

# Landing Gear System

Prendiamo il caso di studio

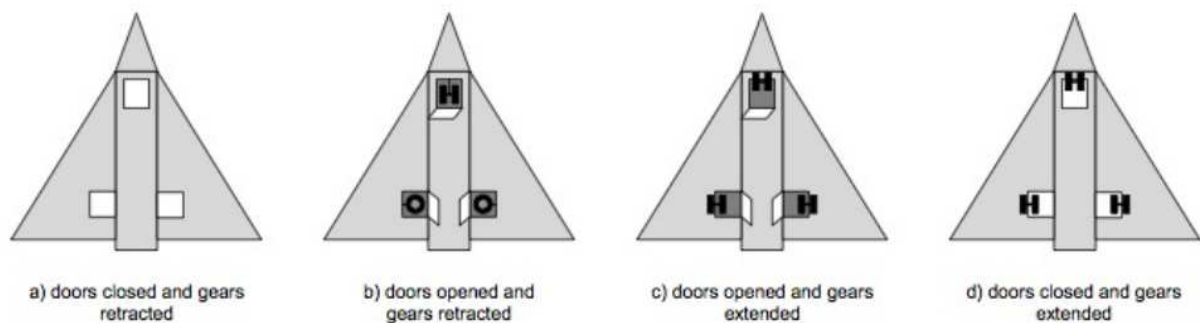
[http://www.irit.fr/ABZ2014/landing\\_system.pdf](http://www.irit.fr/ABZ2014/landing_system.pdf)

Applichiamo ad alcune (2) sue parti le diverse tecniche che abbiamo studiato.

Ci concentriamo su due componenti

## 1. Comando LGS da parte del pilota

In nominal mode, the landing sequence is: open the doors of the landing gear boxes, extend the landing gears and close the doors. This sequence is illustrated in Figure 2. After taking off, the retraction sequence to be performed is: open the doors, retract the landing gears and close the doors.



To command the retraction and outgoing of gears, an Up/Down handle is provided to the pilot. When the handle is switched to “Up” the retracting landing gear sequence is executed, when the handle is switched to “Down” the extending landing gear sequence is executed.

## 2. Sensor monitoring

Each sensor is triplicated. The first activity of the control software is to select one of these three values. Let us call  $X$  a sensor and  $X_i(t)$   $i = 1, 2, 3$  the three values of  $X$  received at time  $t$ :

- If at  $t$  the three channels are considered as valid and are equal, then the value considered by the control software is this common value.
- If at  $t$  one channel is different from the two others for the first time (i.e., the three channels were considered as valid up to  $t$ ), then this channel is considered as invalid and is definitely eliminated. Only the two remaining channels are considered in the future. At time  $t$ , the value considered by the control software is the common value of the two remaining channels.
- If a channel has been eliminated previously, and if at  $t$  the two remaining channels are not equal, then the sensor is definitely considered as invalid.

An anomaly is detected each time a sensor is definitely considered as invalid.

// non viene detto cosa succede se sono tutti e tre diversi

## Requisiti

Requirement R1:

- (R11 bis) When the command line is working (normal mode), if the landing gear command handle has been pushed DOWN and stays DOWN, then eventually the gears will be locked down and the doors will be seen closed;
- (R12 bis) When the command line is working (normal mode), if the landing gear command handle has been pushed UP and stays UP, then eventually the gears will be locked retracted and the doors will be seen closed.

Requirement R2:

- (R21 ) When the command line is working (normal mode), if the landing gear command handle remains in the DOWN position, then retraction sequence is not observed.
- (R22 ) When the command line is working (normal mode), if the landing gear command handle remains in the UP position, then outgoing sequence is not observed.

// non ci sono requisiti per il sensor voting

## Compito

In un file word annota tutto quello che hai fatto.

### 1. Testing basato su programmi

Scrivi l'implementazione in Java del sensor voting. Se fai delle assunzioni spiegate.

Scrivi i test Junit per la copertura istruzioni, condizioni, decisioni, MCDC.  
Esegui e valuta la copertura.

Commenta nei test Junit cosa fanno i diversi test e che copertura ti permettono di ottenere.

Prova a generare casi di test on con Randoop e valuta i casi di test generati.

### 2. DBC/JML

Copia la classe sensor voting e aggiungi i contratti JML.

Prova con i contratti JML con una classe main in cui chiami i diversi metodi.

Prova anche a modificare il codice e controlla che i contratti siano violati. Documenta bene le violazioni e le loro cause in commenti.

### 2b verifica dei programmi (key hoare)

Un nuovo progetto, dimostra la correttezza dei contratti per il sensor voting

### 3. NUMSV.

Scrivi il modello NuSMV per entrambi i moduli.

Scrivi un po' proprietà – seguendo il testo dei requisiti.

Aggiungi proprietà che ti sembrano interessanti, spiegale bene, e provale.

Trova anche qualche proprietà giustamente falsa e commenta lo scenario in cui viene falsificata.

## 4a MBT con le FSM

Descrivi la FSM (con un disegno) per il landing gear system

Testala in modalità offline (genera i casi di test). Implementala in modelJunit e caricala. Salva il disegno della macchina come immagine.

Genera i casi di test con modelJunit e salvali con TXT

Estensione: prova anche con il voting system e test online.

## 4b Input domain modelling

Applica Input domain modelling sensor voting.

# SOLUZIONE

## 1. testing basato sui programmi

Ho scritto il codice Java per il voting system. Ho assunto che il channel sia un boolean.

Ho scritto i casi di test per le diverse coperture.

MCDC della condizione valid1 & valid2 & valid3.

...

Le coperture sono indicate nei casi di test.

Ho esteso con il generico. Ho dovuto assumere che se sono tutti e tre diversi ho comunque invalido il sensore.

Ho generato 20000 casi con randoop e ho ottenuto 38% di copertura...

## 2. DBC

- messi i contratti e spiegati per il sensor voting

- KEY: provati i contratti per il sensor voting

### 3. NuSMV

Scritto il modello nusmv del lgs. Provate un po' di proprietà

#### 4a. FSM

Ho implementato la FSM del

#### 4b. IDM al sensor voting

Ho implementato il sensor voting come interi.

- **interface based:** considero come input le tripe di valori interi. Il singolo dominio intero lo diviso in tre partizioni NEGATIVO, ZERO, POSITIVO. L'intero dominio saranno 27 partizioni:

...

- **functionality based:** anche in questo caso considero come input le tripe di valori interi. Per diviso l'insieme nelle seguenti partizioni

partizione	funzionalità	s1	s2	s3
p1	I tre valori sono tutti e tre uguali	0	0	0
p2	I tre valori sono uguali tranne uno	10	10	9
p3	I tre valori sono tutti e tre diversi	10	9	8

Potrei in verità divide la partizione p2 in tre partizioni

partizione	funzionalità	s1	s2	s3
p21	I tre valori sono uguali tranne il primo	10	0	0
p22	I tre valori sono uguali tranne il secondo	0	10	0
p32	I tre valori sono uguali tranne il terzo	0	0	10

#### Combinatorial testing

- versione piu' semplice: scrivo il modello con i tre numeri ognuno. Ho 9 casi si test

- versione piu' complicata: aggiungo il valore di validità del sensore e i constraint per dargli il valore giusto.

- versione ancora piu' complicata: ogni input può assumere due valori, ma in in due istanti di tempo successivi.

- ho generato i casi di test e costruito i casi Junit Parametrici.