Interface between Matlab and HAWC2

By Lars Christian Henriksen, larh@risoe.dtu.dk

Introduction

Is this paper two different ways of using Matlab functionality in HAWC2 is described: The first method uses the TCP/IP protocol where Matlab starts HAWC2 and the two programs communicate with each other in an orderly fashion. The second method compiles the Matlab code as a dll and a wrapper-dll written in C is the interface between the Matlab dll and the normal HAWC2 dll interface.

TCP/IP communication

HAWC2: TCPserver.dll

This dll could potentially be used with other programs than Matlab. E.g. CFD codes, where the aerodynamic forces would be calculated in the CFD code and applied to the structures in HAWC2 and the structural deformations would then be sent back to the CFD code for the next time step.

Procedure tcplink

In every iteration of HAWC2 the *tcplink* procedure does the following in the specified order:

1. Read data from HAWC2 sensors (those within the "**begin/end output**" paragraph)
2. Send read data via TCP/IP to Matlab
3. Receive data via TCP/IP from Matlab
4. Write received data to HAWC2 (which can be used in the "**begin/end actions**" paragraph or by other dll's)

Next is an example on how to add the TCPserver to an htc file used by HAWC2.

```
…
  begin hawc_dll;
    filename  ./tcpip/TCPserver.dll ;
    dll_subroutine tcplink ;
    init_string 1239 ; port number (optional, default is 1239)
    arraysizes  40 40 ;
    begin output;
      general time ;
      mbdy state_rot omega shaft 1 0 shaft only 3 ;  Generator speed
      wind free_wind 1 0.0 0.0 -90.0 ; Wind speed measurement at specified point
    end output;
;
```

```
  begin actions;
  end actions;
end hawc_dll ;
…
```

If another input/output array size is needed, the TCPserver should be recompiled in the Delphi IDE. The source code is found in the tcpip directory in the hawc2 directory.

If the default port number 1239 has to be changed, the **init_string** command should be added to the htc file with an alternative port number.

Procedure tcplink_delay

When time step iterations are finished and time has changed *tcplink_delay* procedure does the following in the specified order:

1. If time has changed send inputdata_delay and receive outputdata_delay via TCP/IP
2. Read data from HAWC2 and copy to inputdata_delay
3. Write data to HAWC2 from outputdata_delay

Matlab

Relevant files: Main.m, jtcp.m

Two methods are possible in Matlab: The first is in Simulink and the second is in a Matlab script. They both use the **jtcp** file to communicate via the tcp/ip protocol. This m-file is developed Kevin Bartlett and can be found on the internet. To improve speed a java class by Rodney Thomson is included in the **data_reader** directory in the **matlab** directory. See **jtcp.m** for further details regarding the **data_reader** class.

To run either method open Matlab and execute the **Main.m** file, where the string variable **SimulationType** should be either 'script' or **'simulink'**.

Initialization

Hawc2 is started, possibly from within Matlab. The TCPserver.dll awaits connection from client, in this case the Matlab command **jtcp.m**.

If Hawc2 and Matlab runs on the same computer the command to initialize the connection from within Matlab is:

```
TCPportnumber = 1239;
JTCPOBJ = jtcp('REQUEST','localhost',TCPportnumber);
```

If HAWC2 and Matlab are to be run on different computers, the name of the HAWC2 computer needs to be known. In Windows XP, this can be achieved using the **ipconfig /all** command in the command prompt. This

