SLSQP :

* If the entire design space is in the feasible region, it converges to an optima ( most probably local), could have gone further down by decreasing thickness factor.
* Pitch angle does not change at all
* Trying to run FAST every time for small changes also did not make any difference. FAST deflection does not show any change with such small dx steps.

Started new SLSQP with a different starting point :

* Running with 2 different Thickness factors did not make too much difference. Mainly, tip deflection is a driver and not stress at 75 % span. Also, having one thickness factor ensures a smooth ply drops as in the original layout. Also, just changing thickness factor at 0.75 region didn’t make too much of a difference.
* Again, the entire design range seems to be feasible. Chord and T.F. limits lowered, T.S.R limits were lowered. A high TSR of 8.3 ~ 8.5 resulted in really high Vy values and for low wind speeds, gave an error of no valid phi found.
* Also, pitch angle does not change much. (try with scaler)
* Again, run the analysis and see if it can be manually made better.

Run Static optimization with lower bounds and notice the differences.

* Is it the optimizer causing the issue or FAST. Can chord and thickness factor values go further down ?
* Compare Optima

SLSQP for Static :

* For the first time, it went smooth but Cp values were high.
* After changing Cp values, when re-running, optimizer got stuck at a local optima the second time. Could be lowered manually.
* Twist values do not change. Design space explored is too limited.

**Genetic Algorithm :**

* Genetic explores design space really well. Does not converge within 10 generations for a small population size of 18. Trying to run static model with a pop size of 30 and 15 generations.
* It tries unrealistic designs which might lead to tip deflection as high as 71 m.
* Need to find a way to avoid running FAST when unrealistic conditions are given as an input. ( Whenever mass of the blade goes below 11 or 11.5 tonnes, tip deflection and stress values are unrealistic )

Driver debug print for iter coord: rank0:SimpleGA|4

---------------------------------------------------

Design Vars

{'indep2.chord\_coefficients': array([1.08870968, 0.78225806, 1.16935484]),

'indep2.design\_tsr': array([0.92903226]),

'indep2.pitch': array([-1.40322581]),

'indep2.tau\_75': array([0.6]),

'indep2.tau\_root': array([0.6]),

'indep2.twist\_coefficients': array([1.17741935, 0.75806452, 0.75806452])}

[3.85620968 2.35459677 2.70471774] [15.66909677 6.82258065 2.36895161]

Blade mass: 8540.629636153304

Cp: [0.35330885]

Tip deflection : 71.8940650322215

Max Stress Spar : 2684.9180296816985

Support costs: 428779179.6780171

[10.31537593]

Objectives

{'obj.f': array([1.18038402])}

* Trying different combinations of Genetic on Static model to check it’s performance.
* For pop size of 30 and max generations 15, it performs 480 function evaluations. However, in the last generation, there is a lot of difference in values and it does not converge to one particular range.
* For Genetic, a smart selection of bounds is needed so as to restrict the design space to realistic values.
* The global optimum of genetic, when used in Static, does not change much. Trying out values near the optimum of genetic and running static optimization.

Make a proper table that compares all scenarios and draw conclusions.

* Why is the optimizer getting stuck ?
* Is twist not changing because of the optimizer or because its already optimum
* Genetic, is computational time the only issue
* Running Dynamic SLSQP with a point near GA optimum resulted in the optimizer getting stuck in an infeasible region (8.8 m tip deflection) for about 6 hours.
* Run Dynamic SLSQP with a different starting point

Bounds revised for Dynamic as well :

Chord : 0.75 – 1.25

Twist : 0.7 – 1.3

Tf : 0.7 – 1.3

FAST did not give any error for blades as slender as 9 tonnes in mass and a rated wind speed of 12.1 m/s. Deflection : 13.5 m . Did give an error once.

Better to restrict TF and Chord lower bounds to these limits so as to avoid really slender blades with very large deflections to prevent -Vx error.

For GA, implement penalties for constraints :

* Keep exponent to 1
* A coeff of 0.5 would lead to an increase in LCOE by 0.25 if violation is 0.5m. However, if violation is 1m (Tip defl : 8m), LCOE would increase by 0.5. GA would easily discard that point.