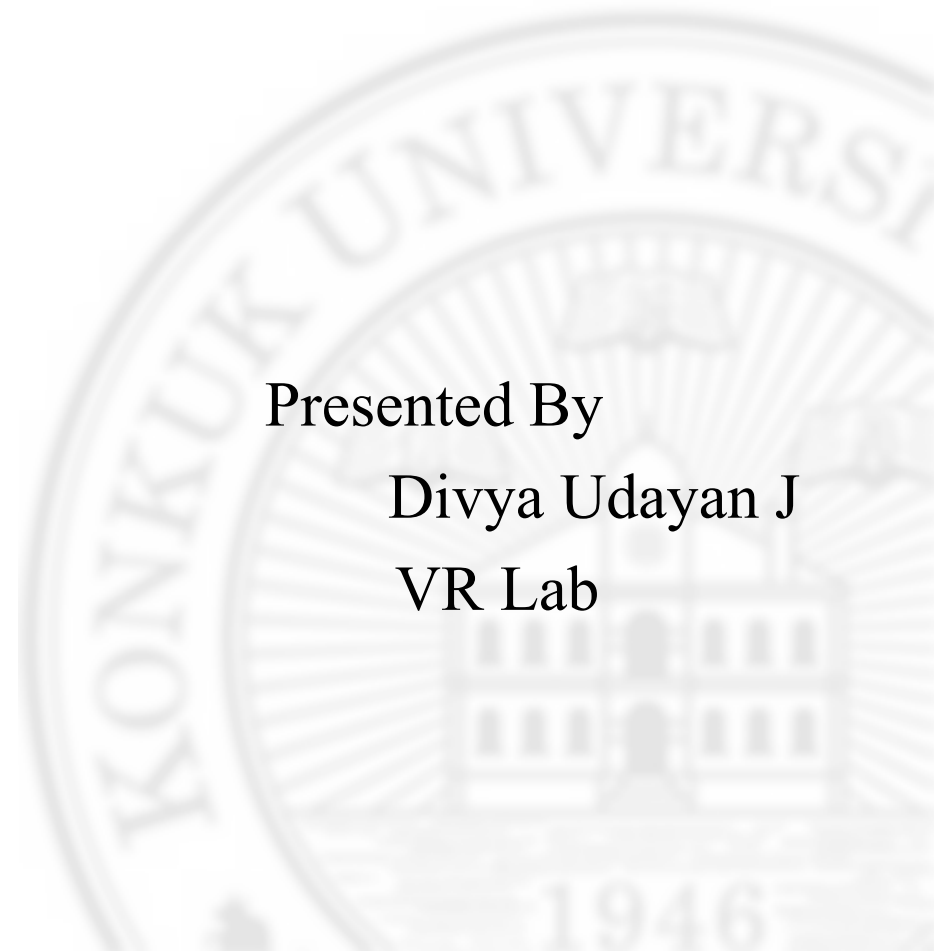


DESIGN VERIFICATION OF STABILITY CONTROLLER MODEL OF AIRPLANE USING SCADE

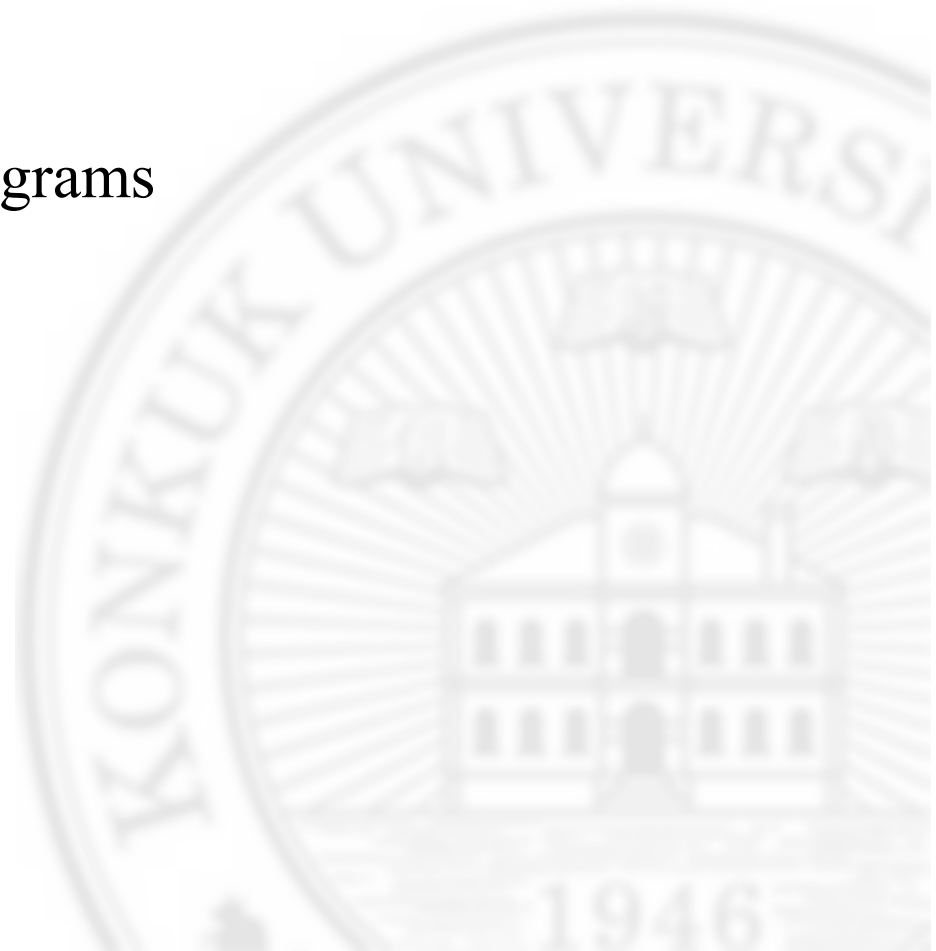


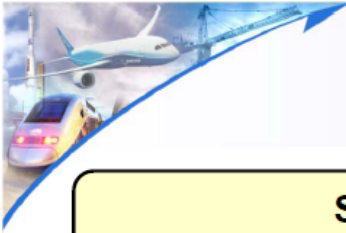
Presented By
Divya Udayan J
VR Lab



Overview

- Introduction to SCADE SUIT
- Objective of term project
- Requirement Specification
- Data flow and Control flow diagrams
- Code generation
- Verifying Design Correctness
- System Testing
- Simulation and demo
- Conclusion

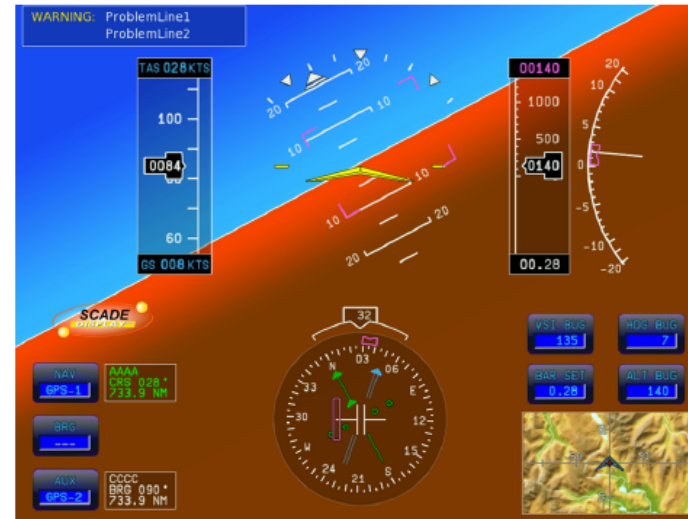
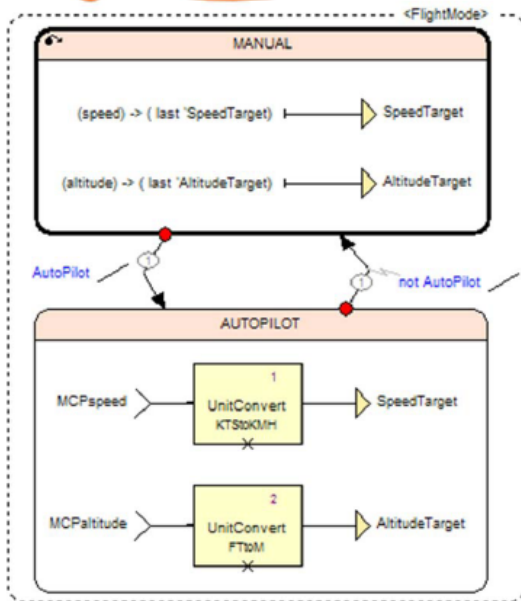




