**Date**: 29/Jun/2020

**Author**: Nikhil Madduri (nikhil.madduri@gmail.com)

**../5\_replacing Sfunction with own rk4/**

**3\_kin\_dyn\_working\_v1/**

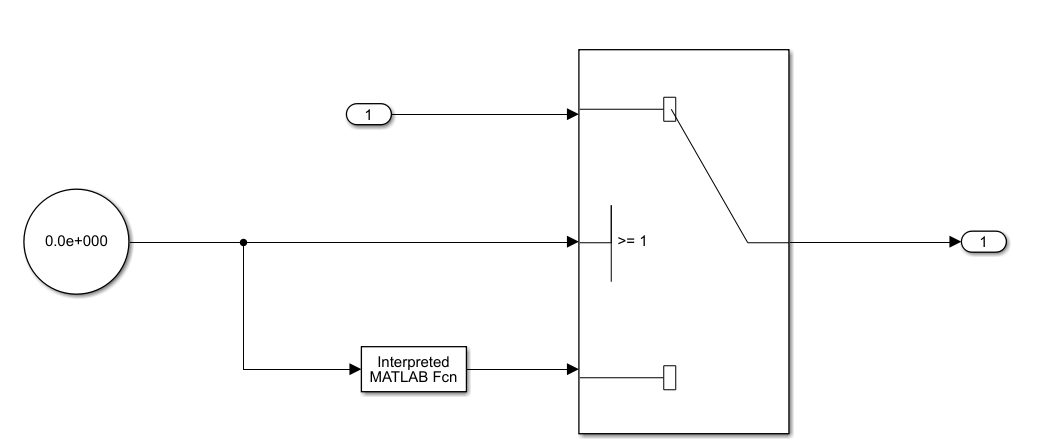
* Replaced vtol\_dynamics that used S-function with own vtol\_dynamics that uses rk4.m and sixDOF.m to integrate the 6-DOF equations.
* Since it was the first trial passing of vtol structure is a bit crude which I have figured out in the next cases.
* **Successful.**

**3\_kin\_dyn\_rk4\_refined\_v2/**

* Refined the vtol\_dynamics.m, rk4.m and sixDOF.m by removing unnecessary lines.
* But passing of vtol variable is still crude – defined all the numbers within sixDOF.m file rather that reading vtol from other file as a structure.
* **Successful.**

**3\_kin\_dyn\_v3/**

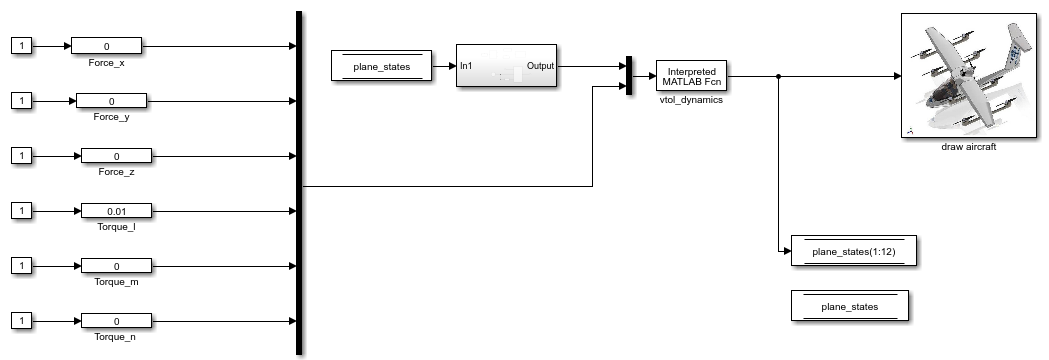
* I’m now able to read vtol as a structure from vtol\_parameters.m without having to hard code the values within sixDOF.m. This was made possible by replacing the vtol\_dynamics matlab function block with interpreted matlab function block.
* Secondly even the initialization block has been made more succinct as follows –



