IOR Analysis Documentation

Release 0.1.1

John C Harrington

Apr 27, 2018

CONTENTS

This module provides tools for the analysis of IOR era yachts, using VPP files for each yacht and values from it's rating certificate.

It allows changes to be made to the yacht rigging, which can then have it's new rating calculated and it's performance tested in the VPP.

class `IORAnalysis.`**`IOR`**(*s, cert={}, ballastChange=[0, 0]*)

Calculates IOR Rating for Defined Yacht. Default values are for Indulgence.

Calculates IOR rating from certificate values. Provides access to all calculated parameters from the IOR certificate. The class performs all calculations required to achieve a correct SC and CGF for a sloop rig such as Indulgence. It does not contain any calculations regarding a mizzen or other unusual sails, or any calculations based on hull measurements.

Parameters

- **`cert`** (`dict`) – Dictionary containing certificate values for the yacht.
- **`ballastChange`** (`list`) – List containing [Ballast Amount, Distance Moved]. Direction follows standard ship conventions, with upwards being positive.

`Certificate Values`

`floats` – All certificate values are available as attributes

`Rating`

`float` – Rating in it's official form, rounded to 1 d.p.

`ballastChange`

`list` – List containing difference from basis vessel ballast position in the format [Ballast Amount, Distance Moved]. Direction follows standard ship conventions, with upwards being positive.

`actualCGF`

`float` – Calculated CGF value. Differs from certificate CGF only if less than the minimum value of 0.9860.

`Calc`(*s*)

Calculates final rated length.

Runs calculations for CGF and SC, then calculates MR, R, Rating and TCF

`ReqRMChange`(*s*, *reqRating: float*) → float

Calculates the RM change required to acheive a certain rating.

Parameters **`reqRating`** (`float`) – The desired rating for which RM change should be calculated, to 1 d.p.

Returns The required change in righting moment to achieve reqRating, given in lb feet.

Return type `float`

`updateCert`(*s*, *changes: list*)

Updates certificate values stored in attributes

Parameters **`changes`** (`list`) –

- **`dict`** Dictionary containing changes to certificate values for the yacht.
- **`list`** List containing [Ballast Amount, Distance Moved]. Direction follows standard ship conventions, with upwards being positive.

class `IORAnalysis.`**`BaseYacht`**(*s*, *cert: dict*, *VPPFile: str*)

Define the basic yachts from which changes are made

`cert`

dict – Dictionary containing certificate values for the base yacht.

VPPFile
str – String containing path to VPP file for base yacht.

class `IORAnalysis.`**`Yacht`**(*s, BaseYacht: IORAnalysis.BaseYacht.BaseYacht, changes: list = [{}, [0, 0]], h5ID: str = ‘’*)
Generates a yacht ID, runs IOR calculations and allows retrieval of speed data if it has been calculated.

BaseYacht
`IORAnalysis.BaseYacht.BaseYacht` – The basis vessel to be modified, provides VPP file and initial IOR certificate values.

IOR
`IORAnalysis.IOR.IOR` – Instance of the IOR() class. Used to access all rating data.

ID
str – Unique identifier string for this yacht, generated from *changes*.

changes
`list` –

- **`dict`** Dictionary containing changes to certificate values for the yacht.
- **`list`** List containing [Ballast Amount, Distance Moved]. Direction follows standard ship conventions, with upwards being positive.

If blank, the yacht is a copy of the basis vessel, but can then be used to access VPP data if it exists.

getSpeed(*s*, *ws: int*, *wa: int*) → int
Returns the time for 1nm for this yacht at the given wind speed and angle.

getSpeedDF(*s*) → pandas.core.frame.DataFrame
Returns the times for 1nm for this yacht.

h5ID
`str` – The identifier for the specific h5 file the VPP data for this yacht is stored in. Can be used to separate sets of data from the same basis vessel into different files, for instance one for each case study.

class `IORAnalysis.`**`WinDes`**
Wraps WinDes4 class, enforce usage within ‘with’

This ensures queue will always be either saved to file after use without continuous file changes.

Returns **VPP** – The VPP object.

Return type *`WinDes4()`*

Example

```
>>> with WinDes() as VPP:
        queue(yacht)
```

class **`WinDes4`**(*s*, *yachtName='Indulgence'*)
Run WinDesign4 through PyAutoIt.

Allows queueing of yachts for analysis when WinDesign is not available. Yachts are stored in HDF5 database with name format: “SailVar1-SailVal1.SailVarN-SailValN_BallastWeight-BallastDistance”

queue(*s*, *Yacht: IORAnalysis.Yacht.Yacht*)
Adds a new yacht to the current queue, if it isn’t already there
Parameters **Yacht** (*`Yacht()`*) –

runQueue(*s*)

Runs VPP for all yachts currently queued

saveQueue(*s*)

Writes the current queue to a file

PYTHON MODULE INDEX