A Computer Aided Development Procedure to Test Multiple System Integration

Sabrina CORPINO (Politecnico di Torino S.P.A.I.C. srl)
Nicole VIOLA (Politecnico di Torino)
Marco FIORITI (Politecnico di Torino)
Giovanni MEDICI (Politecnico di Torino)

The work has been performed through a close cooperation with ALENIA AERONAUTICA staff with constant technical meetings and a continuous information exchange.In particular Authors wish to thank eng. Maria Airoldi, Mr. Marco Mantovani eng. Alessandro Pasquino and eng. Massimiliano Paternoster.

SMAT-F1 Project

SMAT-F1 is a research project funded by the Regione Piemonte (Italy), managed by Finpiemonte and promoted through the Promoter Board of Piedmont's Aerospace District. It is also co-funded by European fund for regional development (F.E.S.R.).

SMAT-F1 Project

SMATF1

SMAT main objective is to define, design and develop an Advanced Environment Monitoring System, based on Unmanned Air Systems

The system will be able to cover different potential needs, such as:

–Surveillance of areas subject to natural disasters (landslides, floods, earthquakes, fires)

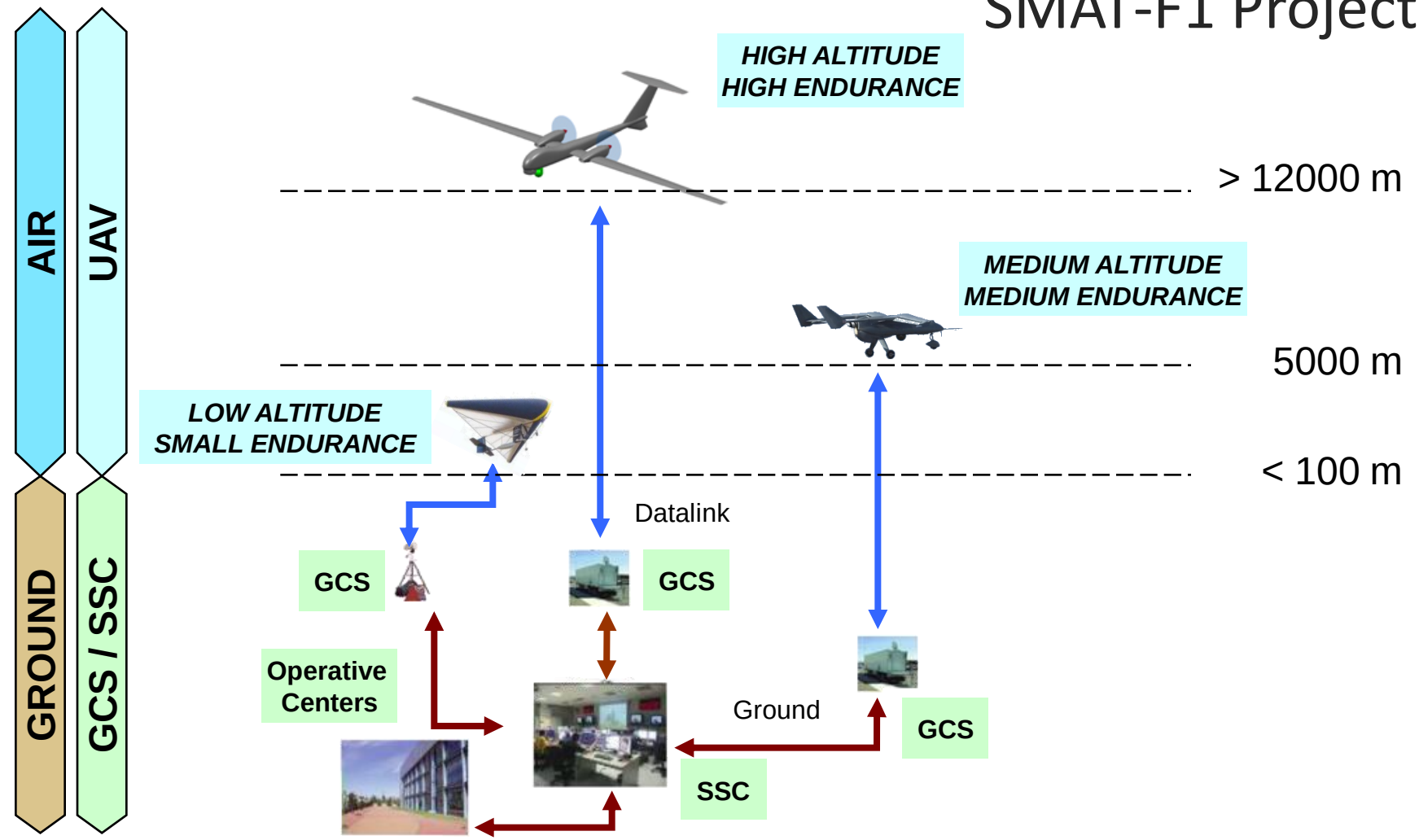
–Border patrol

–Surveillance of areas subject to human intervention.

–Specific areas monitoring for prevention purposes

–Territory surveillance for planning purposes

SMAT-F1 Project



The author's task

The author's have been asked to develop the control laws of various systems:

1. LDG Landing gear system
2. NWS Nose Wheel Steering system
3. WBS Wheel Braking System (2 configurations Hydraulic Electrical)
4. FS Fuel System

Computer Aided Development Procedure

During the systems controller logic definition, performed for the SMAT-F1 project, we used a three steps development method:

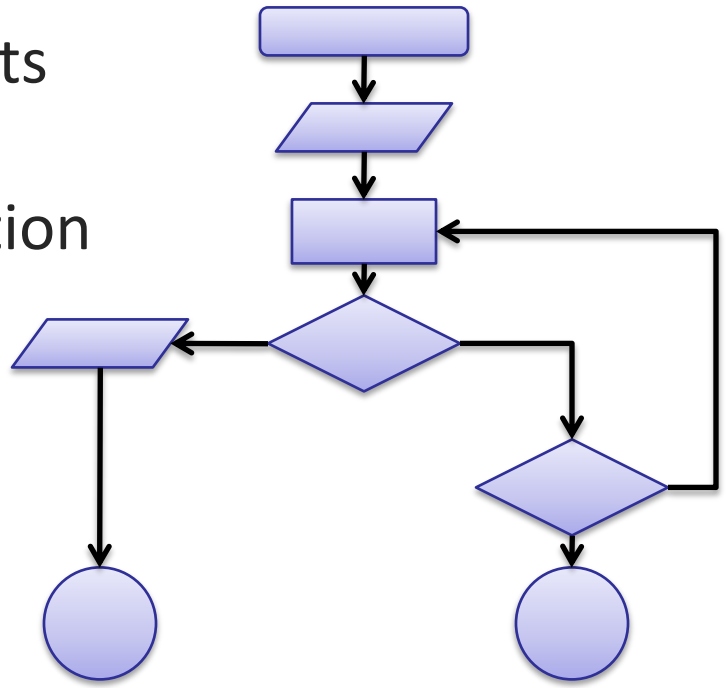
1. Algorithm development through visual flow charts technique.

