To Do List:

In progress

Modify “SimulationScheduler.m” to look more like FlexDAR problem.

RP (Resource Period) = 40ms

Search Horizon: 36 ms ( 4 CPIs each 9 ms)

Search Above Horizon: 18 ms

Three tiers of tracks. Each takes 18ms.

**Star-Log March 18, 2020**

Added 9-row search function that captures Horizon and above horizon search. Three types of sequence scheduling: EST, BB, and NN. Each one output a sequence of tasks. However the sequencers are dumb to the fact that tasks cannot cross a resource period boundary. Next these task sequences are given to a FlexDAR scheduler that assigns the task to an available timeline. This function knows about the boundary problem and bumps tasks to the right if they cross a resource period boundary.

**Comments**: In future make sequence schedulers aware of boundary issue? Is this possible? So far the EST, BB, and NN all are working with the changes. The NN was originally trained with all tasks having the same dwell of 5ms and all same start times of 0s and varying slopes and it is still performing well. That’s good.

Now we are scheduling 8 resource periods in advance. 8\*40ms = 320ms. The EST and NN only take like <1ms for EST and < 10 ms for NN. This gives us an additional 30 ms to use during the RP to try better task sequences 🡪 do Monte Carlo rollouts.

Next steps still want to process more than 8 tasks at once. Maybe start working on distributed stuff.

**March 19, 2020**

1. Added FixedPriority flag to ensure each algorithm gets the same inputs at each time step. It just cycles through 8 tasks at a time then goes to the next 8. It's a fixed schedule.
2. Also added some diagnostic tools queueRecord, TaskExecution, TaskSequence to capture what each algorithms inputs and outputs are at every iteration
3. Investigated why EST was performing better than BB. Turns out that the enforcement of no tasks crossing a resource period results in some instances of the optimal BB output becoming suboptimal. Solution is to incorporate these constraints into the BB algorithm, so it’s aware of the constraint. May result in slower execution, TBD.