Mission and Safety-Critical Design with Embedded Graphics

SCADE Suite
Integrated Data Flow & State Machines

SCADE Display
Embedded Graphics



Fully Integrated Design Suite



The SCADE Certified Software Factory

DESIGN

VERIFY

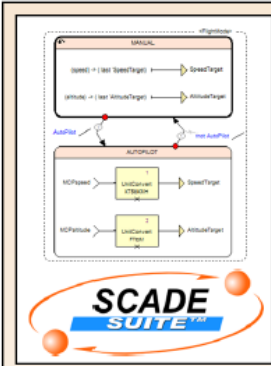
GENERATE

SYSTEM TEST

3 Requirements
 3.1 Create State Management
 3.1.1 State description
 3.1.2 Inputs
 3.1.3 Outputs
 3.1.4 Detailed specification

Algorithm Design Capture

Architecture Design Capture



Debugging & Simulation

Formal Verification

Time & Stack Analysis

Object Code Verification

Model Coverage Analysis

SCADE Suite/SCADE Display Integration

Rapid Simulation

Design Checking

SCADE Suite KCG

RTOS Adaptors

SCADE Display KCG

OpenGL/SC Compliant



Requirements Management Gateway

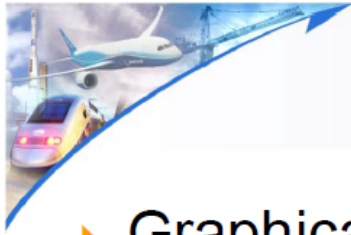
Integrated Configuration Management

Automatic Design Documentation

DO-178B
IEC 61508
EN 50128

Certification Kits, Certificates & Handbooks

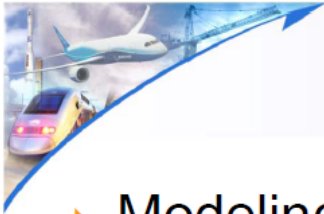
MANAGE & TRACE



Unified Modeling Style

Modeling Capabilities

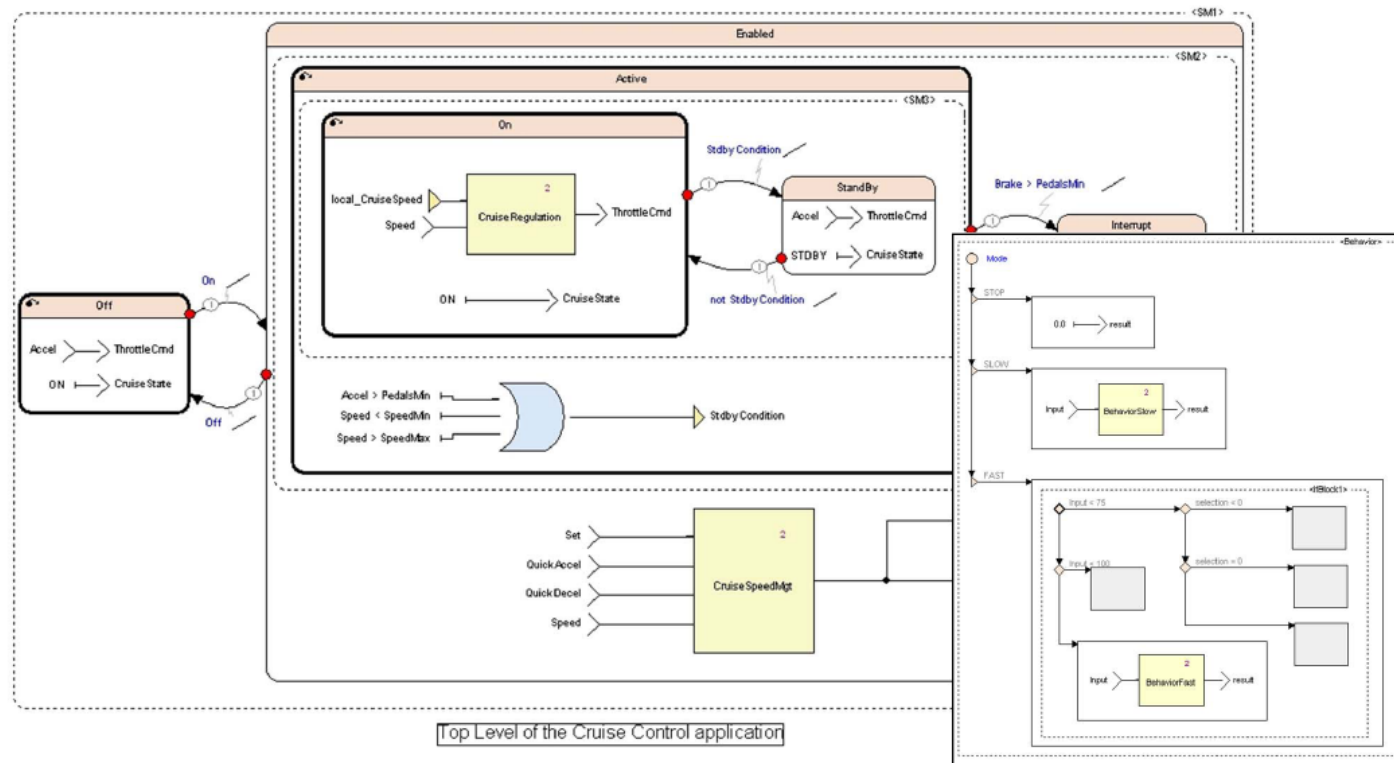
- ▶ Graphical formalism
 - ▶ Block diagrams, to specify the algorithmic part of applications, such as control laws and filters
 - ▶ Hierarchical state machines, to model the control part of applications
 - ▶ Decision diagrams
 - ▶ Packages, data types, constants
 - ▶ Arrays & iterators
 - ▶ Libraries
- ▶ The unique **integration of data flow and safe state machines** allows you to model the whole application with the same formalism



Unified Modeling Style

Integrated Data Flow & State Machines

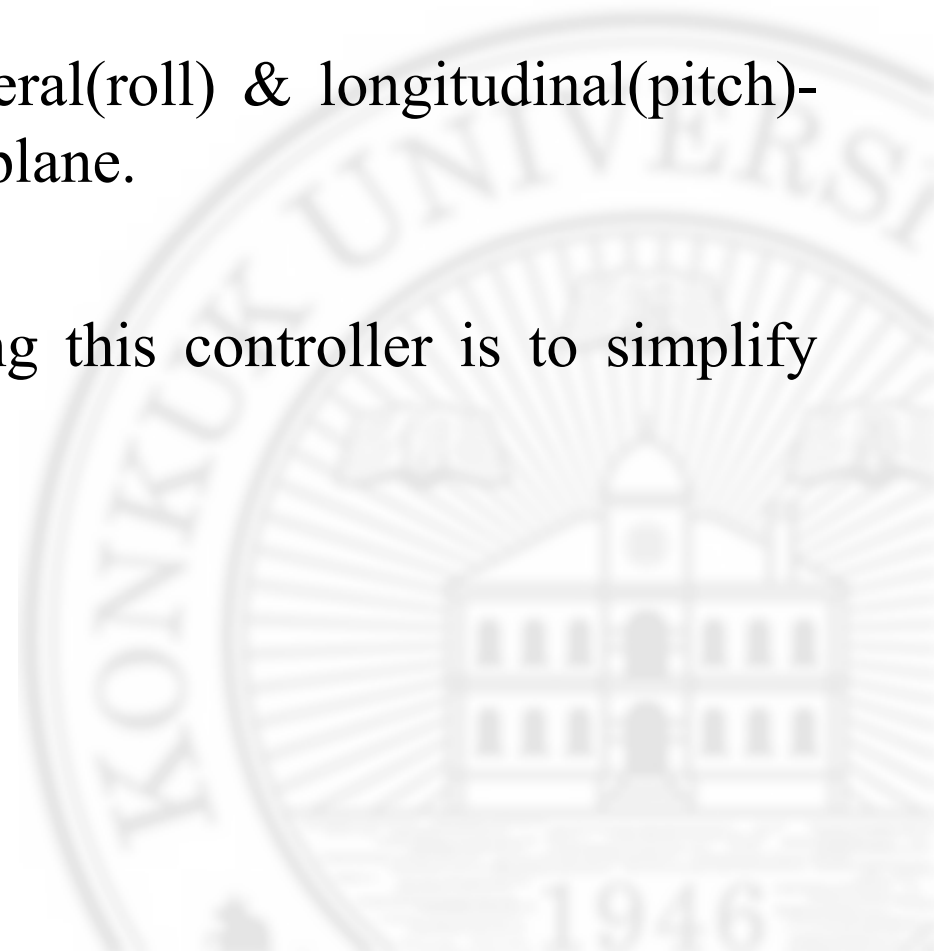
- ▶ Modeling flexibility:
Power of nested data flow & control flow