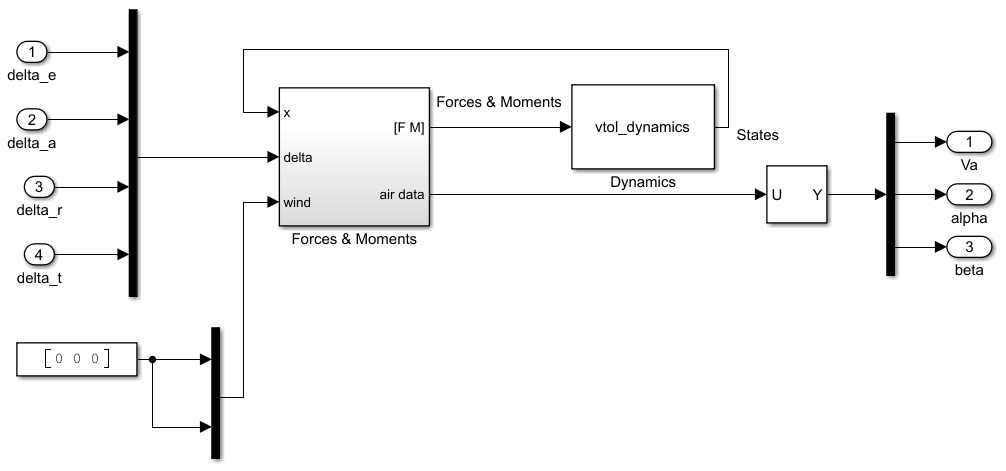
* **Successful.**

**3\_kin\_dyn/**

* Testing folder – unsuccessful.
* When I tried to use the logic developed in “**3\_kin\_dyn\_v3/**” by replacing the vtol\_dynamics in vtolsim\_trim .slx in the folder \e\_version\5\_ldm\testcases\3\_replacing\_S\_function\_with\_own\_rk4, trimming is not successful. Program throws errors.
* Hence I was trying to figure out what was causing the issue in the model below (“**3\_kin\_dyn\_v3/**” )—



I thought may be the states are being stored in plane\_states memory block whereas in the old vtol\_dynamics model I used to loop like this for trimming —



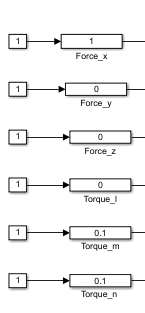
* So in “**3\_kin\_dyn/**” I tried such looping but errors are thrown.
* **Unsuccessful.**

**3\_kin\_dyn\_v3\_compare\_C/**

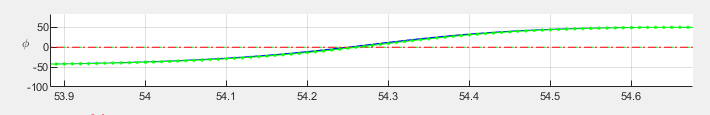
**CASE01**

In C: struct force\_n\_moments **fm\_in** = {1,0,0, 0,0.1,0.1, 0,0,0, 0,0,0};

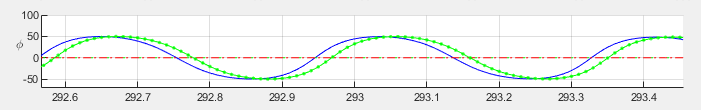
In M:



* Conversion to C code is carried out
* Compares well for small ‘t’. But as ***t*** increases to say 300 s, difference between C and m plots slightly observable but still very small. May be because in C I have used floats whereas M uses doubles.
* For example look at **phi** plot for t = near **54 sec**.



But for larger t, say t = about **293 sec**, the difference is noticeable –

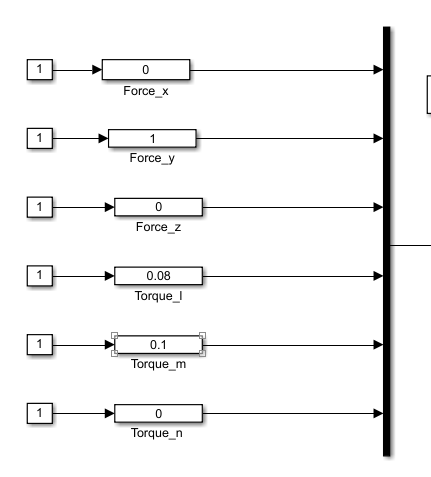
****

But understand that this is happening at a very large zoom levels. Hence the difference is pretty small.

**CASE02**

In C: struct force\_n\_moments **fm\_in** = {0,1,0, 0.08,0.1,0, 0,0,0, 0,0,0};

In M:



* Good match.
* Note that Va, alpha, beta are not calculated in this testing. Hence ignore the deviations in those plots. We will focus on them in 4\_fm folder.