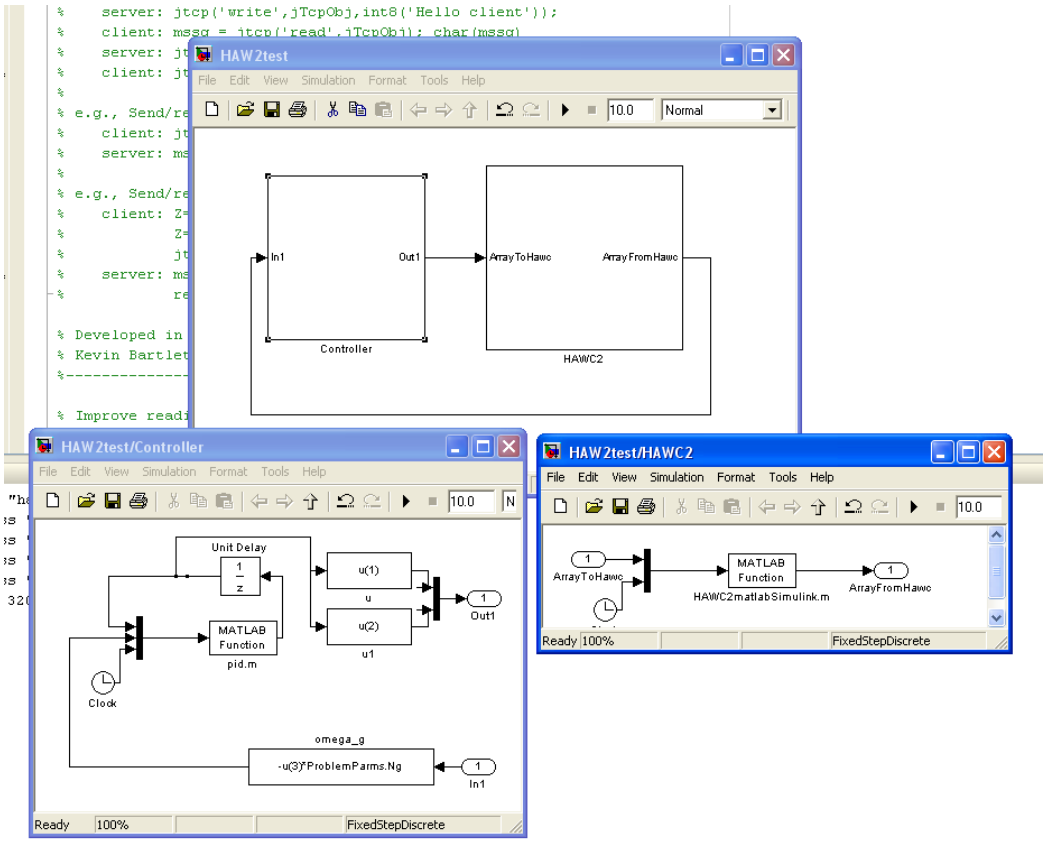
will reveal the name of the HAWC2 computer, e.g. PC-XXXXX, and the name should be put into the **jctp** call in the following way:

```
JTCPOBJ = jtcp('REQUEST','PC-XXXXX',TCPportnumber);
```

Simulink

Relevant files: HAW2test.mdl, HAWC2matlabSimulink.m

In Simulink the order of execution is determined by the solver and the problem of calling an implicit solver like HAWC2 from within an explicit solver has to be handled. At the moment this is handled by letting the HAW2 block in the Simulink model calling HAWC2 until a new time has been reached in HAWC2.

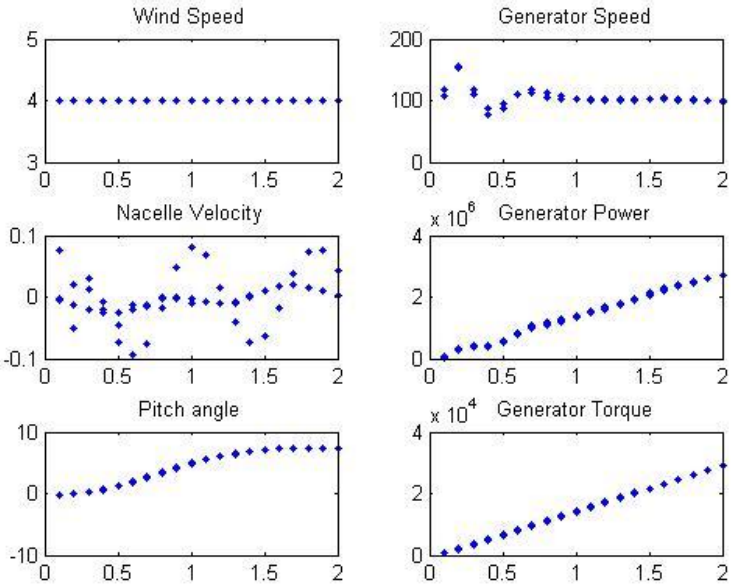


Script

Relevant files: HAWC2matlabScript.m

The option to interact with HAWC2 via a Matlab script gives greater freedom to determine the order of execution for Matlab and HAWC2. Furthermore the Matlab functions can be called more than once per time step as HAWC2 iterates towards the solution of that particular time step.

The shown figure displays more than one data point per time step indicating several iterations before numerical tolerances where met.



The controller

Relevant files, pid.m, SetupProblemParms.m

The controller is a simple pid controller which is able handle iterations of the implicit solver. This is done by waiting to save relevant time varying variables until the current time has changed.

Compiled Matlab functions used as regular dll in HAWC2

Not included yet…