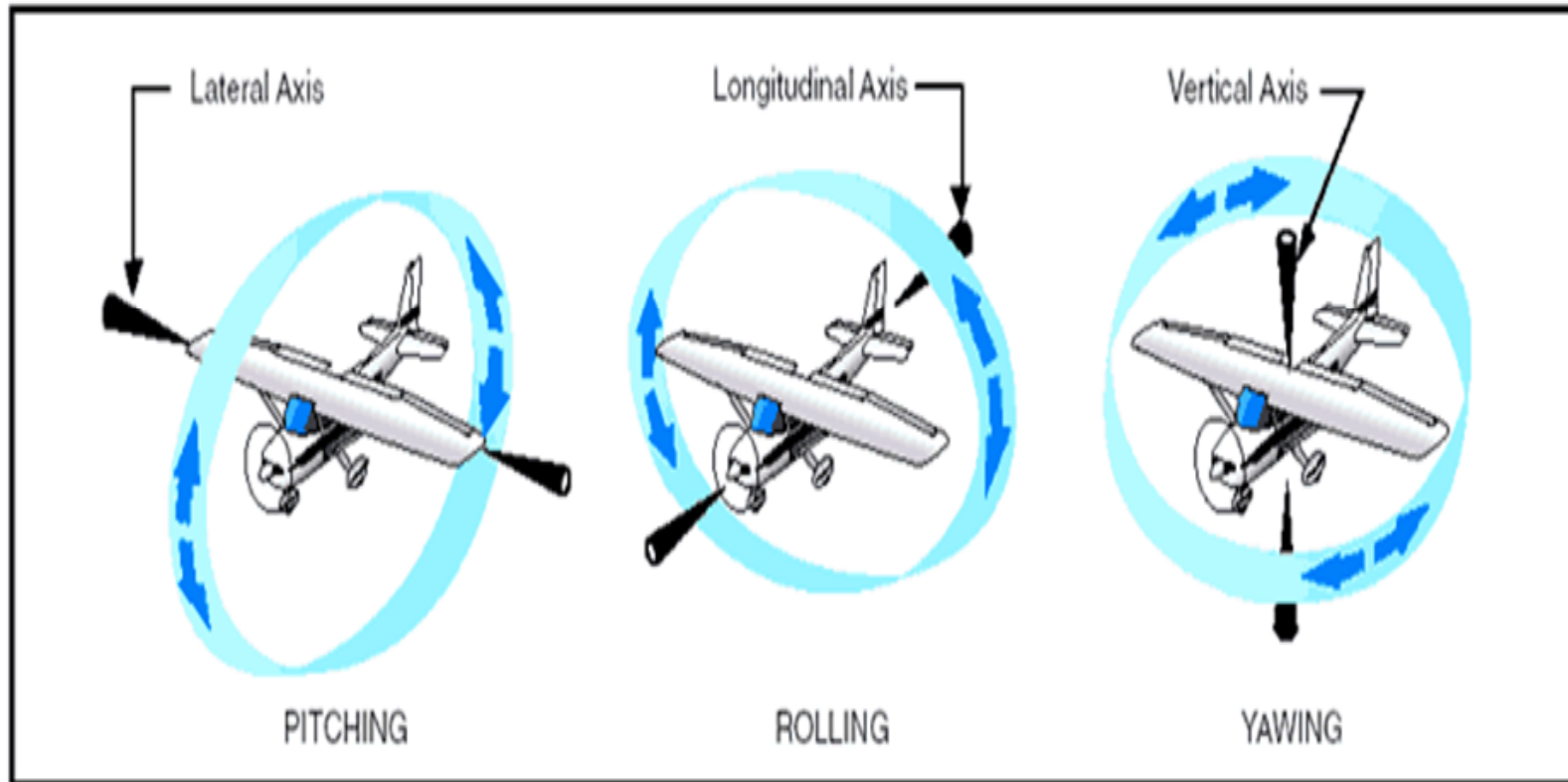
Objective



- Design verification of the stability controller for an airplane
- Checking its effect on the lateral(roll) & longitudinal(pitch)-directional dynamics of the airplane.
- Motivating factor for designing this controller is to simplify the piloting of the airplane.

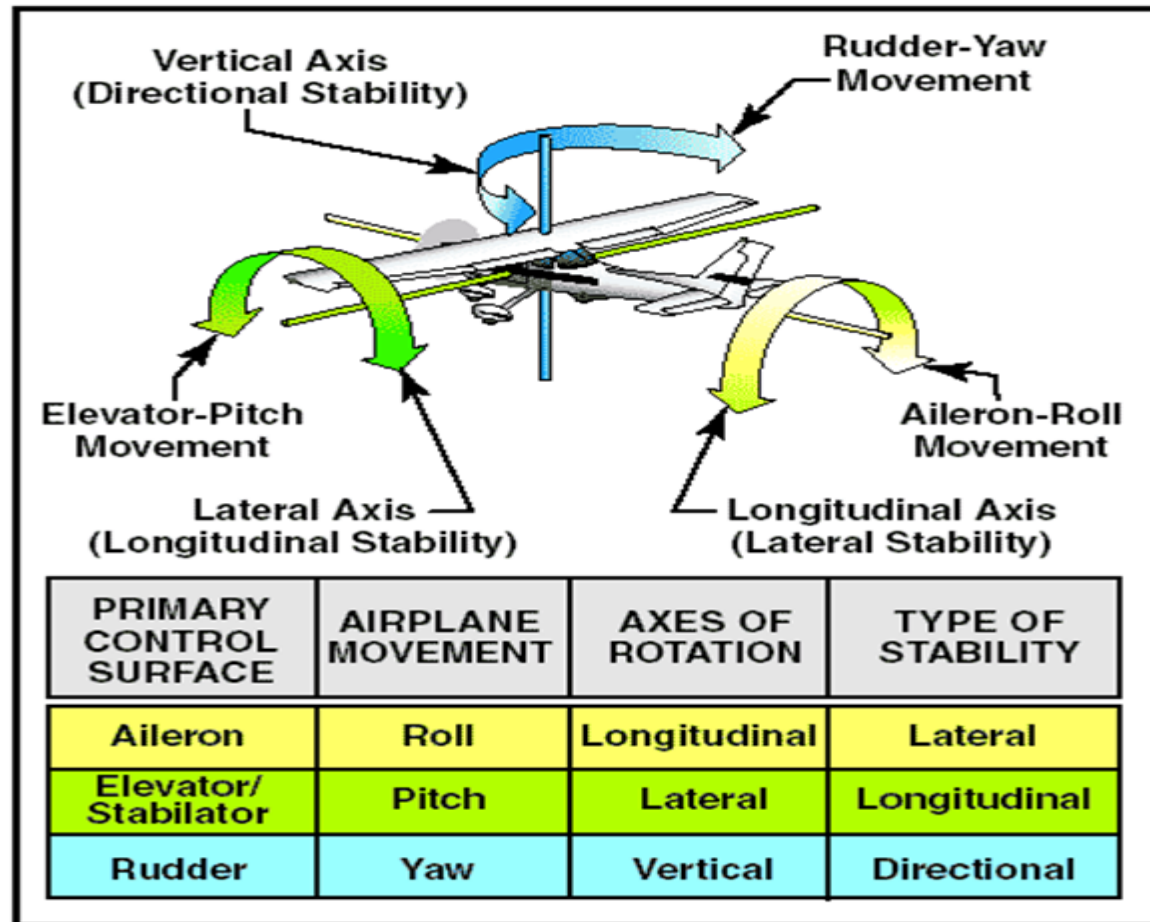


Flight controls



Axes of an airplane

Airplane stability



Airplane controls, movement, axes of rotation, and type of stability

Requirement Specification(1/4)

1) Roll rate calculation requirement :

The roll rate calculation subsystem calculates the plane roll rate, according to a joystick command and the adverse yaw coupling effects.

Inputs : Joystick command, Adverse yaw induced by left wing, Adverse yaw induced by right wing.

