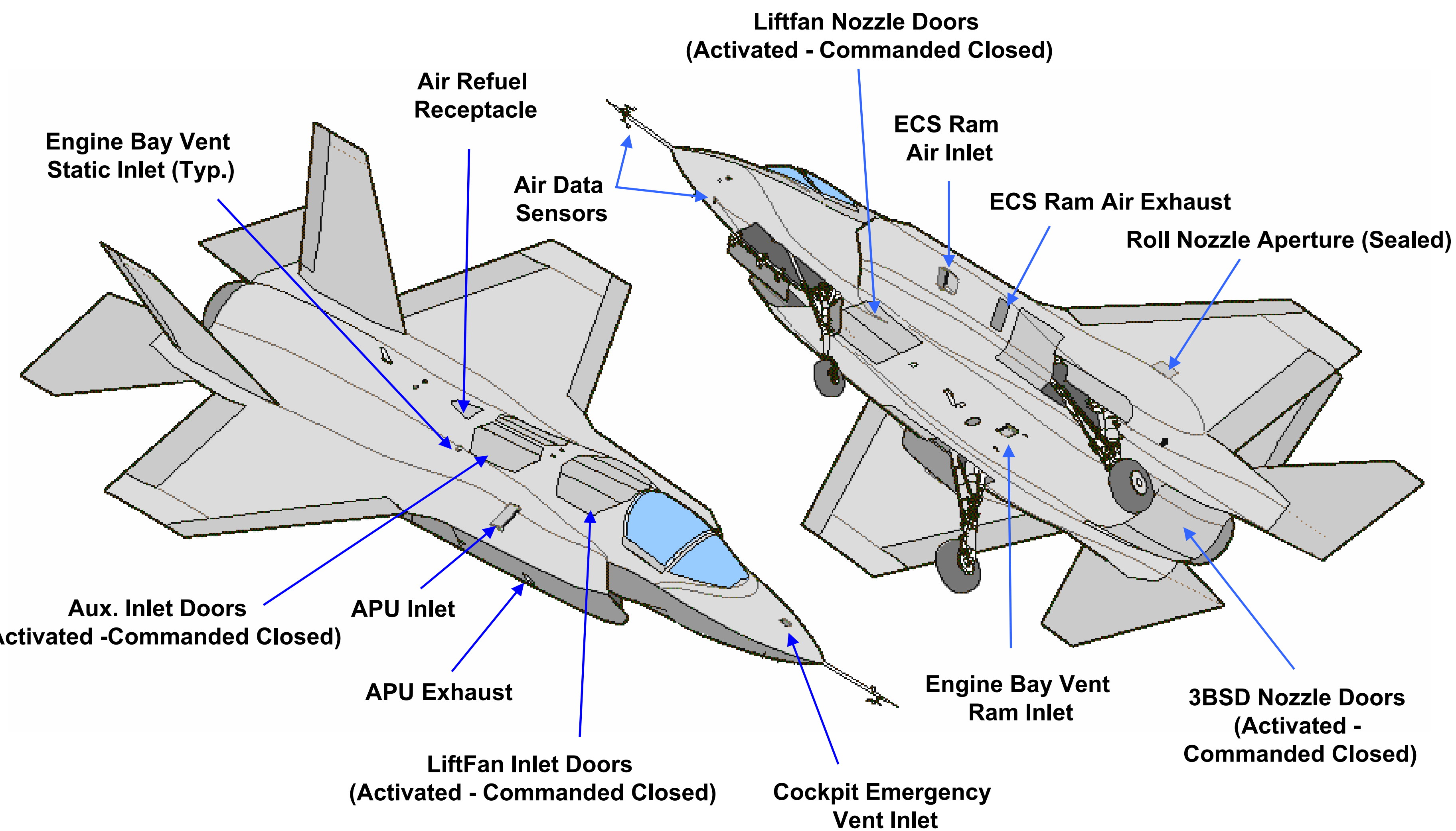
Flying Qualities, Engine-Inlet Compatibility, and Flight Performance at Representative Mission Points