2. According to the flow charts a code file (written in C programming language) has been developed.

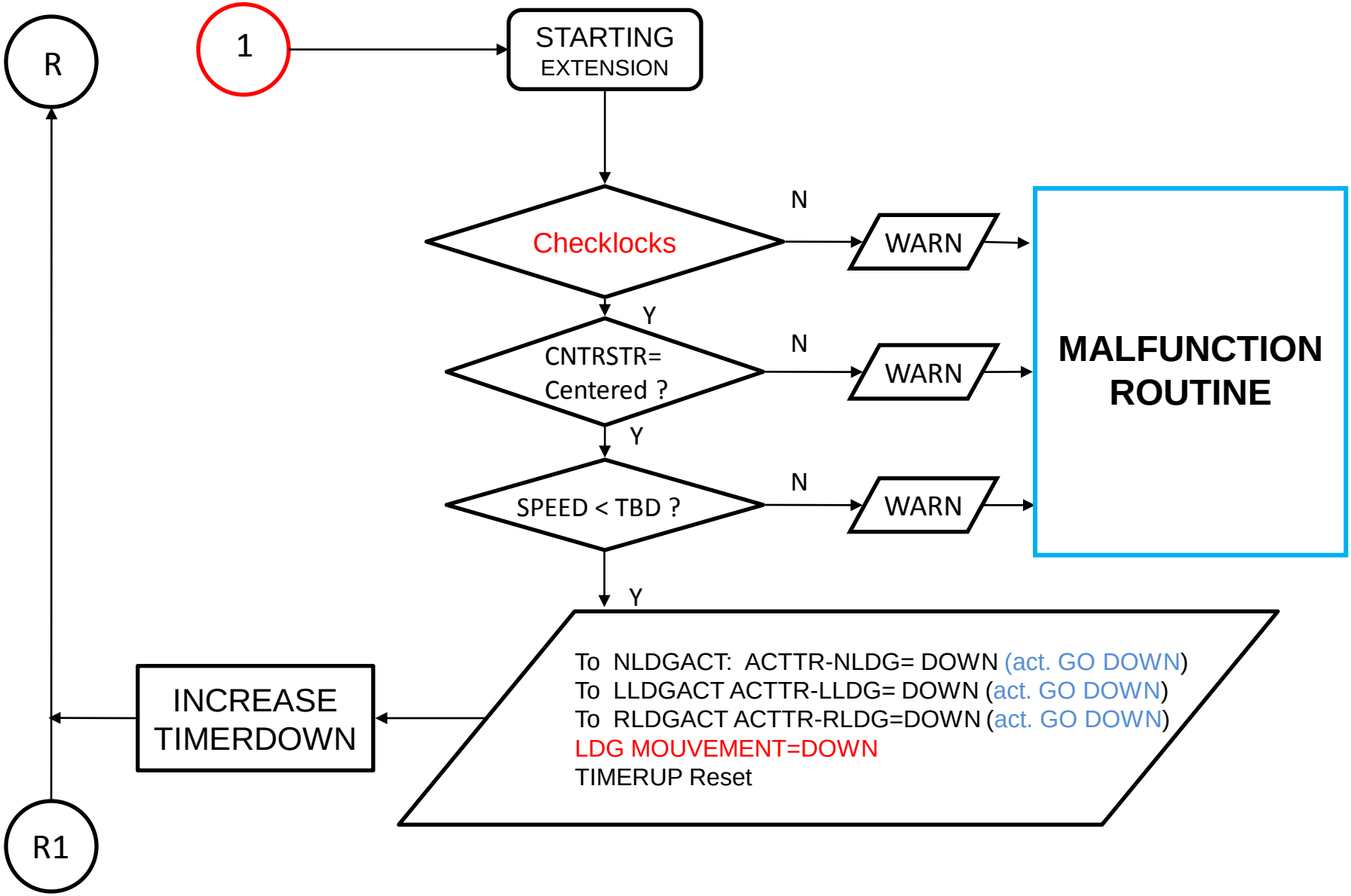
3. With the aim to demonstrate and validate the code a “Status Model” was also built.