Outputs : Plane roll rate

simplified coupling effect is calculated as follows:

$$\mathit{rollCoupling} = (\mathit{leftAdverseYaw} - \mathit{rightAdverseYaw}) \times 0.1$$

plane roll rate is calculated as follows:

$$\mathit{rollRate} = (\mathit{joystickCmd} - \mathit{rollCoupling}) \times 0.25$$

The absolute value of the plane roll rate has to be saturated to 25.0.

Initialization : At system initialization, the plane roll rate is 0.0.

Requirement Specification(2/4)

2) Roll rate warning alarm requirement :

The roll rate warning alarms subsystem computes left and right warning alarms, which sound, respectively if the plane roll rate is strictly less than -15.0° per second or strictly greater than 15.0° per second.

Inputs : Plane roll rate

Outputs: Left warning alarm, Right warning alarm

Initialization : At system initialization, the left and right warning alarms do not sound.

3) Roll Mode Management

The roll mode management subsystem computes the plane “roll mode” (either Off, Nominal, or Failsoft) according to the ON/OFF button being pressed and the plane roll rate value.

Inputs : - Plane roll rate (absolute value), ON/OFF button

Output : Roll Mode

Requirement Specification(3/4)

The roll mode has three possible states:

OFF:

Active when:

- At initial state;
- Previous state was NOMINAL or FAILSOFT and ON/OFF button is pressed.

The roll mode value is then Off.

NOMINAL:

Active when:

- The previous state was OFF, the ON/OFF button is pressed, and the absolute value of the plane roll rate is less than FailSoftRoll.
- The previous state was FAILSOFT and the absolute value of the plane roll rate is less than FailSoftRoll

The roll mode value is then NOMINAL.

FAILSOFT:

Active when:

- The previous state was OFF, the ON/OFF button is pressed, and the absolute value of the plane roll rate is strictly greater than FailSoftRoll.
- The previous state was NOMINAL and the absolute value of the plane roll rate is strictly greater than FailSoftRoll

The roll mode value is then FAILSOFT.

Initialization : At system initialization, the roll mode is Off.

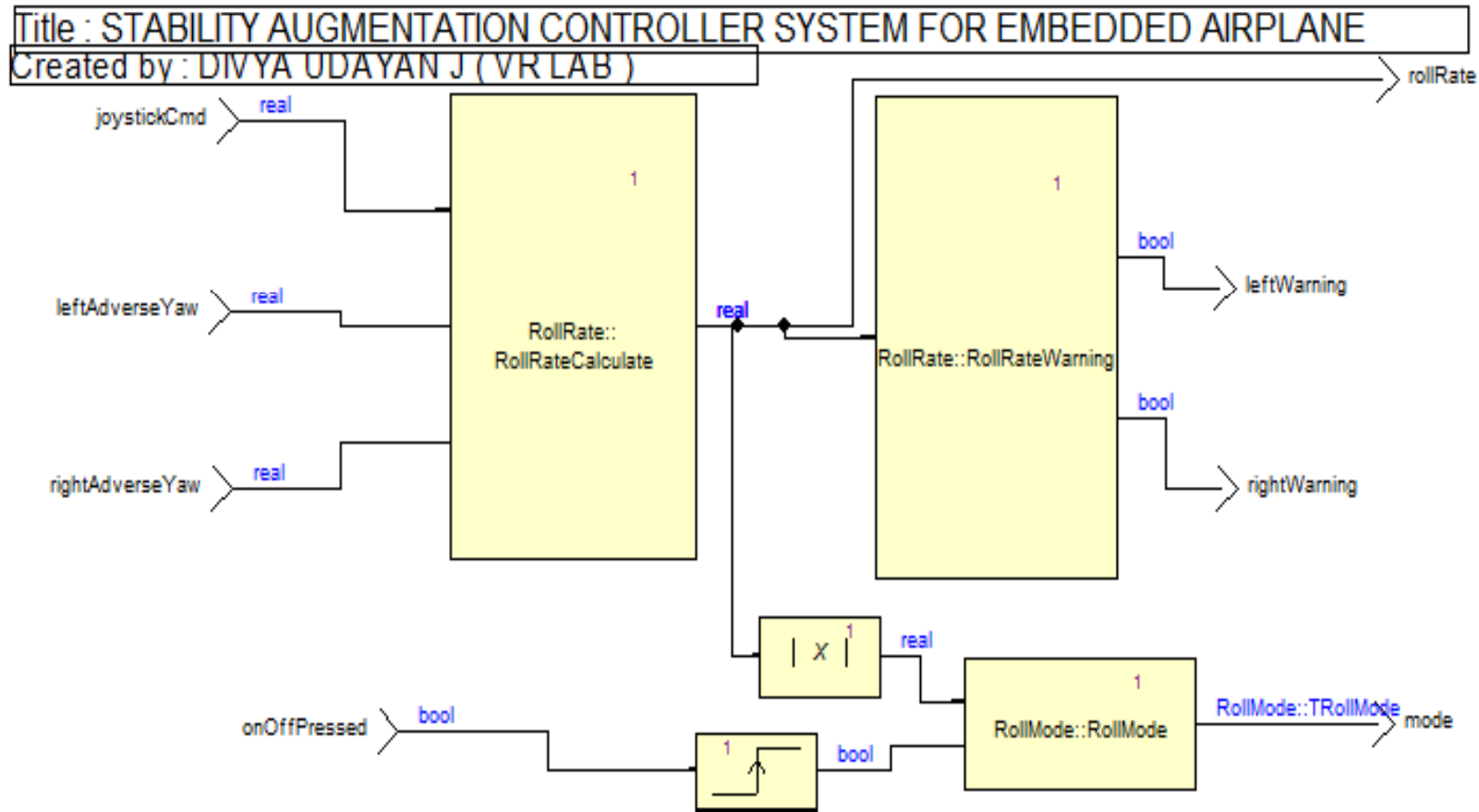
Requirement Specification(4/4)

4) Constant definitions

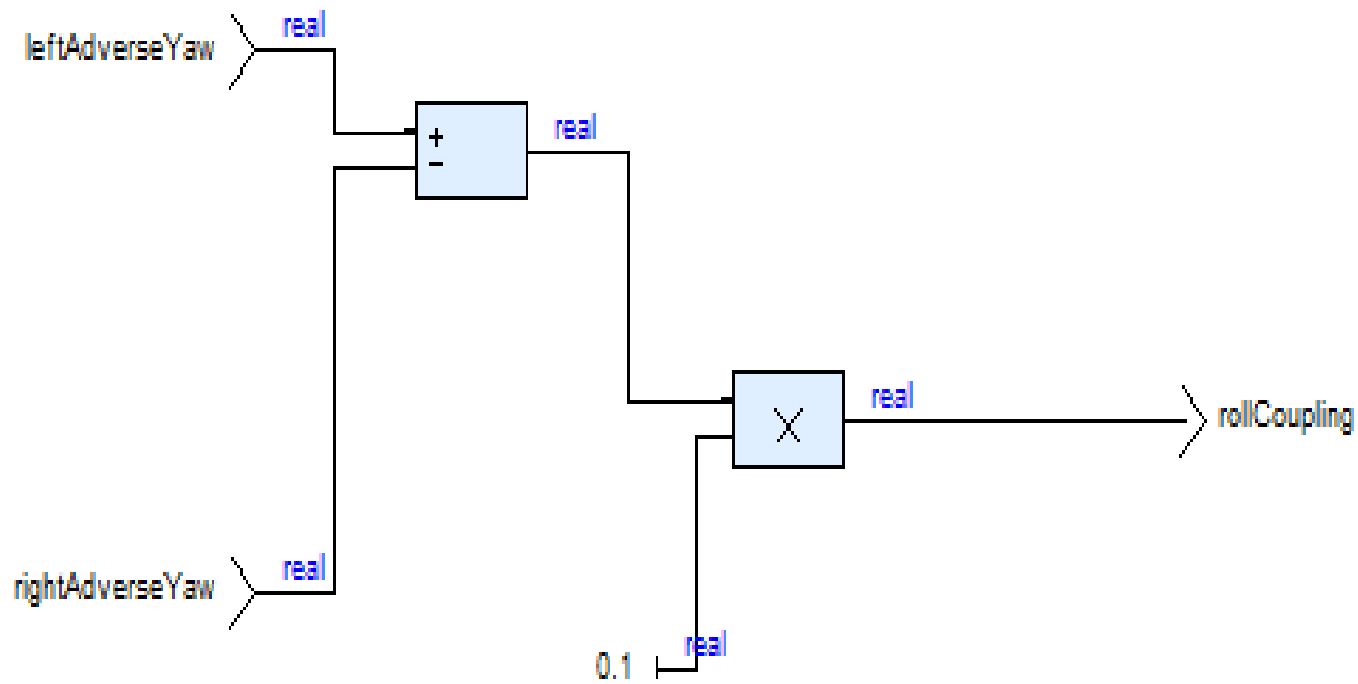
Name	Field	Value	Unit
RollRateWarning	Left	-15.0	Degrees/sec
	Right	15.0	Degrees/sec
FailSoftRoll		20.0	Degrees/sec