CV

Carrier Suitable Flying and Handling Qualities and Flight Performance at Representative Mission Points

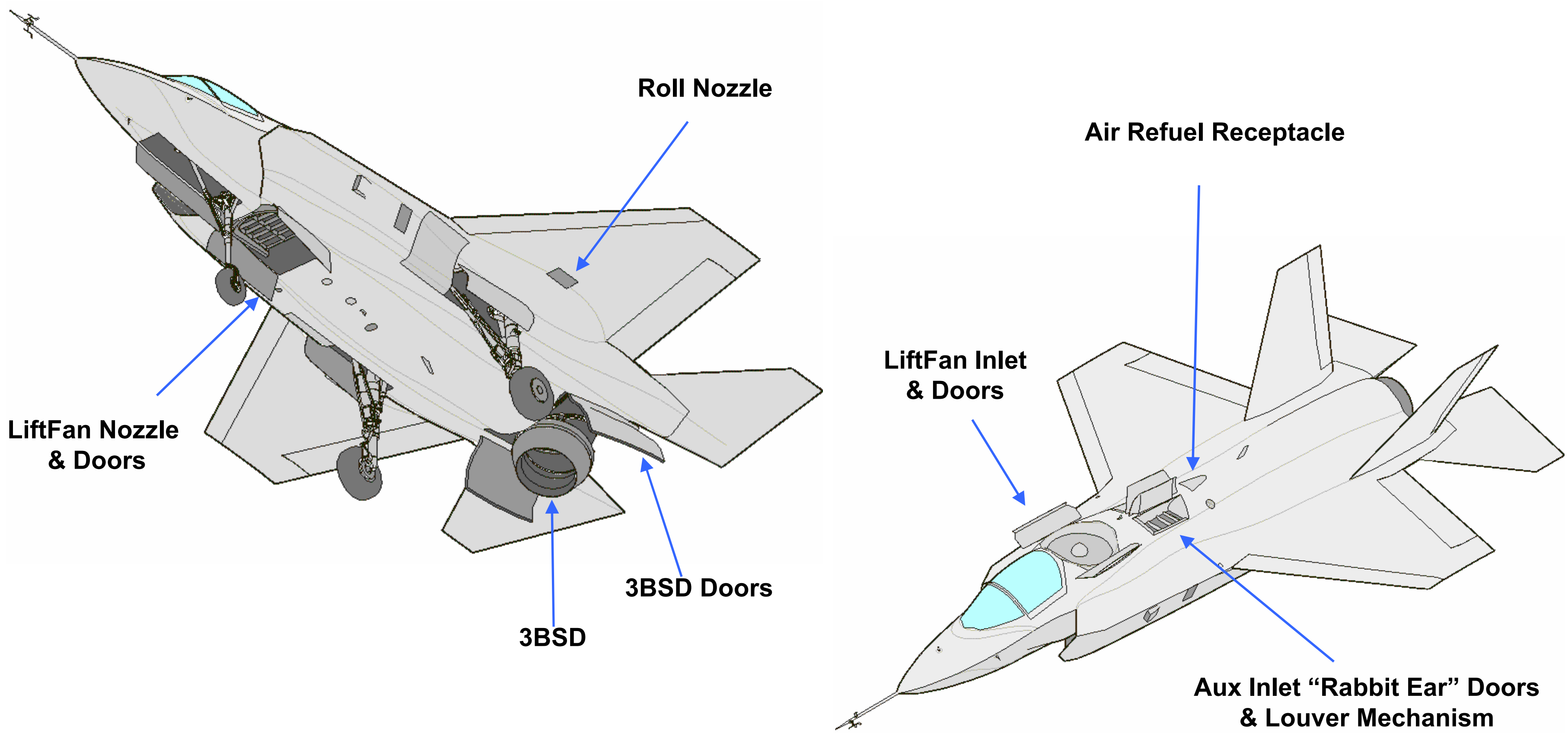
X-35A/B Features

Conventional Configuration



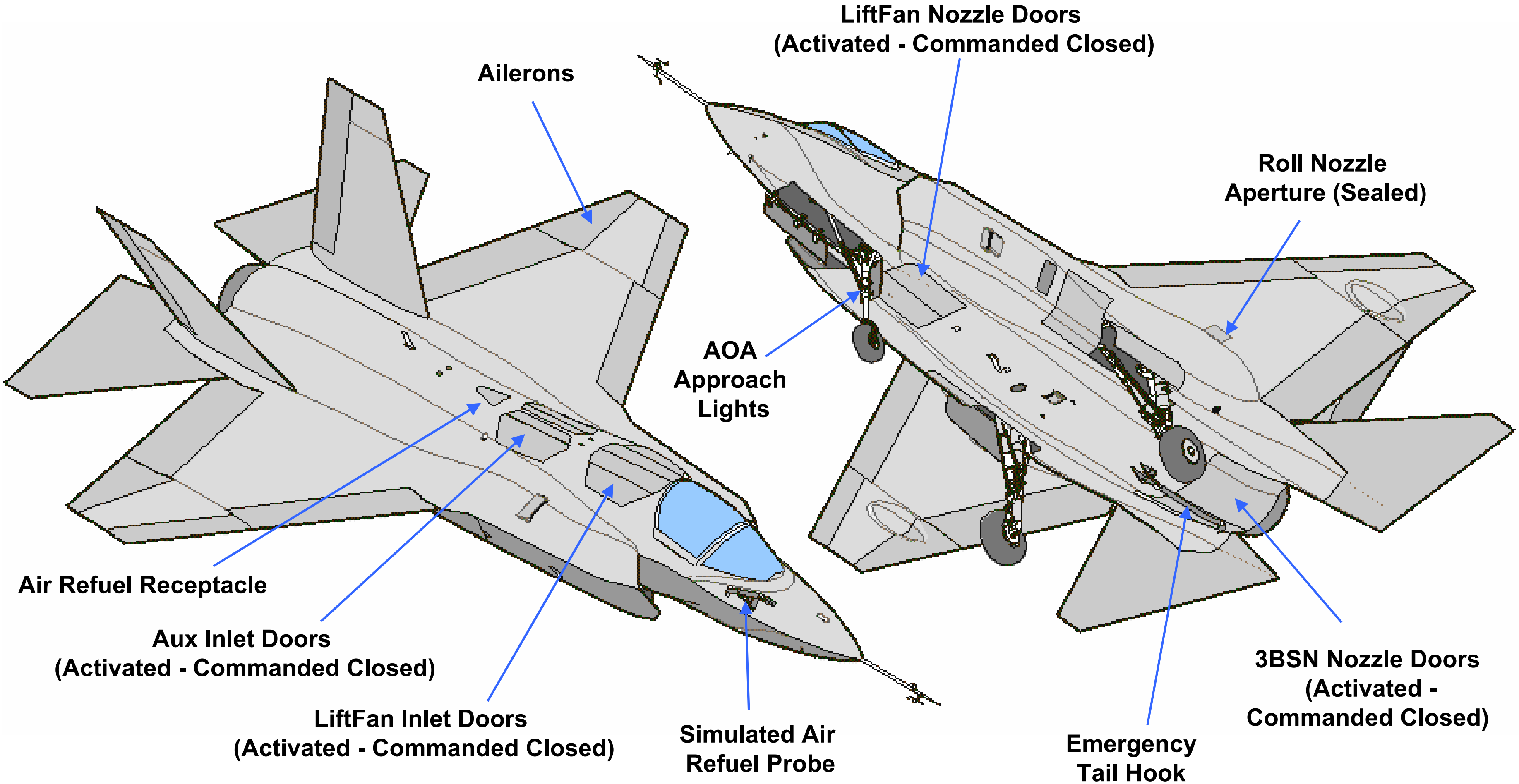
X-35A/B Features

