Ship Impact Model Version 1.0 – Instruction Booklet

1. How to Run

Start by downloading Python 3.4 as is on Dropbox, Anaconda installs Python, a integrated development environment, for editing and running files, Spyder, and the most frequently used packages.

The numpy package is heavily used for maths and the user interface uses tkiner and ttk.

Ensure all the program files are in the same folder.

Open “inputdata.py” and click run from Spyder to run the program.

Click a “preset” ship a the top KCS and KVLCC are represented by VLCC and Container Ship. This will populate the interface with “recommended values”, you can then change some of the characteristics as you wish.

1. Status

This version is completely missing a number of key functions, space has been made to allow for these to be added:

* Weather routing (Strathclyde).
* Fouling and engine degradation (Strathclyde).
* Equipment/weight breakdown (there are some simple estimates at the moment).
* Propeller model (this has been re-written in Python but needs testing – a simple estimate is used) (UCL). This is in work at the moment.
* Operating profile though this it is defined and linked in to the program via the file readinput.py, the next step is running the program for the operational performance evaluation (UCL).
* Technology Interface and Waste Heat Recovery (UCL).

The user interface also needs updating to reflect variables that have been added, the variables not currently in the interface are in main.py.

There is an error with the GZ/righting moment calculator. It appears to be giving the wrong results.

There is also an error when changing between ships and for the hull generation of the oil tanker.

Some missing functionality is still in Matlab version of program.

1. Testing Strathclyde and Newcastle Models

There should be enough functionality here to test the link between the models. UCL has put dummy variables in each function so it will run, from now on each function is the owners responsibility to update, these are:

* Engine Model (Newcastle)
* Added Resistance Model (Strathclyde)
* Wind Model (Strathclyde)

1. How to add items to the user interface

Instances of variables have to be declared in several places in the user interface, inputdata.py. Therefore it is recommended that this is kept to a minimum where possible. There are two variables, so in this example a variable endurance and endurance\_entry is used.

Take the following example:

data.endurance = [35] <- this is used to define initial value

data.endurance[0] = 35 <-each “preset button” for each ship also contains a value, this may be the same or different for different ships and ship types.

data.endurance[0]=0 <- this is used in the function “def clear\_all\_fields(\*args)”

The fuction def set\_values\_in\_entry\_widgets(\*args) is used to send the values specified to the user interface:

data.endurance\_entry.input.set(data.endurance[0])

The function def add\_to\_queue(\*args) is used to :

if run == 0:

rundata.endurance[run] = endurance\_entry.input.get()

else:

rundata.endurance.append(endurance\_entry.input.get())

Finally there are functions to refer to the type of object in the user interface, so in this case we have a label with the “endurance:” next to it, this is a real number, with “days” after it (to represent the units).

endurance\_entry = labelentry(setupframeright, "endurance: ", DoubleVar(),

data.endurance[run], "days ")

1. Reading comments and testing

HERE IN WORK is used to denote incomplete things. In order to test variables you can use the print() command, to print() them to the output window, similar to disp() in Matlab.

1. Design Process

The main.py contains the main program that calls functions for different calculations.

Currently the design process is carried out in the following order:

Initial Displacement Estimate

↓

Hull Generator

↓

KG Estimate (based on payload volume/weight fraction)

↓

Still water resistance model

↓

Weather Routing

↓

Added Resistance

↓

Propeller Efficiency Estimate

↓

Heel Angle Estimate

↓

Wind Assist

↓

Technology Interface

↓

Marine Systems and Engine

↓

Waste Heat Recovery

↓

Shaft Generator

↓

Marine Systems and Engine

↓

2nd Loop

Equipment and Structure

↓

Hull Generation

↓

Still Water Resistance Model