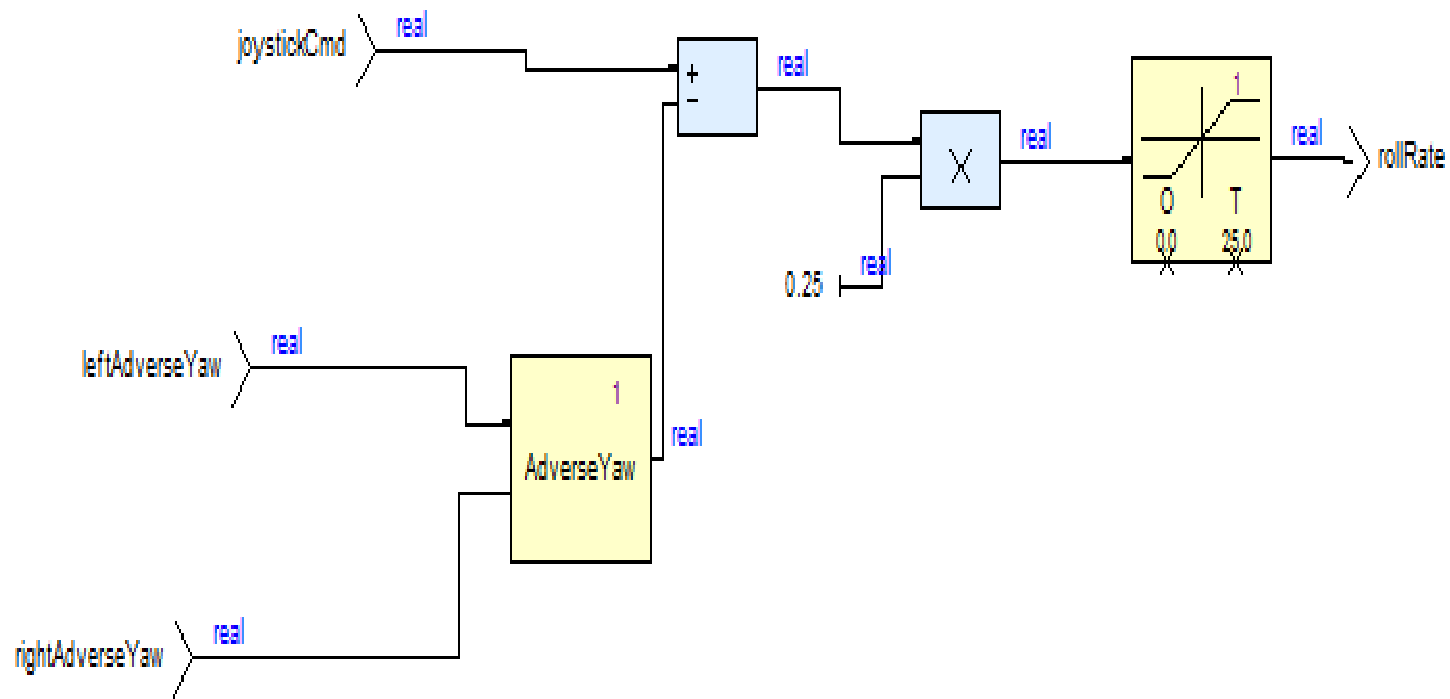
Data Flow of the Roll Controller



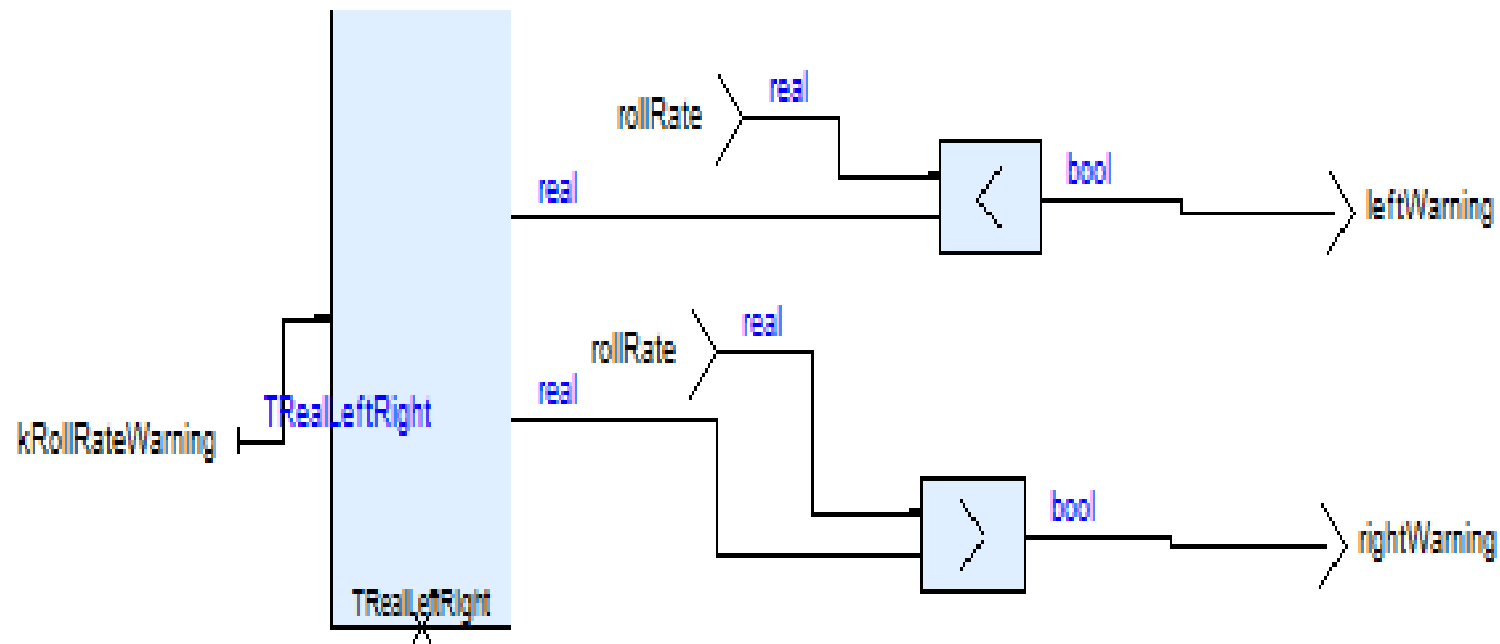
Data flow of roll coupling



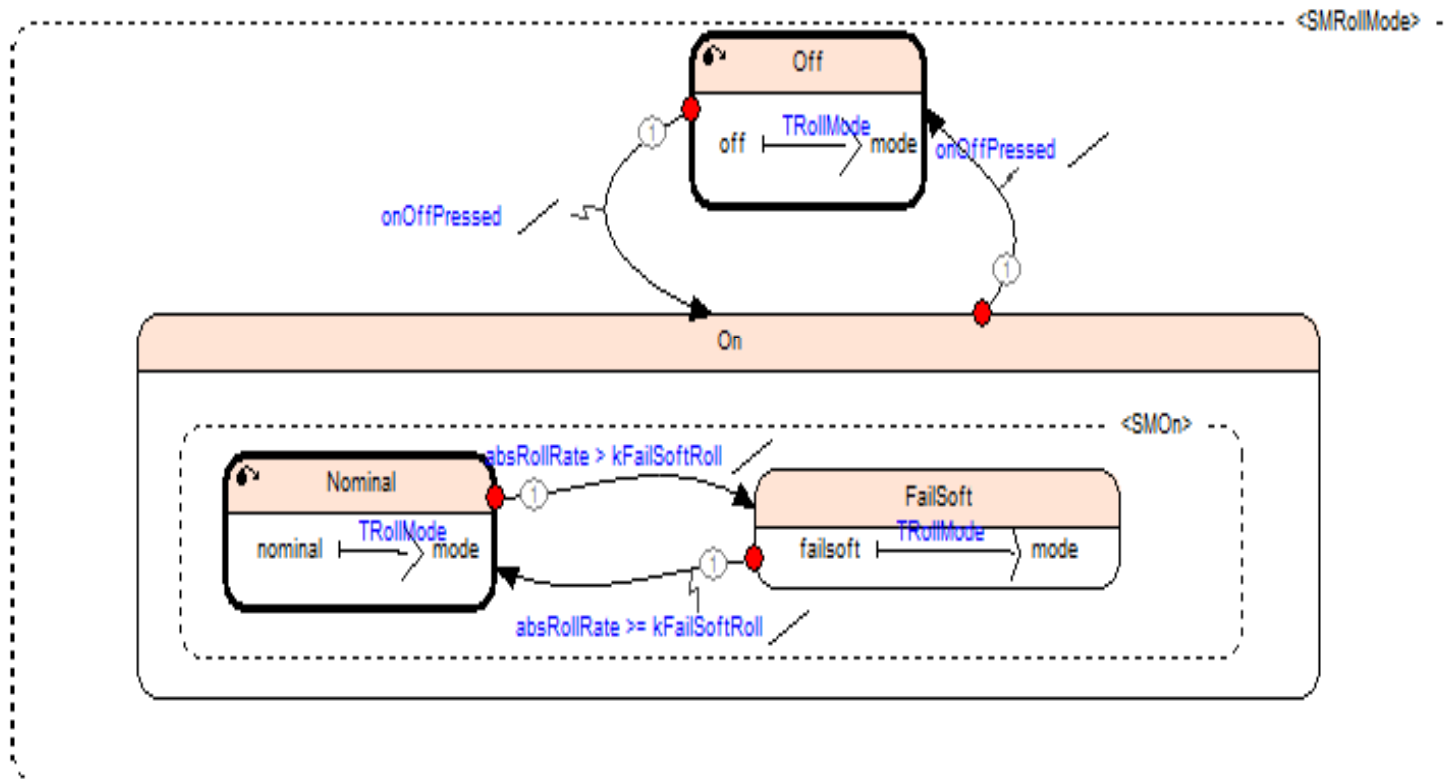
Data flow of RollRate Calculation



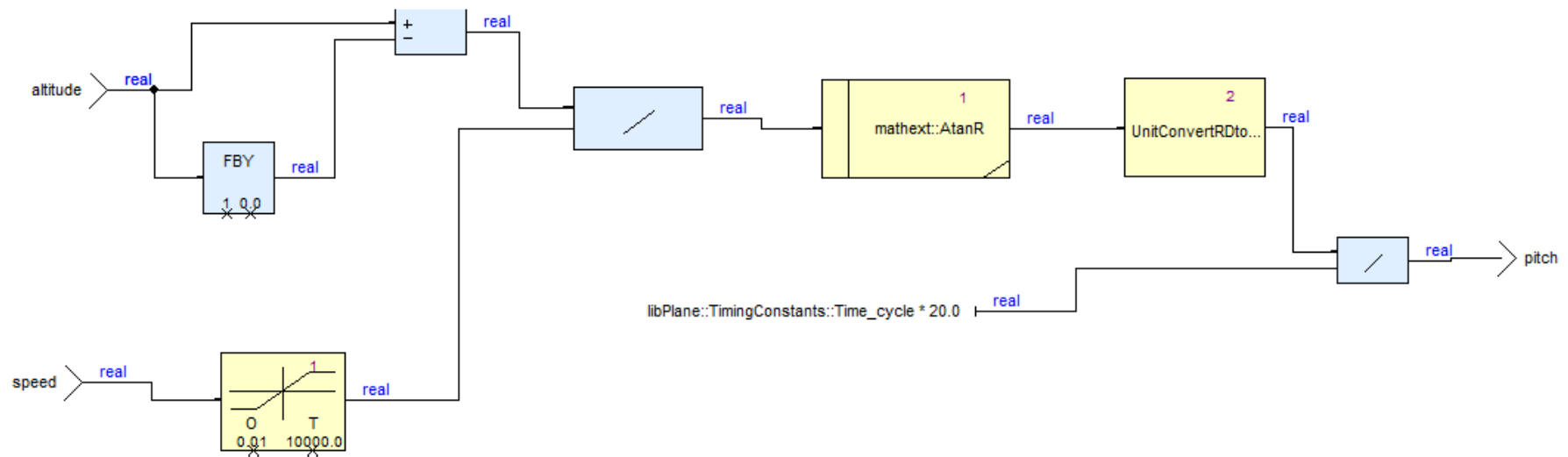
Data Flow of Rollrate warning



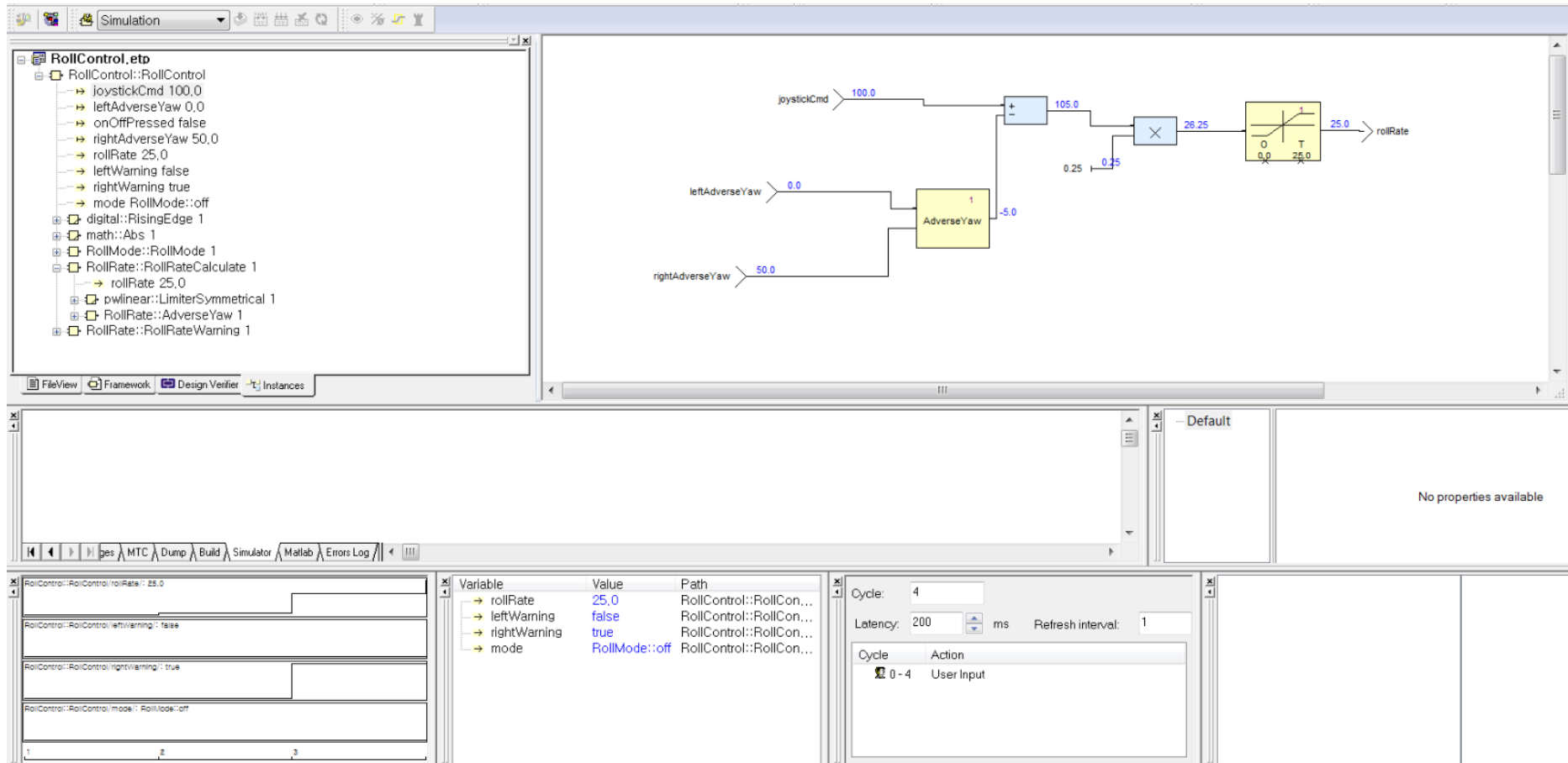
Control Flow Diagrams



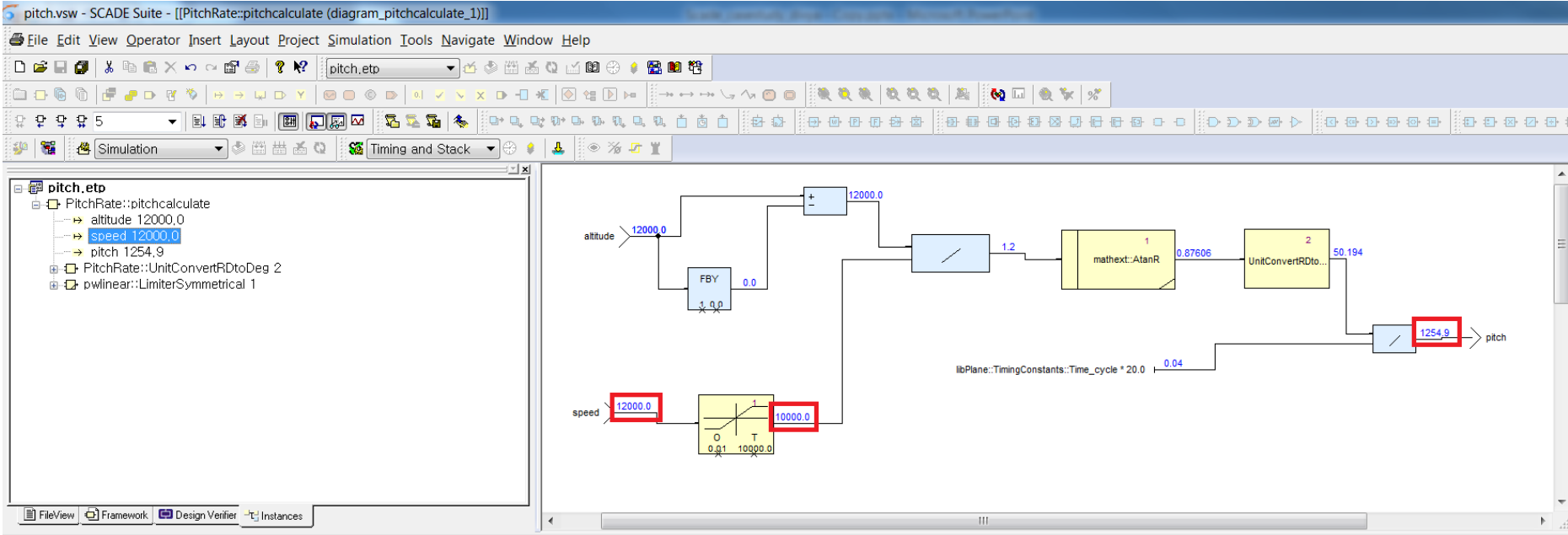
Data Flow of Pitch Control



Simulation of Roll Control



Simulation of Pitch Control



Code Generation

The screenshot displays a software development environment with the following components:

- File Explorer:** Shows a project named 'RollControl.etp' with a 'Generated Code' folder containing various files like 'AdverseYaw_RollRate.c', 'kcg_consts.c', 'kcg_sensors.h', and 'RollControl_RollControl.c'.
- Code Editor:** Contains C code for a function 'RollControl_reset_RollControl'. The code includes headers for 'kcg_consts.h', 'kcg_sensors.h', and 'RollControl_RollControl.h'. It defines a function that resets roll control parameters and calculates the roll rate.
- Message Window:** Displays a table of messages from the code generator.

Category	Code	Message
Code Generator		
Information	Log Files	LOGFIL Log Files
Information	Generated Files	GENFIL KCG- Generated files
Information	Generated Files	GENFIL Code Generator Generated files

Semantic Verification