Flowcharts

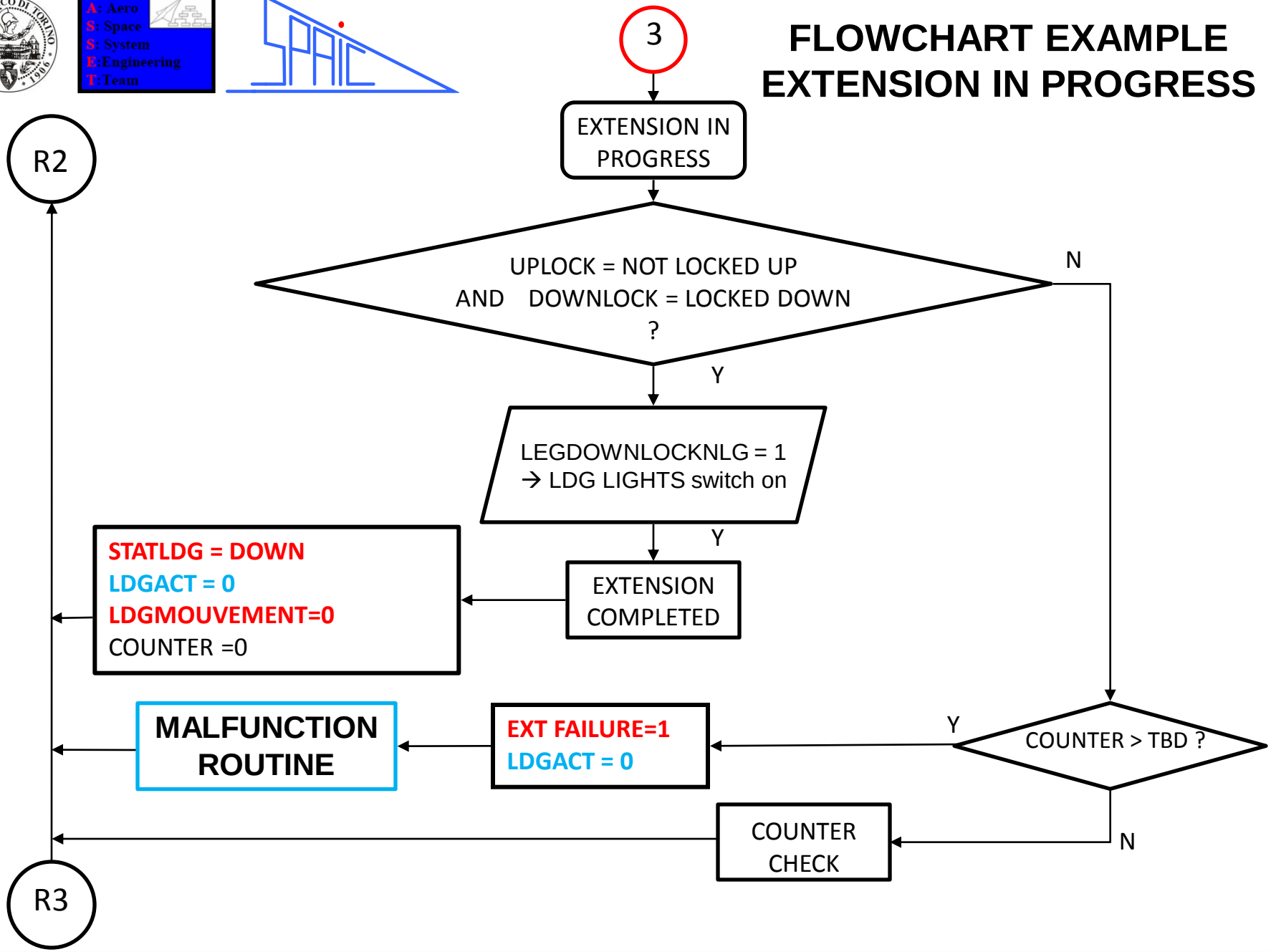
Initially the logics are developed thanks to the Flowcharts' diagrammatic representation.
In the next slides few flowcharts examples will be shown.
This diagrammatic representation immediately gives visually the step-by-step solution to a given problem.



FLOWCHART EXAMPLE
STARTING EXTENSION



FLOWCHART EXAMPLE
EXTENSION IN PROGRESS



2

ANSI C

The next step has been the recode of the flowcharts into C language code:

```
171
172 int FCC(int num) {
173
174     switch (num) {
175         case 1:
176             // ------------------------------------------------------
177             // ------------------------------------------------------
178             // ------------------+   F C C 1   +--------------------
179             // ------------------------------------------------------
180             // ------------------------------------------------------
181             // ------- ROUTINE EMERGENCY EXTENSION -----
182             if(EMEREXTENS==1)
183             {
184                 if(timer<0.1)
185                 {
186                     // The function around is here used in order to start and stop a counter
187                     // Since the command to the EMERGENCY INIBITION RELAY has to be received
188                     // by the relays, we set up a consolidation time of 0.1 s in the first examples
189                     // the around(CONDITION) is used, then starting from the Gear Command, just a
190                     // comment will be reported.
191
192                     around(START);
193
194                     EMEX_INIB_LDG[9] = "NOT-ARMED";
195
196                     timer=timer+around(STOP);
197
198                     return 0;
```

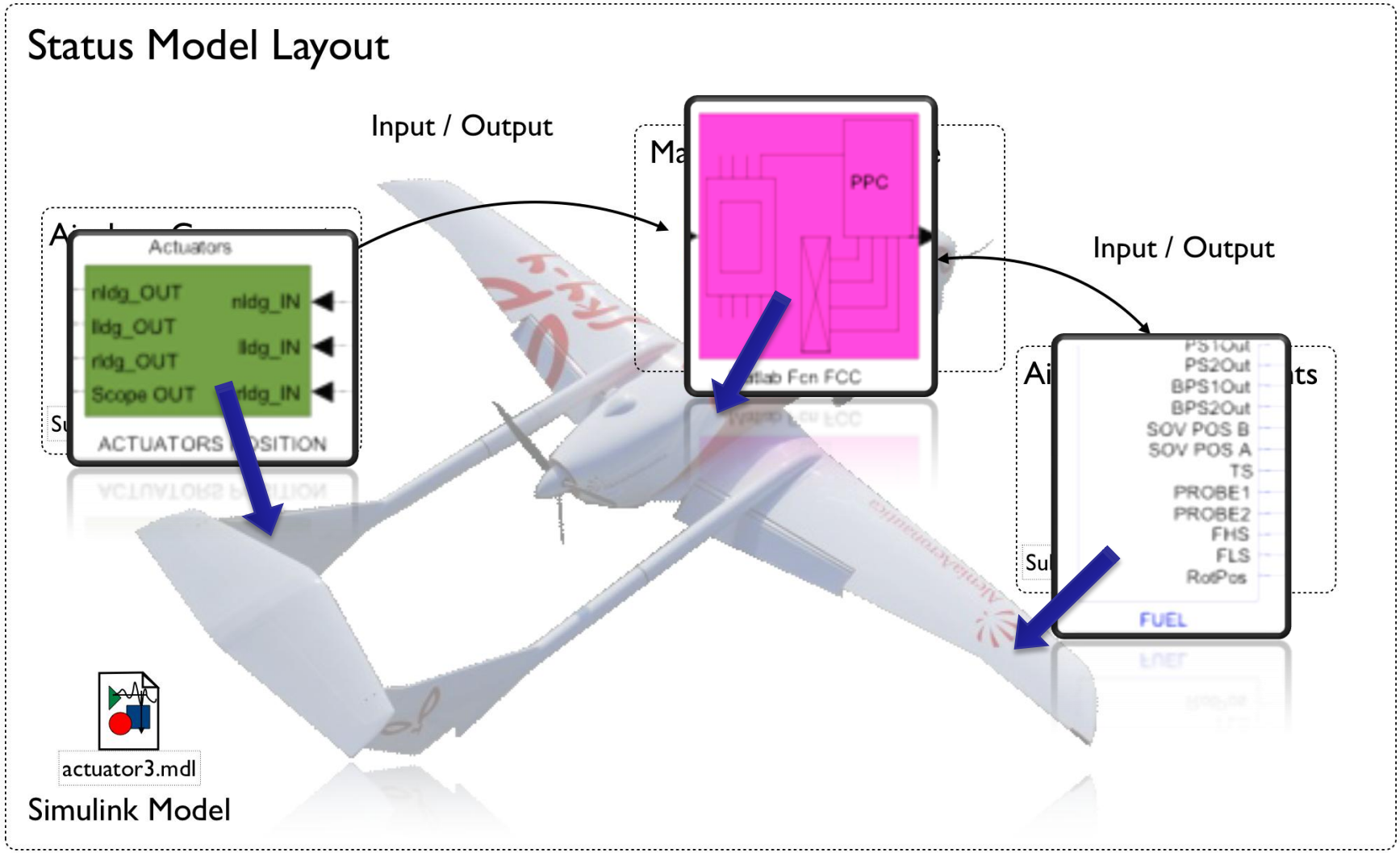
3

“Status Model” development:

