Flight Control

# Initial Project Setup

The flight control tool requires the following files to run: an OperatingCondition file, at least one Synthesis file, and at least one Requirement file.

## OperatingCondition File

The OperatingCondition file will be generated from the FLIGHT Dynamics tool.

## Synthesis File

The Synthesis file will be create using the Object editor. There are two ways to open the object editor for creating a synthesis file.

1. In the tool ribbon – “New” -> “Synthesis”
2. In the tree – Right click the Synthesis node and choose “Insert Synthesis Object”

Once the synthesis object editor is open you will need to specify three items;

1. The title that will show up in the tree.
2. The model used to synthesize the gains.
3. The script used to synthesize the gains.

## Requirements File

Currently there are four types of Requirements files a user can create: Stability, Frequency Response, Handling Qualities, and Aeroservoelasticity. There are two ways to open the object editor for creating a requirements files.

1. In the tool ribbon – “New” -> “Requirement” -> “Stability” or “Frequency Response” or “Handling Qualities” or “Aeroservoelasticity”
2. In the tree – Right click the corresponding node and choose “Insert (Object Type) Object”

Once the object editor is opened you will need to specify:

Method – Used to operate on each iteration of the run function

Model Name – The model used to generate parameters on each iteration.

Title – The title that shows up in the tree.

Output Data Index – If the method it used in more than one file then this will determine which output corresponds to this requirement.

Requirement Plot – The plot that shows in the background and displays requirement information to the user.

Background Plot – The plot that shows in the background, similar to the requirement plot.

# Using the Flight Control Tool

You will be able to reload all of the set up at any time once you have created the initial configuration. However the first time you will need to load all necessary files.

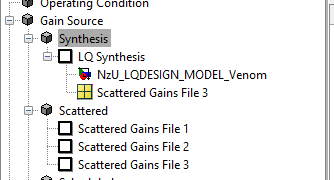
Do the following:

1. Load the OperatingCondition file.
   1. Tool Ribbon – “Load” -> “Operating Condition”
   2. Tree – Right click Operating Condition node and choose “Insert Operating Condition”
2. Load the gain source. The gains can come from one of three places: Synthesized by a method, previously created scattered gains file, or a scheduled gains file. Note: All three can be loaded at one time and the user can select between them.
   1. To Load a Synthesis
      1. Tool Ribbon – “Load” -> “Source Gain” -> “Synthesis”
      2. Tree – Right click Synthesis node and choose “Insert Synthesis Object”
   2. To Load a Scattered gain file
      1. Tool Ribbon – “Load” -> “Source Gain” -> “Scattered”
      2. Tree – Right click Scattered node and choose “Insert Scattered Gain Object”
   3. To Load a Scheduled Gain
      1. Tool Ribbon – “Load” -> “Source Gain” -> “Scheduled”
      2. Tree – Right click Scheduled node and choose “Insert Scheduled Gain Object”
3. Load the requirements. .
   1. To Load a Synthesis
      1. Tool Ribbon – “Load” -> “Source Gain” -> “Synthesis”
      2. Tree – Right click Synthesis node and choose “Insert Synthesis Object”
   2. To Load a Scattered gain file
      1. Tool Ribbon – “Load” -> “Source Gain” -> “Scattered”
      2. Tree – Right click Scattered node and choose “Insert Scattered Gain Object”
   3. To Load a Scheduled Gain
      1. Tool Ribbon – “Load” -> “Source Gain” -> “Scheduled”
      2. Tree – Right click Scheduled node and choose “Insert Scheduled Gain Object”

How gains are handled:

When a new Synthesis object is added a new scattered gains file is also added and is named “Scattered Gains 1,2,3…” by default.

The gains are automatically stored in a scattered gains object. In order to know which scattered gains object the user can look at what is displayed under the chosen synthesis node. For example below the User has chosen “LQ Synthesis” as his synthesis method and the gains are being stored in “Scattered Gains File 3”.



How the gains are stored:

The gains are stored automatically and will be overwritten if these conditions are met:

* The design operating conditions are equal.
  + The design operating conditions are equal if:
    - Flight Conditions are equal.
      * Flight Conditions are equal if:
        + Mach is equivalent
        + Qbar is equivalent
        + Alt is equivalent
        + KCAS is equivalent
        + TAS is equivalent
    - Mass Properties are equal.
      * Mass Properties are equal if:
        + All defined parameters are equivalent and..
        + Weight Code is equivalent
        + Label can be different

If not then they are appended.

The gains are stored in a Gain File Object. The Gain file object has properties of Name, Date, and ScatteredGainCollection. The ScatteredGainCollection is an array of GainCollection objects. Each GainCollection object has the properties: DesignOperatingConditon, Gain, DesignParameter, and Date. The DesignOperatingConditon is a 1x1 object of class OperatingCondtion that was selected by the user to synthesize the gains. The Gain Object is a 1xN collection of ScatteredGain.Gain objects. The Design Parameter is a 1xN array of ScatteredGain.Parameter. The DesignParameter is a collection of all variables that were selected within the tool to appear in the “Requirements Parameter” area.

Gain Storage:

* SelectedScatteredGainFileObj (GainFile) 1x1
  + Name – String containing file name
  + Date – String containing the date the file was created
  + ScatteredGainCollection(ScatteredGain.GainCollection) 1xNd
    - Date – String containing the date the file was created.
    - DesignOperatingConditon(lacm.OperatingCondition) 1x1
      * Label(String)
      * ModelName(String)
      * States: [1xNS lacm.Condition]
      * Inputs: [1xNI lacm.Condition]
      * Outputs: [1xNO lacm.Condition]
      * StateDerivs: [1xNSD lacm.Condition]
      * FlightCondition: [1x1 lacm.FlightCondition]
      * MassProperties: [1x1 lacm.MassProperties]
      * LinearModel: [1xNLM lacm.LinearModel]
      * TrimSettings: [1x1 lacm.TrimSettings]
      * SuccessfulTrim(Boolean)
      * Cost(Double)
    - DesignParameter(ScatteredGain.Parameter) 1xNDP
      * Name(String)
      * Value(Double)
    - Gain(ScatteredGain.Gain) 1xNG
      * Name(String)
      * Value(Double)

Running with a Scattered Gains file selected:

When running with a scattered gains file as the gain source the gain comes directly from the scattered gains file and not from the synthesis. The tool will check the selected design model for a corresponding design model gain. If one exists it will run with the attached gains. If one does not exist the tool will inform the user.

Running with a Scheduled Gains file selected:

When running with a scheduled gain selected the gain for each selected analysis model will be linearly interpolated from the scheduled gains file.