The screenshot displays a software interface for semantic verification. The main window shows the result of a check for the operator `RollMode::RollMode/` in the model `RollControl`. The result is "No error detected." and "End of document." The interface includes a file tree on the left, a command palette, and a properties panel at the bottom right. The status bar at the bottom shows the current project and simulation status.

Result of check for operator `RollMode::RollMode/` in model `RollControl`

No error detected.

End of document.

Properties Panel:

- Name: RollControl
- Path: RollControl:RollControl/
- Filename: RollControl1.xscade
- Separate File Name
- Visibility: Public Private

Status Bar: Messages | MTC | Dump | Build | Simulator | Matlab | Err

System Analysis



Verifying System Correctness(1/2)

- Using Observer Property, Divide by zero, Overflow stack

The screenshot displays a software verification tool interface. On the left, a tree view shows a project named 'RollControl2_etc' with sub-items like 'Proof Objectives', 'RollControl.leftWarning', and 'RollControl.rightWarning'. A red box highlights 'Proof Objectives'. In the center, an 'Analysis' dialog box is open, showing a log of analysis steps such as 'depth: 0', 'Strategy: strategy_type_generic.Sub-strategy: strategy_type_induction', and 'po 0 leftWarning: po_falsifiable'. A green progress bar is visible at the bottom of the dialog. On the right, a 'General Info' panel shows analysis details: 'time of analysis' (03 12 2011 15:37), 'model' (RollControl2), and 'user' (VRLab). Below this, a 'Sum Up' section shows 'RollControl.leftWarning' as 'Falsifiable'. At the bottom, a table lists tasks and their results:

Task	Result
RollControl.leftWarning	Falsifiable

A red box highlights the 'Falsifiable' result in the table.

