Connecting Simulink to OpenFAST

August 5, 2018

Nikhar Abbas

The current Cert\_Test files referenced in the Simulink examples (i.e. Run\_OpenLoop.m) no longer exist. The Simulink examples and respective Matlab files to run them need to be updated.

I have an active openfast repo with these changes pushed at:

https://github.com/nikhar-abbas/openfast/tree/run\_simulink

Necessary flags in cmake

BUILD\_OPENFAST\_SIMULINK\_API = on

BUILD\_SHARED\_LIBS = on

\*\* FOR NOW – need to change CMAKE\_INSTALL\_PREFIX = /usr/local/

To install/run:

cd build

ccmake ..

(make sure necessary flags are on, and install directory is in /usr/local

Type “c” “e” “g” “e” (configure and generate cmake directives)

make install

run create\_FAST\_SFunc.m

Changes to *glue-codes/simulink/src/create\_FAST\_SFunc.m.*

\*The directories need to be changed here if CMAKE\_INSTALL\_PREFIX is not /usr/local/

\*An added case is necessary to create s-function on linux machines

* Added Case:

case 'maci64'

mex -L/usr/local/lib -lopenfastlib ...

-I/usr/local/include -outdir . COMPFLAGS='$COMPFLAGS -MT' FAST\_SFunc.c

% this builds the SFunction in the current directory (openfast/glue-codes/simulink/src), mex function needs to be in the MATLAB working directory at time of simulation

Changes to *glue-codes/simulink/src/FAST\_SFunc.c*

@@ -28,6 +28,8 @@

#include "mex.h" // for mexPutVariable

#include "matrix.h" // for mxCreateDoubleScalar

#include "FAST\_Library.h"

+#include <math.h>

+#define min(a,b) fmin(a,b)

@@ -57,6 +59,12 @@ static void mdlTerminate(SimStruct \*S); // defined here so I can call it from ch

static void getInputs(SimStruct \*S, double \*InputAry);

static void setOutputs(SimStruct \*S, double \*OutputAry);

+// Hard coding single Turbine definition

+static int iTurb = 0; //zero based

+static int nTurbines = 1;

@@ -190,8 +198,8 @@ static void mdlInitializeSizes(SimStruct \*S)

/\* --------------------------------------------- \*/

// strcpy(InputFileName, "../../CertTest/Test01.fst");

-

- FAST\_Sizes(&TMax, InitInputAry, InputFileName, &AbortErrLev, &NumOutputs, &dt, &ErrStat, ErrMsg, ChannelNames);

+ FAST\_AllocateTurbines(&nTurbines, &ErrStat, ErrMsg);

+ FAST\_Sizes(&iTurb, &TMax, InitInputAry, InputFileName, &AbortErrLev, &NumOutputs, &dt, &ErrStat, ErrMsg, ChannelNames);

@@ -332,8 +340,7 @@ static void mdlInitializeSampleTimes(SimStruct \*S)

if (n\_t\_global == -1){ // first time to compute outputs:

// getInputs(S, InputAry);

-

- FAST\_Start(&NumInputs, &NumOutputs, InputAry, OutputAry, &ErrStat, ErrMsg);

+ FAST\_Start(&iTurb, &NumInputs, &NumOutputs, InputAry, OutputAry, &ErrStat, ErrMsg);

@@ -374,8 +381,7 @@ static void mdlOutputs(SimStruct \*S, int\_T tid)

if (n\_t\_global == -1){ // first time to compute outputs:

getInputs(S, InputAry);

-

- FAST\_Start(&NumInputs, &NumOutputs, InputAry, OutputAry, &ErrStat, ErrMsg);

+ FAST\_Start(&iTurb, &NumInputs, &NumOutputs, InputAry, OutputAry, &ErrStat, ErrMsg);

@@ -415,7 +421,7 @@ static void mdlUpdate(SimStruct \*S, int\_T tid)

/\* ==== Call the Fortran routine (args are pass-by-reference) \*/

- FAST\_Update(&NumInputs, &NumOutputs, InputAry, OutputAry, &ErrStat, ErrMsg);

+ FAST\_Update(&iTurb, &NumInputs, &NumOutputs, InputAry, OutputAry, &ErrStat, ErrMsg);

@@ -437,7 +443,7 @@ static void mdlUpdate(SimStruct \*S, int\_T tid)

static void mdlTerminate(SimStruct \*S)

{

if (n\_t\_global > -2){ // just in case we've never initialized, check this time step

- FAST\_End();

+ bool tr = 1; // Yes, stoptheprogram

+ FAST\_End(&iTurb, &tr);

Changes to *modules-local/nwtc-library/src/FAST\_Library.f90*

@@ -93,17 +95,17 @@ subroutine FAST\_Sizes(iTurb, TMax, InitInpAry, InputFileName\_c, AbortErrLev\_c, N

ExternInitData%NumCtrl2SC = 0

ExternInitData%NumSC2Ctrl = 0

ExternInitData%SensorType = NINT(InitInpAry(1))

-

+ ! -- MATLAB Integration --

+ ! Make sure fast farm integration is false

+ ExternInitData%FarmIntegration = .false.

Some changes to *modules-local/nwtc-library/CMakeLists.txt*