STOVL Configuration



X-35C Features

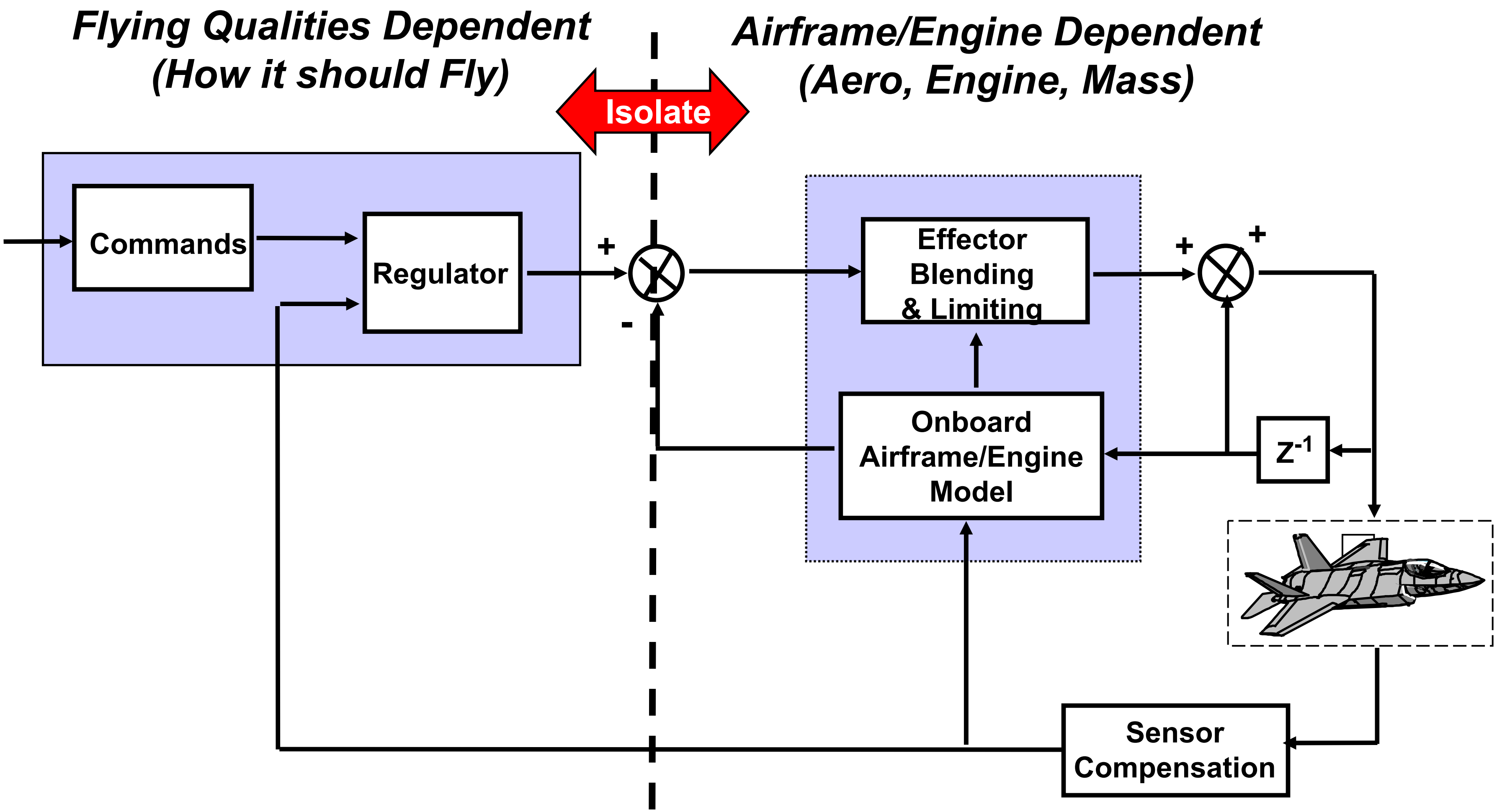
CV Configuration



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



Common Control Law Structure for All Aircraft Variants

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*

