Simplified Test Model of the M600, i.e. T600

# Overview and Philosophy of the T600

This short document provides a simple guide to the MBdyn Input File, which is a collection of text (ASCII) files. It is based on the simplified model of the M600, which does not have the correct properties in terms of mass and stiffness, but it may very well capture the overall geometric dimensions of the actual M600 and therefore can be run coupled to a KiteAD Input File of the M600.

Note that the mass distribution is fictitiously constant as is the stiffness distribution. This is where most of the effort of the end-user should go, i.e. defining lumped masses at the 3nodes for each beam and 6x6 stiffness matrices (K) for the two cross-sections at the Gaussian points (intermediate points between node 1 and 2, and between nodes 2 and 3 of each beam).

Additionally, for this model, an attempt was made to retain a small number of structural nodes for computational speed and trouble-shooting ease.

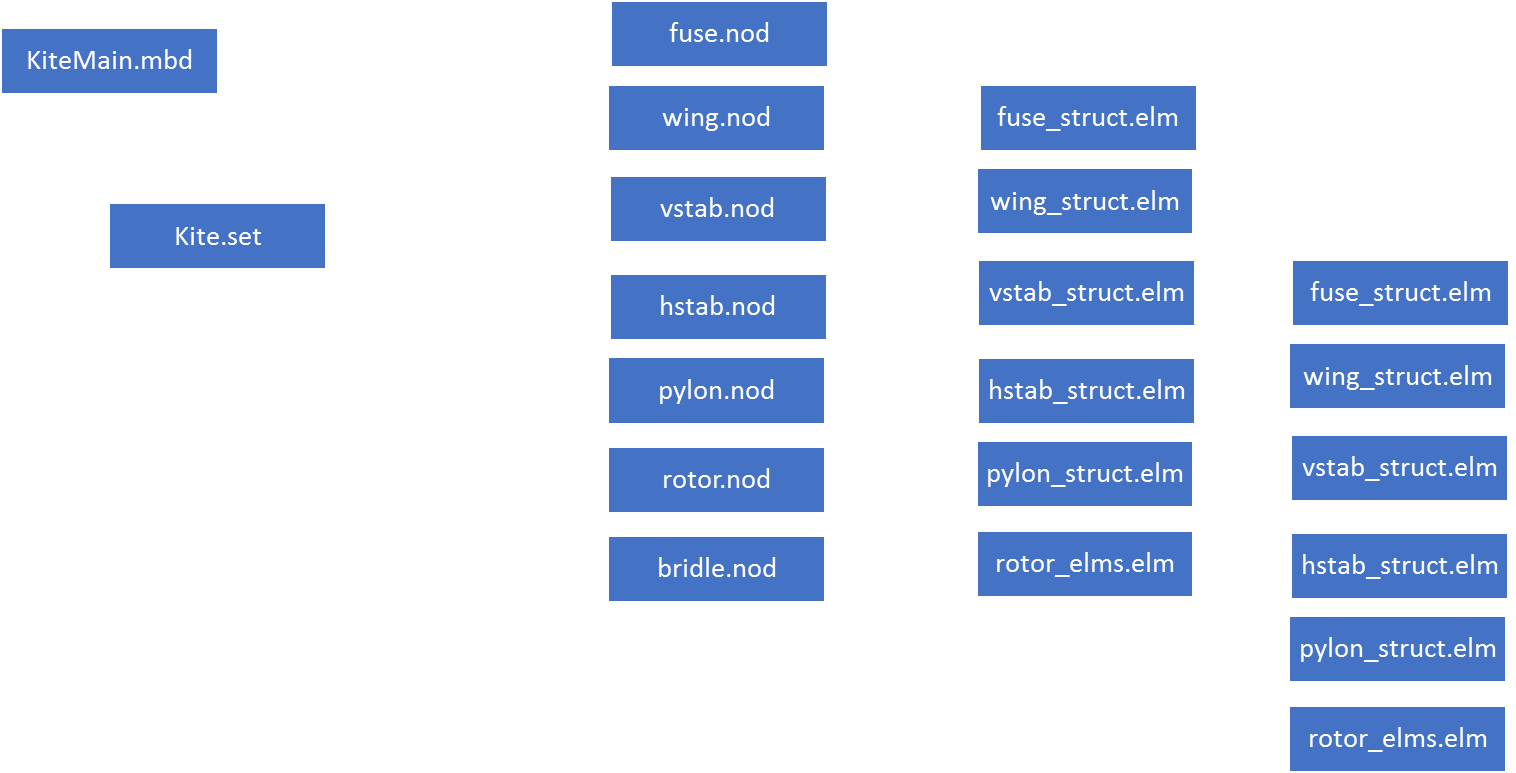
Finally, the nodes are supposed to lie along the line connecting the shear centers. Arbitrarily, this was achieved by shifting by 1/4c upwind all the aeronodes given in KiteAD.,inp. This is just to show the mapping capabilities, and it could be adjusted following the correct locations of the shear centers.