The third step has been the recode of the flowcharts into Matlab® and Matlab Simulink ™ language code:

- The Status Model help to test the control laws dynamically.
- The block approach keep FCC code isolated from other components.
- Many different FAILURES can be simulated, among all:
  - Flight Control Computer FAILURE.
  - UPLOCK and/or DOWNLOCK FAILURE.
  - SLOW ACTUATOR or TOTAL ACTUATOR’s FAILURE.
  - WOW FAILURE.

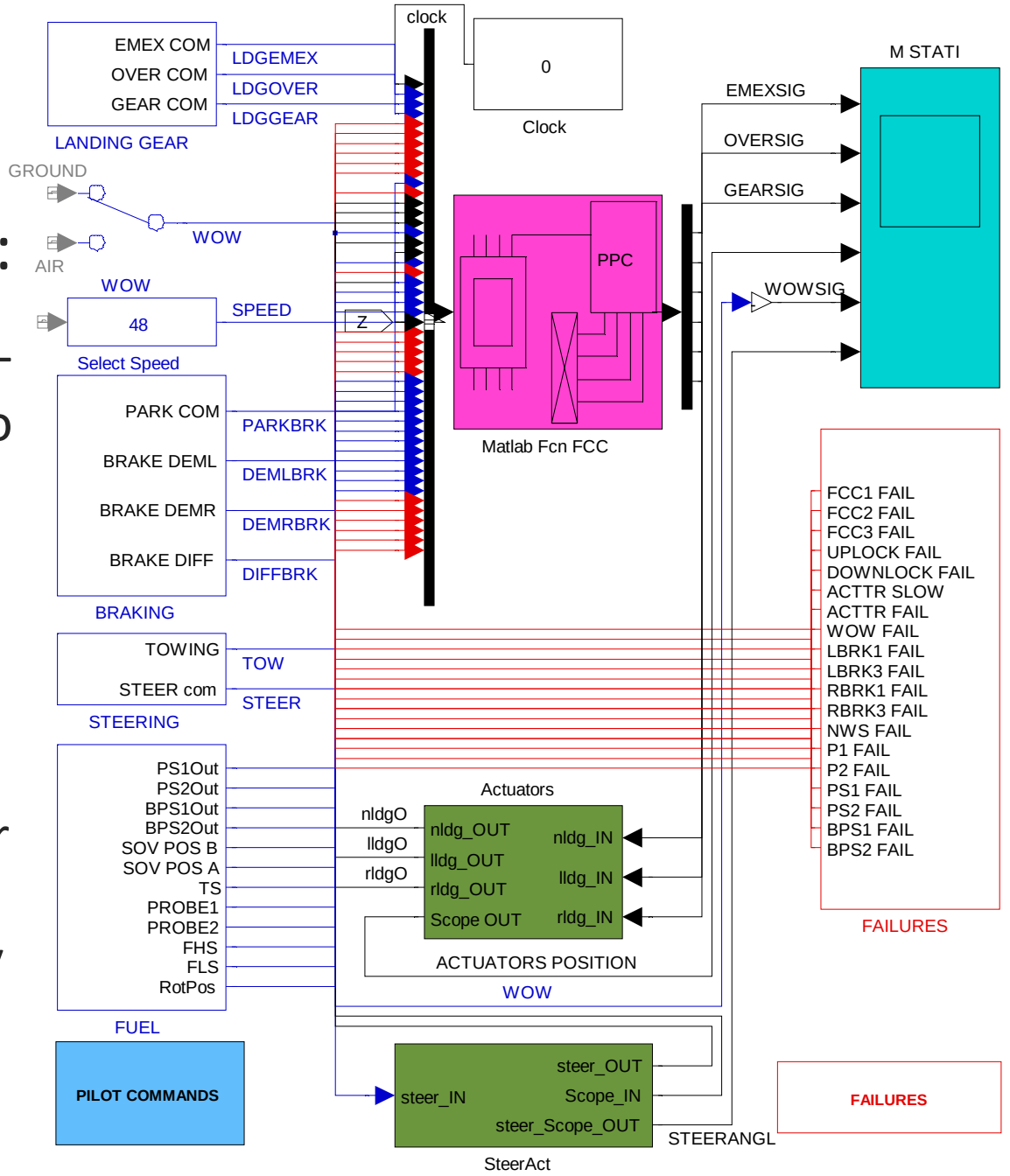