<u>Linear Aircraft Equations of Motion</u>

$$\dot{x} = Ax + Bu$$
$$cv = Cx$$

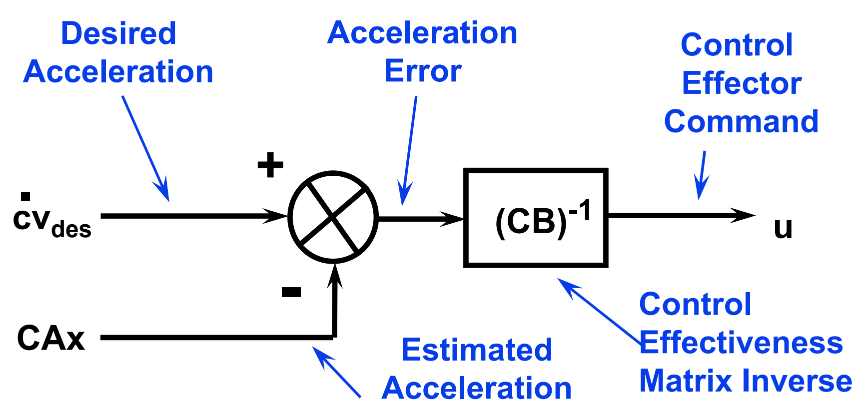
x - states
u - effectors
cv - control variable

A - Aircraft Dynamics Matrix
B - Control Effectiveness Matrix
C - Control Variable Matrix