This model contains some curved beams, which were created for the fuselage to show the potential advantage in using fewer nodes along a non-rectilinear reference. The preprocessor, however, will create straight beam elements, therefore there will be a segment-wise organization of straight beams in the longer term.

The organization of the Input File follows common practice among MBdyn users, but it could be replaced by any other preferred organization, as long as the syntax and the data blocks are adhered to per the user’s manual that can be downloaded at:

<https://www.mbdyn.org/?Software_Download>

# File Structure



The main input file is KiteMain.mbd. It contains parameters for the simulation (dt, t0, tfinal) etc..

The ‘data’ block should not be touched.

The ‘initial value’ block has simulation control parameters; most of these parameters should only be changed if convergence problems arise. Some statements could be uncommented for debugging purposes to see standard output messages and especially to see whether connections among nodes and joints are ok (=0 ok, >>0 potential problems).

The ‘control data’ should not be changed, except to indicate different output choices. This block instantiates arrays for nodes, bodies, joints, beams. It inherits Kite.set constants (e.g. lwnNodes, i.e. the number of nodes in the port wing). If you intend to change these numbers, edit Kite.set.

The ‘nodes’ block calls include statements for the various ‘.nod’ files. If you intend to change the geometry of the model, or the resolution of the model for various components, then edit the respective ‘<component>.nod’ files and potentially the Kite.set file for constants.

The ‘elements’ block, should not be changed as long the joint locations and the respective nodes that make a joint do not change name. Each node (element, beam, etc.) can be given a name (string), which is then connected to an integer ID. This makes it convenient to refer to particular nodes and to use in joints: e.g.:

joint: vsltip\_node , total joint,

vsltip\_node,

rhsroot\_node,

position, reference, other node, null,

[…]

This ‘total joint’ is labeled ‘vsltip\_node’ (for convenience it is taking the same ID as the vsltip\_node’s), and connects the tip of the vertical stabilizer(vsltip\_node) with the root of the starboard horizontal stab (rhsroot\_node ).

Other joints create hinges between pylon nodes and rotor nodes.

The Structural definition of the beams are given through ‘<component>\_struct.elm’ files. These files set some inertial and stiffness constants that are used within <component\_beamel.elm> files. In these latter files, the actual 6x6 K matrices and lumped inertial properties are set for each beam of each component.

If you are changing nodes, you will need to change beam definitions too within these kinds of files . The beams are 3-node beams, so the y connect 3 nodes. For now, the stiffness (K matrix) is assumed uniform for all components. The end-user is encouraged to determine the appropriate 6x6 K matrices at the 2 Gaussian points of each beam. Inertial properties are given at the three nodes as lumped masses + CM location + 3x3 Inertia tensor, and this is where the user must define proper values as well.

Next in the .mbd, ‘element’ block are forces and couples or the user defined element ‘KiteADelem’. The first two are to test the model in MBdyn standalone with forces or moments associated to certain nodes, for example at the MIP node. The latter is to interface with KiteFAST, and it needs node information as shown. SO if the numbers of nodes are changed, you will need to change this as well. Also input and output file names that are needed by KiteFast are passed through this call.

Remember number of nodes, elements, etc. are to match in the instantiations and in the usage at all stages of the input file.

The main input file is called KiteMain.mbd. It sets general properties and calls a number of files, including Kite.set, which is used as a repository for other parameters.

The main input file constains blocks delimited by keyrods ‘begin’ and ‘end’; Every statement is terminated with a semicolon.