“Status Model” Layout



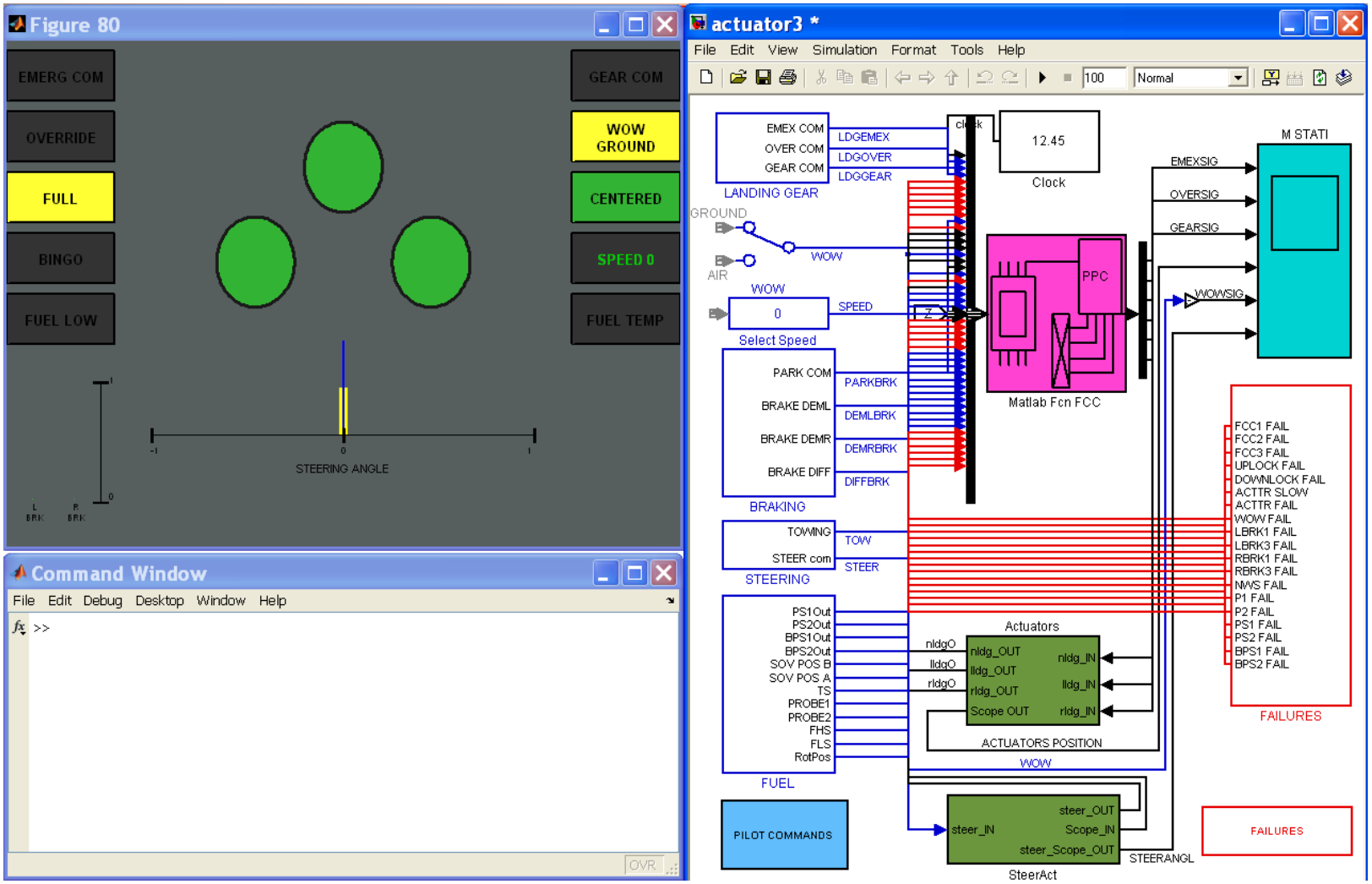
“Status Model” layout:

The Simulink model high-lights the signals related to each system:

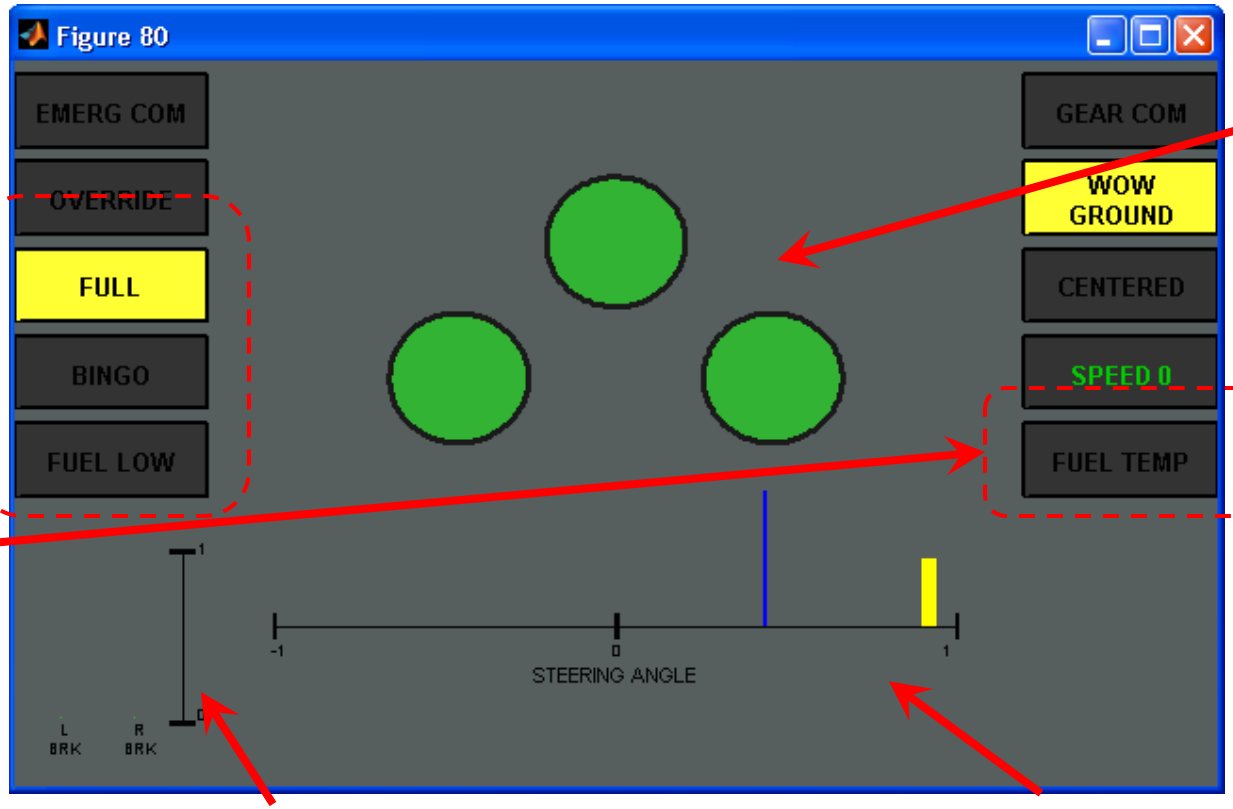
- Each system is contained inside a mask block, for cluttering reasons.
- Customized components (with dedicated icon) are easier to understand.
- Through custom icons an easy to use graphical user interface can be developed.



