**RastriginDigitalTwin.py**

\_\_init\_\_

super().\_\_init\_\_(rastrigin, x0=utils.get\_random\_x0(dim,-5.12, 5.12), bounds = bnds)

[roy] Super returns an object of the parent class (AlgoDigitalTwin). But it is never assigned to anything. There is no “instantiation” here. I know it initializes RastriginDigitalTwin – but where is the instance ?

======================================================================

\_\_main\_\_

cfg.main(RastriginDigitalTwin(), args)

[roy] Here the “class” is being passed to the main routine in cfg.

Which is odd to me – why wouldn’t you pass an instance of the class?

I presume that THIS is when the object is instantiated in cfg.main ? Where it

becomes “algo\_wrapper” ?

======================================================================

RastriginDigitalTwin.py RUN ok when executed from the Optimus6-7-20 directory…

it finds the OptimusPrime module/library (?) just fine. Which is it btw…a library??

But when I try to execute it from the examples directory – it cannot find it. Which makes sense to me really – but on my work computer – it finds it. Why can it find it there – and not here?

=======================================================================

within \_optimus.py

solve()

references: self.solver\_dict[self.solver\_name].solve

from the dictionary in the same file – the entry is the class “BasinhoppingSolver()”

which is then executed.

Similar question to above – how is the “class” an “instance” ?

==========================================================================