<u>Dynamic Inversion Formulation</u>

$$\dot{cv}_{des} = C\dot{x} = CAx + CBu$$

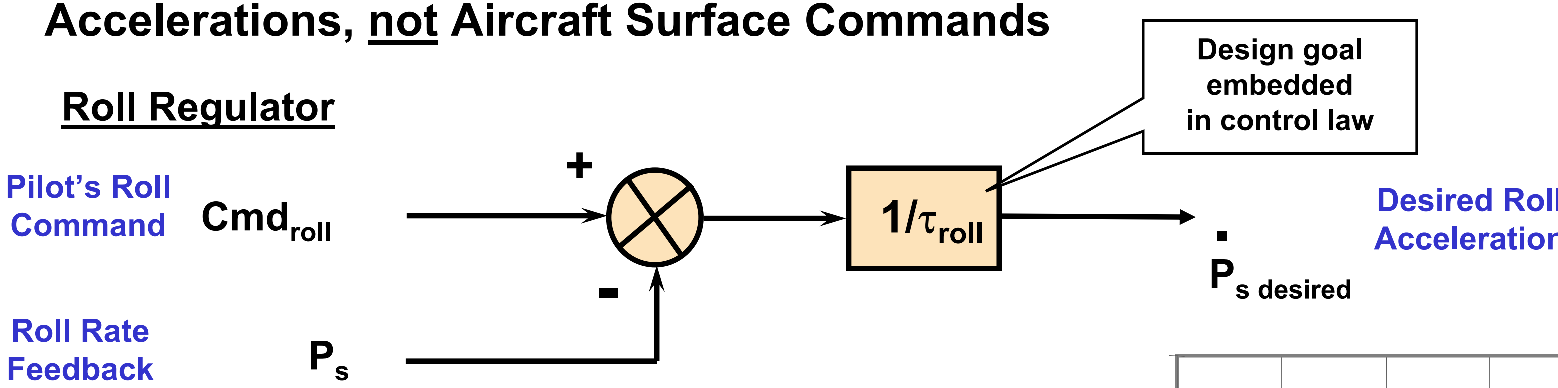
$$u = (CB)^{-1}(\dot{cv}_{des} - CAx)$$



Roll Regulator Example

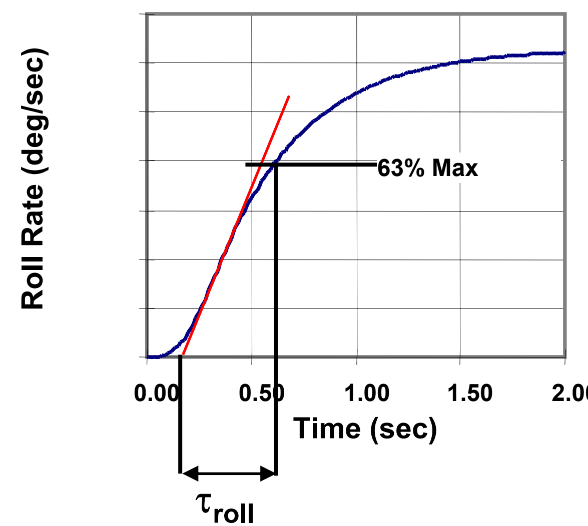
- Map the Pilot Commands and Feedbacks into the Desired Aircraft Accelerations, <u>not</u> Aircraft Surface Commands