“Status Model” splash screen:



“Status Model” display window overview:



LDG STATUS LIGHTS

FS LIGHTS

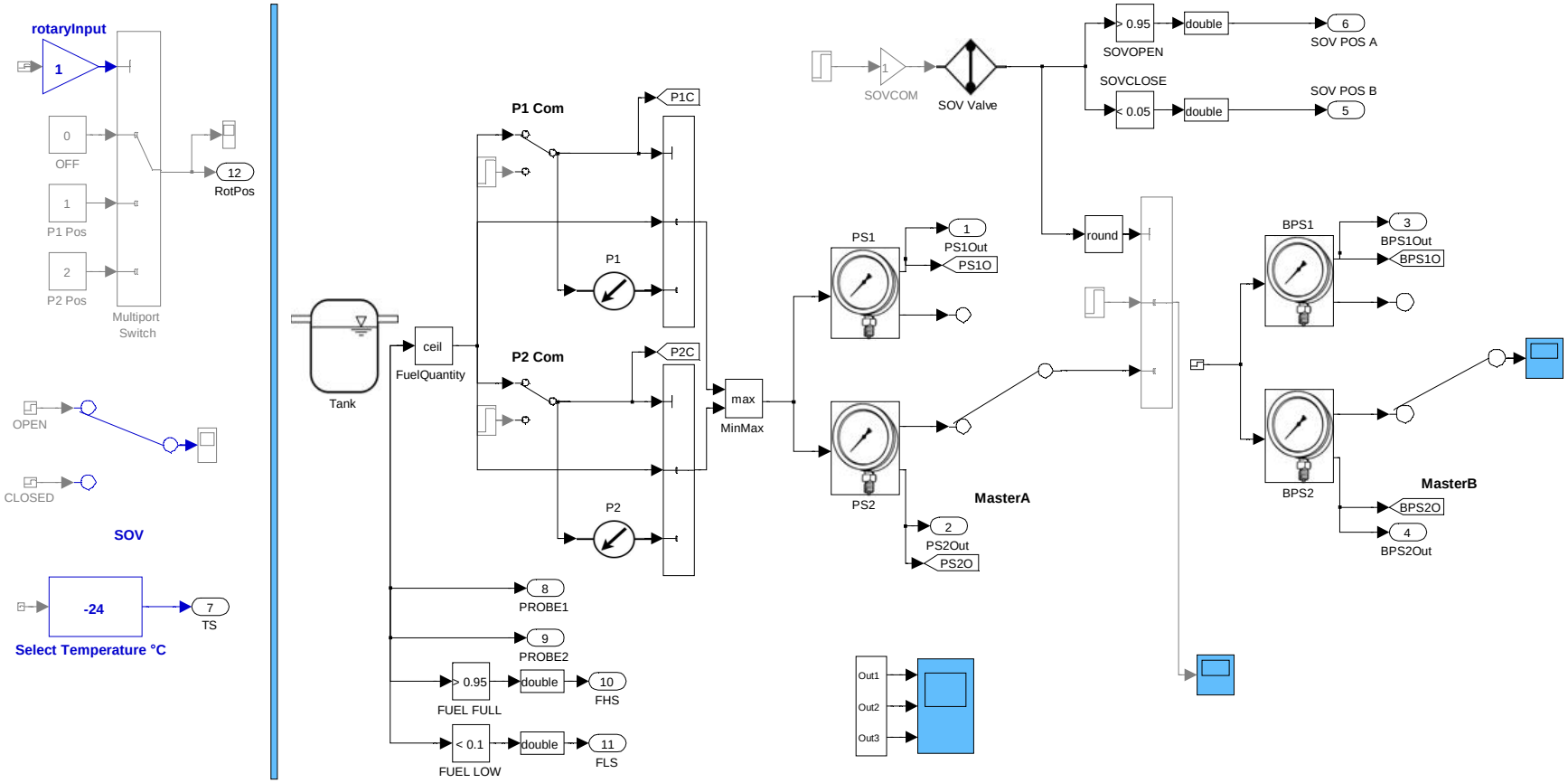
WBS BRAKES POWER

NWS STATUS INDICATOR

Command Window.

The user can dynamically change the position of Each switch and verify the software response. Each System: LDG, NWS, WBS, and FS can be adjusted by clicking on the proper mask.

“Fuel System” Simulink model.