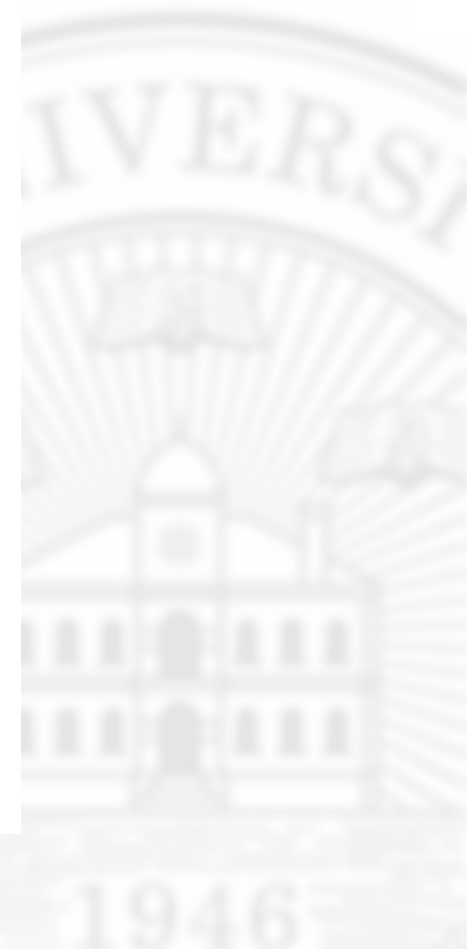
Verifying System Correctness(2/2)

The screenshot displays a software verification tool interface. On the left, a tree view shows a project named 'RollControl2.eto' with several check categories: 'Proof Objectives', 'Division-by-Zero Checks', 'Overflow Checks', and 'Strategies'. The 'Division-by-Zero Checks' category is highlighted with a red box. In the center, an 'Analysis' dialog box is open, showing a log of the verification process. The log includes: 'Analysis Initialization.', 'DV Version 6.2 (build 13)', 'ProverSL Data Edition v4.1.15', 'Starting proof with 0 po', and a list of strategy types: 'strategy_type_generic.Sub-strategy', 'strategy_type_pre_processing', 'strategy_type_generic.Sub-strategy.strategy_type_expansion', 'strategy_type_generic.Sub-strategy.strategy_type_saturation', and 'strategy_ok'. The dialog box has a green progress bar and buttons for 'Skip', 'Abort', and 'Close'. On the right, a 'Tasks' panel shows the task 'RollRateWarning.check_division_by_zero' with a 'Valid' result. Below this, a 'General Info' section provides details: 'time of analysis' (2011 15:39), 'model' (RollControl2), and 'user' (VRLab). A 'Sum Up' section shows 'RollRateWarning.check_division_by_zero Valid'. A 'Tasks' section below that shows 'RollRateWarning.check_division_by_zero' with details: 'Node: RollRate::RollRateWarning', 'Strategy: Default Division by zero - Prove', 'Result: Valid', 'Translation time: 0 s', and 'Analysis time: 0 s'. At the bottom, a table lists the task and its result:

Task	Result
RollRateWarning.check_division_by_zero	Valid

Proof Meaning

Proof result	Meaning
Valid	The verified property is always true mathematically.
Falsifiable	Property is false because Design Verifier detects a valuation of your system inputs such that the output of the observer operator is not equal to the value specified in the proof objective. Such input valuation is called a counter-example. To access the counter-example of a falsifiable property, see "Displaying Counter-Examples" on page 681.
Indeterminate	The proof reaches no significant conclusion.
Interrupted	Either you manually aborted the analysis in the status window, or Design Verifier finds no counter-example before strategy time-out (see "Setting Standard Strategy Options" on page 661).
Stop Depth Reached	The analysis reaches its execution cycle depth set in the debug strategy and Design Verifier cannot report any significant result (see "Setting Debug Strategy Options for Proofs" on page 662).
Raised an Error	The cause of error displays in the message field of the report. Possible errors are: use of unsupported Scade language features, bad syntax or semantics in your design, or an internal error of the proof engine (see "Understanding Design Verifier Error Messages" on page 677).
Error: Non Linear Property	Verification is impossible because the property is expressed with non-linear expressions or functions. Check your design and simplify the property operator or its context in the observer operator.
Contradictory	You expressed contradictory assertions in the analyzed design. Design Verifier is unable to resolve the analysis and stops the process. Revise your design to solve this error.



System Testing

- Graphical panel display

The screenshot displays a MATLAB/Simulink environment. On the left, the 'RollControl.etb' model is shown with the following parameters:

- rollRate: -18.189
- leftWarning: true
- rightWarning: false

The central part of the image shows a Simulink block diagram with a 'TRollLeftRight' block. The input is 'kRollRateWarning' with a value of (15.0, 15.0). The block outputs 'rollRate' with a value of -18.189 to two destinations.

On the right, a 'Graphical Panel - RollRateWarning' window is shown. It features a central 'Roll Rate' gauge with a needle pointing to -18.19. The gauge has a scale from -50.00 to 50.00. To the left of the gauge is a red 'LED' labeled 'Left Warning', and to the right is a grey 'LED' labeled 'Right Warning'.

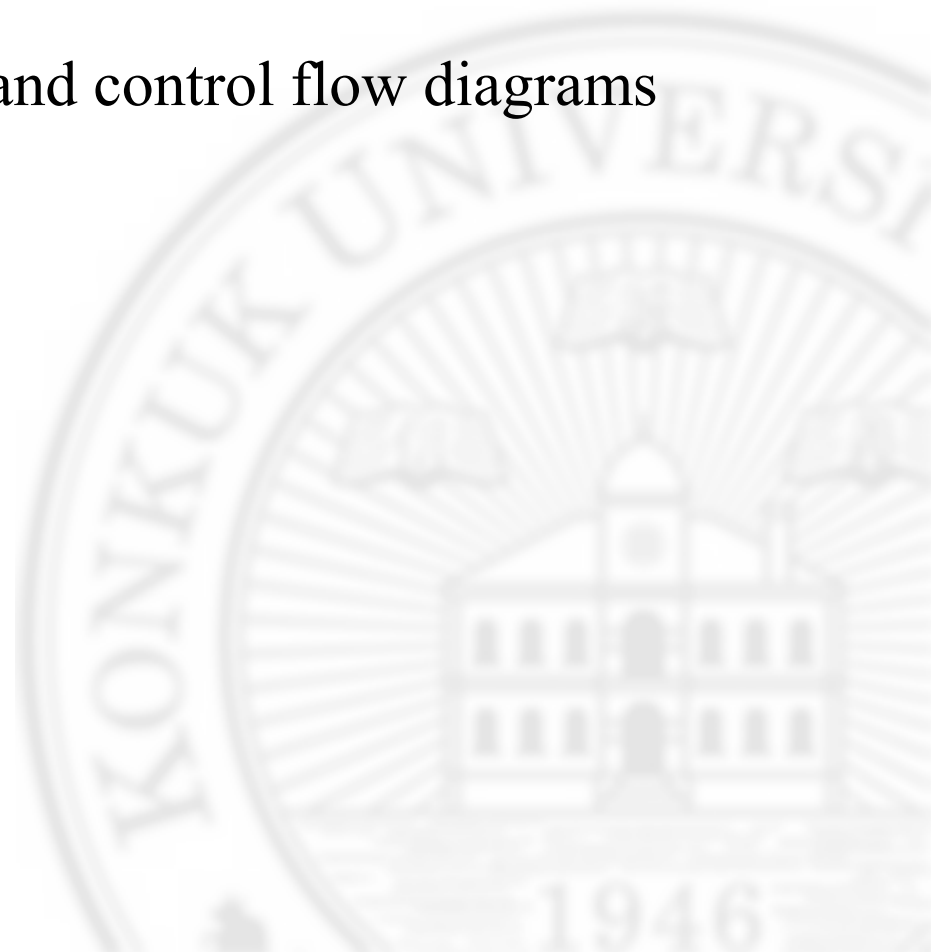
At the bottom of the MATLAB interface, there is a 'Variable' table and a 'Cycle' table.

Variable	Value	Path
----------	-------	------

Cycle	Action
0-2	User In

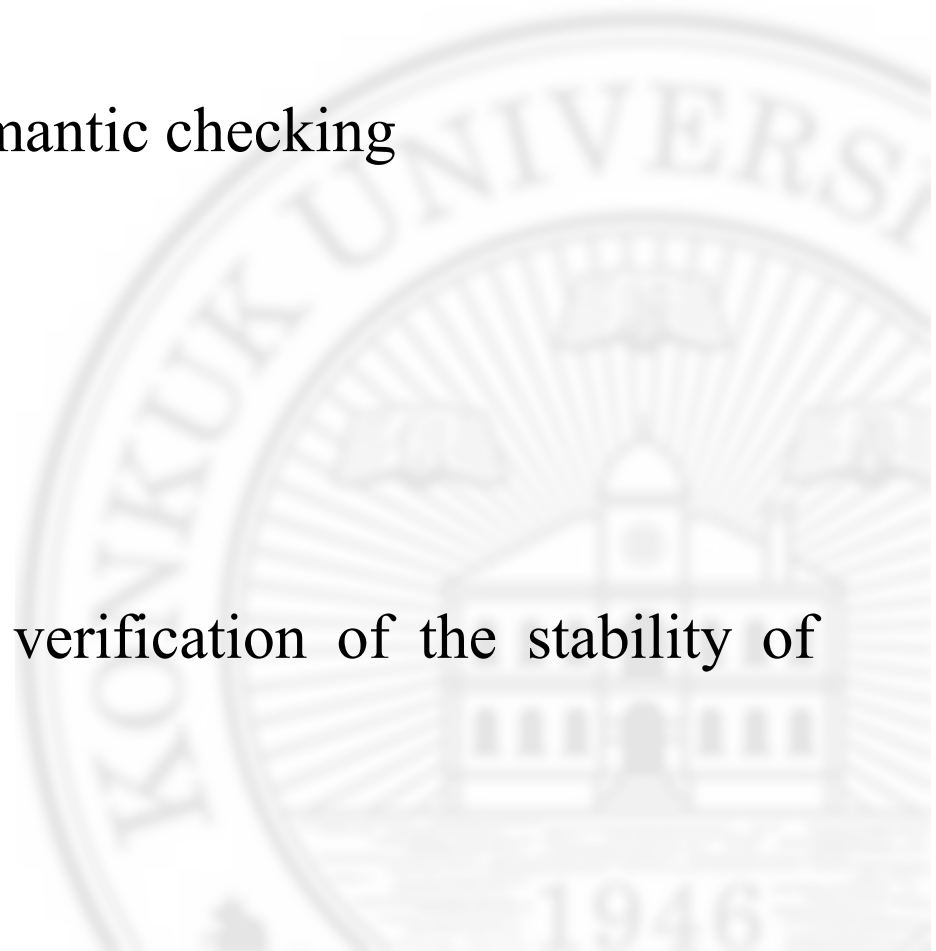
Report Generation

- Automatic report generation
- Shows the data flow diagrams and control flow diagrams
- Represent tables



Conclusion

- Model verification of the functional design with data flow and control flow of the roll and pitch of airplane.
- Automatic code generation, semantic checking
- Verifying system correctness
- Simulation using scade suit
- Scade suite graphical display verification of the stability of airplane



References

[1] www.esterel-technologies.com/products/scade-suite/



Thank You!