<u>Roll Regulator</u>



$$\dot{P}_{s\ des} = 1/\tau_{roll} * (Cmd_{roll} - P_s)$$

$$\frac{P_s}{Cmd_{roll}} = \frac{(1/\tau_{roll})}{(s + 1/\tau_{roll})}$$



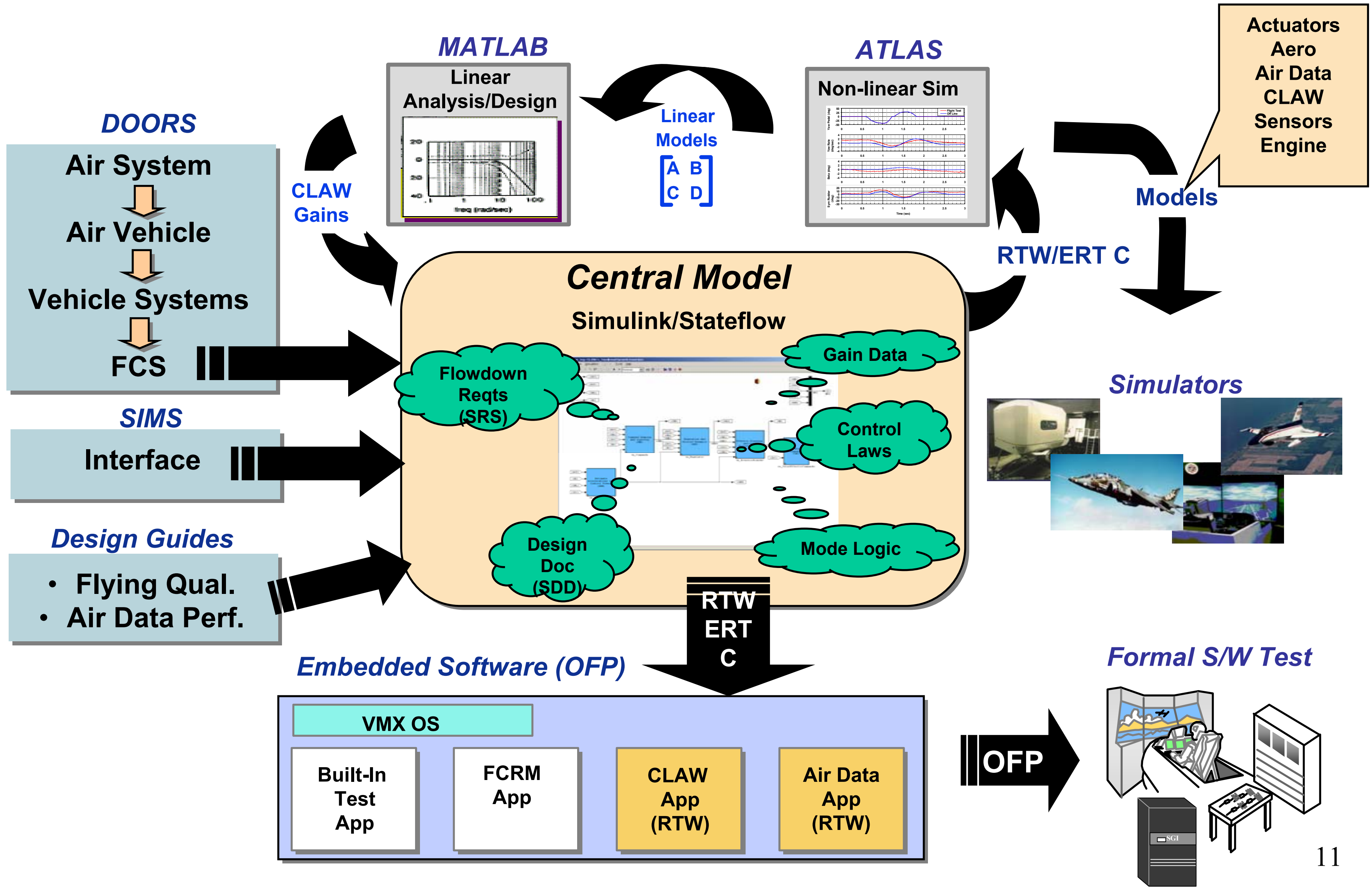
Simple Dynamic Inversion Roll Control Law Provides a Classical First Order Roll Response