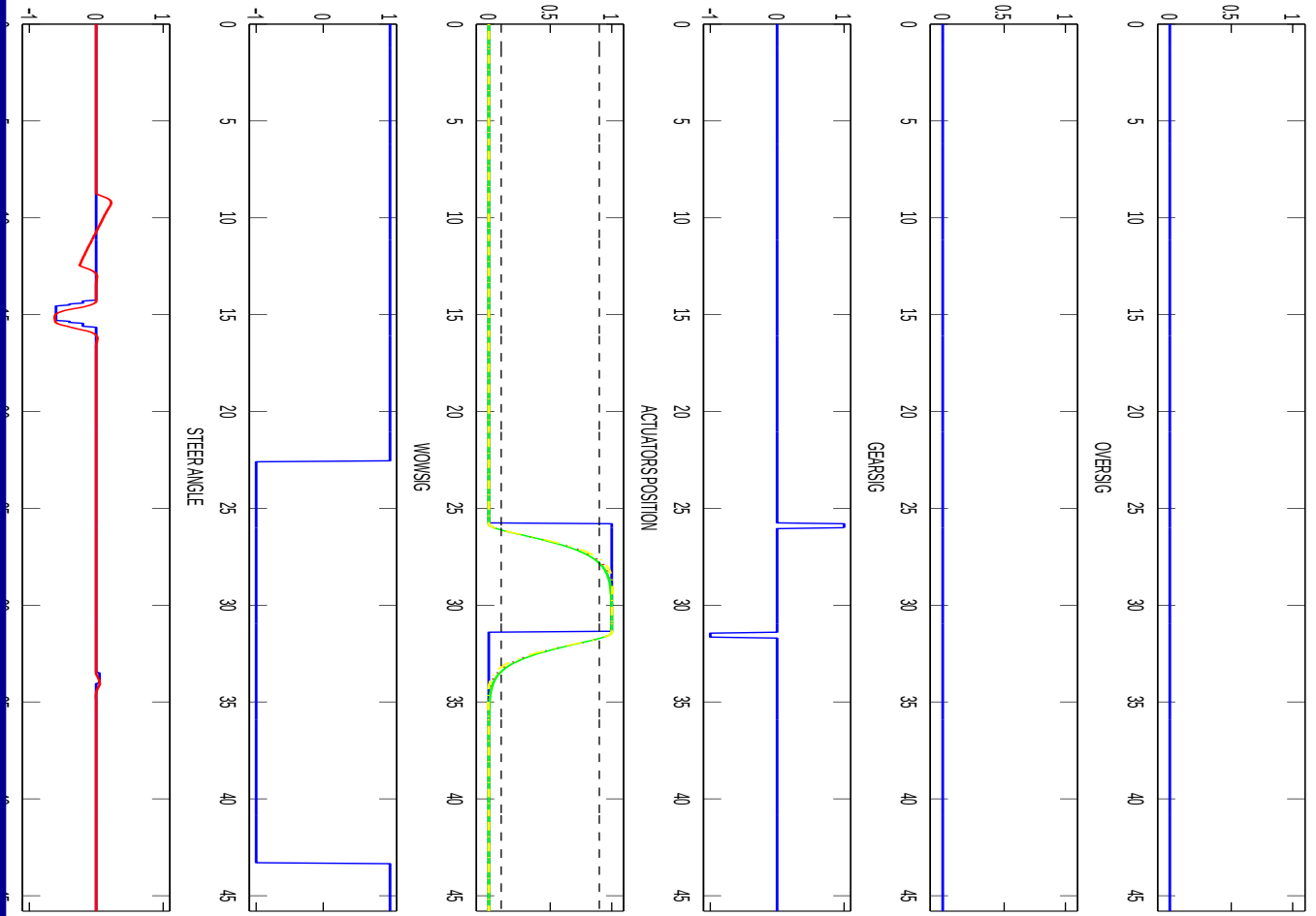
“Status Model” Ouputs

The user can verify the simulation results through various post-processing tools:

- Plots: reports the most important variables signals vs. time.

- Logs: collects the messages that are produced by the FCCs (Flight Control Computers).

“Status Model” plot example:



- Parking
- Towing
- Steer & Brakes
- Taxing
- Takeoff
- Retraction
- Extension
- Dyn BIT
- Steer BIT
- Landing

“Status Model” log example:

-+ PILOT COMMAND TCS GEAR UP +-
FCC1 ACTTR-NLDG=GO-UP
FCC1 ACTTR-LLDG=GO-UP
FCC1 ACTTR-RLDG=GO-UP
0.05
FCC2 ARM-REL-STR=NOT-ARM
FCC2 Stering System Powered OFF
0.40
  UPLCKNLDG=LOCKED-UP
INTUPLCKNLDG=LOCKED-UP
1.90
  UPLCKLLDG=LOCKED-UP
INTUPLCKLLDG=LOCKED-UP
2.00
  UPLCKRLDG=LOCKED-UP
INTUPLCKRLDG=LOCKED-UP
 GEAR IS UP
FCC2 ARM-REL-LDG=NOT-ARM and Check it
-+ PILOT COMMAND TCS GEAR DOWN +-
FCC1 ACTTR-NLDG=GO-DOWN
FCC1 ACTTR-LLDG=GO-DOWN
FCC1 ACTTR-RLDG=GO-DOWN
0.00
DOLCKNLDG=LOCKED-DOWN
INTDOLCKNLDG=LOCKED-DOWN
1.80
  DOLCKNLDG=LOCKED-DOWN
INTDOLCKNLDG=LOCKED-DOWN
1.85
  DOLCKLLDG=LOCKED-DOWN
INTDOLCKLLDG=LOCKED-DOWN
1.90
  DOLCKLLDG=LOCKED-DOWN
INTDOLCKLLDG=LOCKED-DOWN
1.95
  DOLCKRLDG=LOCKED-DOWN
INTDOLCKRLDG=LOCKED-DOWN
 GEAR IS DOWN
FCC2 ARM-REL-LDG=NOT-ARM and Check it
FCC2 ARM-REL-STR=ARM
FCC2 Stering System Powered ON
Dynamic Actuator Test
Dynamic Test Succeded
Center Steer Command Sent
Center Steer Command Received

