Addendum for Pre Built Trajectories Capability

Claire Walton, cwalton@soe.ucsc.edu

Many problems SPOC deals with contain relevant conditionally deterministic trajectories of the form $y(t, \omega)$ where ω is a realized value of the probabilistic parameter and $y(t, \omega)$ is a known function, as a part of the cost function calculations. These trajectories can be incorporated into a cost function by pre-building them at all the relevant time and parameter node values. A helper function, Extract_Nodes is provided to aid this:

[time_nodes, parameter_values] = Extract_Nodes(Discretization, Methods)

This function returns two variables: time_nodes and parameter_values. The variable time_nodes is a vector containing all the values of the time nodes as determined by method choice and discretization level. To evaluate parameter nodes over whichever scheme is chosen for approximating the cost integral, multiple combinations of parameter values are necessary (i.e. a meshgrid is created). The matrix parameter_values contains all the combinations of parameter values which will be called in the course of evaluating the cost function. These combinations and the amount of combinations are determined by method choice and discretization level.

The number of rows in parameter_values is equal to the amount of necessary combinations and the number of columns is equal to the dimension of parameter space. Each row contains a set of parameter values for which a trajectory value must be computed at each time node. To create the necessary trajectory values, one must index through each row of parameter_values and compute the trajectory for these parameter values. The index of the row of parameter values used is referred to throughout other routine files as the mesh_index.

These pre built trajectories are created in the Problem_Definitions file and stored in whatever form is easiest for the problem as a part of the global CONSTANTS struct. The only constraint is that trajectories must be stored in a format which makes referencing them by their mesh_index possible. The routines which may involve these trajectories all contain as part of their function call the variable mesh_index. When utilized in these routines, only the trajectory associated with the index value of mesh_index is used.

An example is provided in Multi_Agent_Search_user_gradient.