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*

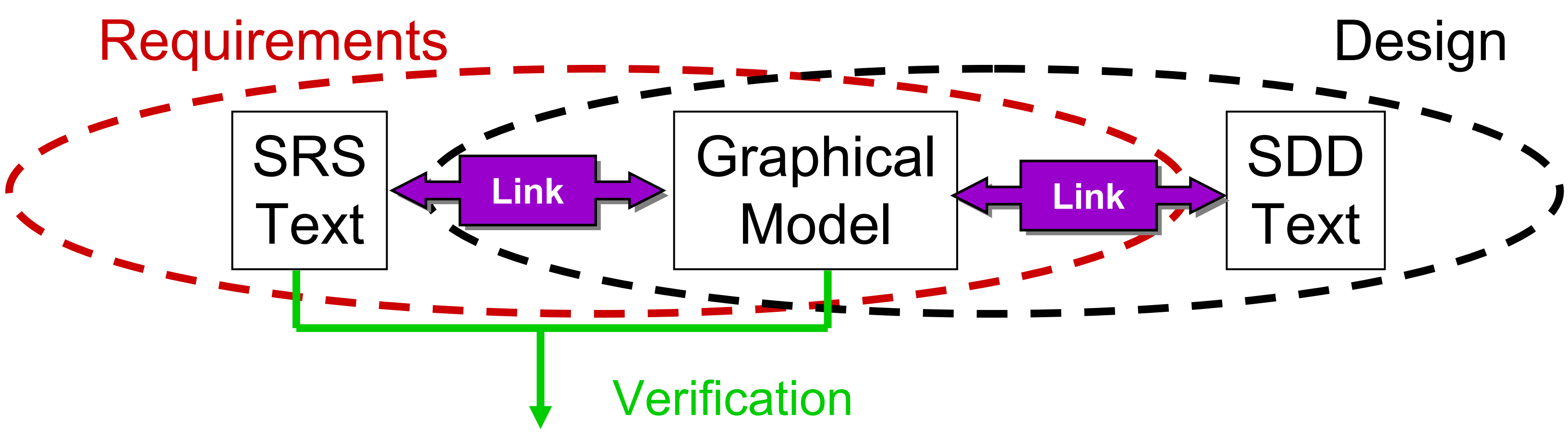
Not Just A Higher Level Language for Programming – A Different Software Development Paradigm

Model-Based Development Process



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