- LDG Retraction
- LDG Extension

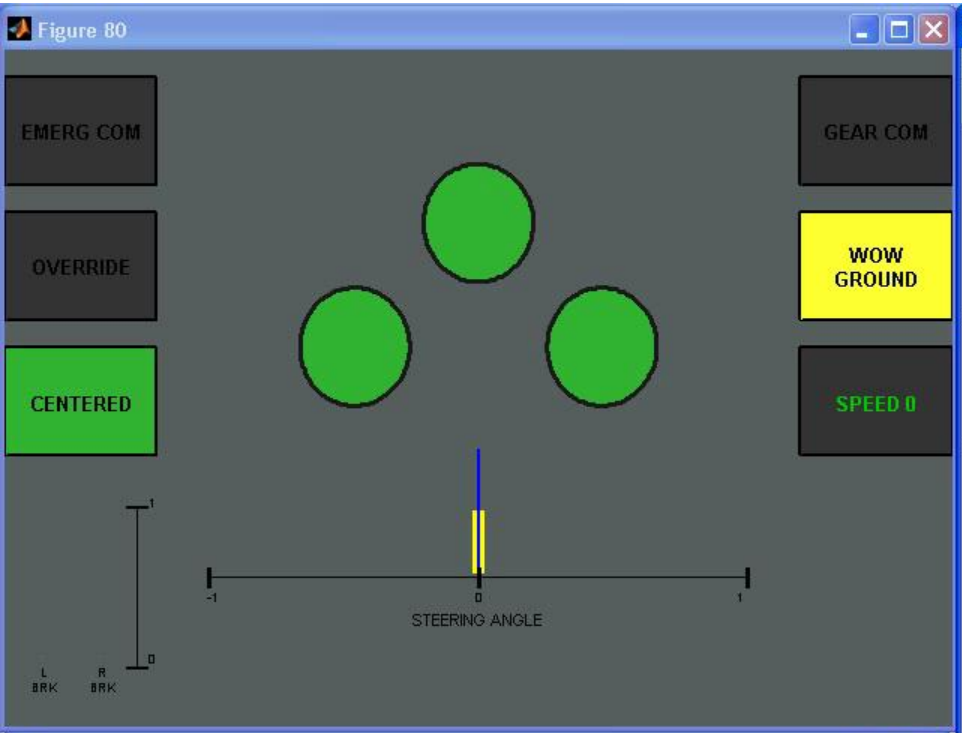
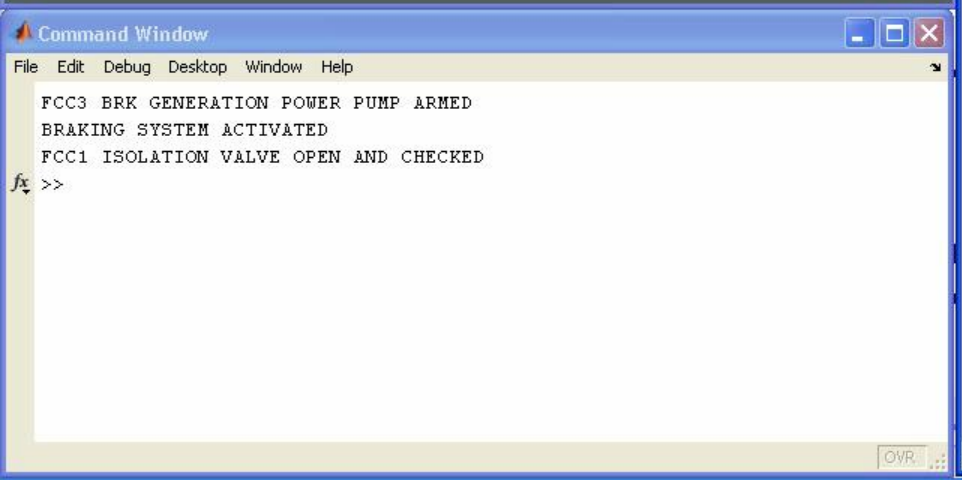
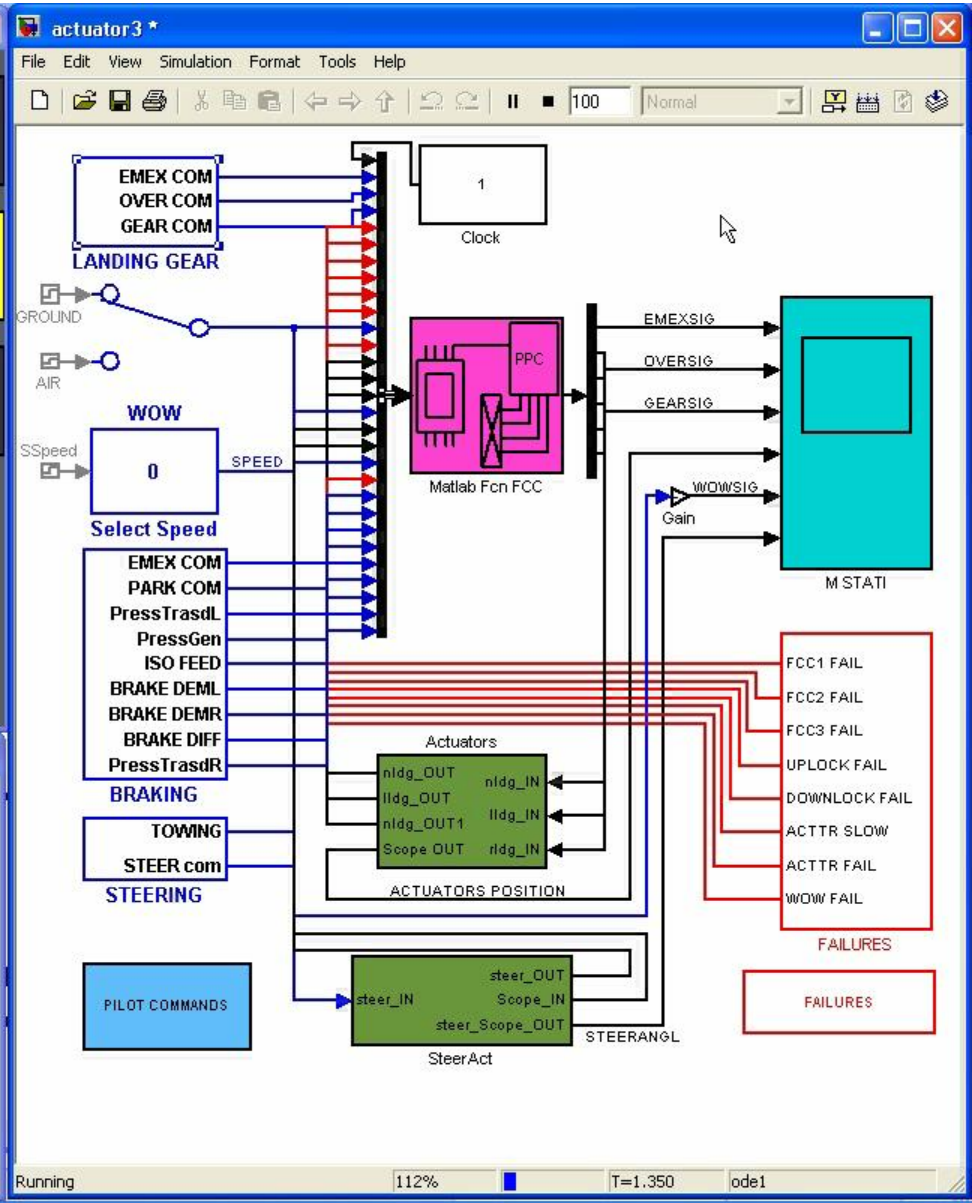


Figure 80
EMERG COM
OVERRIDE
CENTERED
GEAR COM
WOW GROUND
SPEED 0
STEERING ANGLE
L BRK
R BRK


Command Window
File Edit Debug Desktop Window Help
FCC3 BRK GENERATION POWER PUMP ARMED
BRAKING SYSTEM ACTIVATED
FCC1 ISOLATION VALVE OPEN AND CHECKED
>>


actuator3 *
File Edit View Simulation Format Tools Help
EMEX COM
OVER COM
GEAR COM
LANDING GEAR
GROUND
AIR
WOW
SSpeed
0
SPEED
Select Speed
EMEX COM
PARK COM
PressTrasdL
PressGen
ISO FEED
BRAKE DEML
BRAKE DEMR
BRAKE DIFF
PressTrasdR
BRAKING
TOWING
STEER com
STEERING
PILOT COMMANDS
1
Clock
PPC
Matlab Fcn FCC
EMEXSIG
OVERSIG
GEARSIG
WOWSIG
Gain
M STATI
Actuators
nldg_OUT
lldg_OUT
nldg_OUT1
Scope OUT
nldg_IN
lldg_IN
rldg_IN
ACTUATORS POSITION
FCC1 FAIL
FCC2 FAIL
FCC3 FAIL
UPLOCK FAIL
DOWNLOCK FAIL
ACTTR SLOW
ACTTR FAIL
WOW FAIL
FAILURES
steer_IN
steer_OUT
Scope_IN
steer_Scope_OUT
SteerAct
STEERANGL
FAILURES
Running
112%
T=1.350
ode1

Conclusions

The authors developed a Computer Aided Development Procedure to Test Multiple System Integration.

1. The modular nature of the Status Model enables the creation of a customized blocks library.
2. Various Failure scenarios can be simulated and tested.
3. The tool allows faster and more reliable controller logics development.
4. It prooved to be effective helping the specialist to release adequate FRD thanks to the three step procedure.

THANK YOU, ANY QUESTION?