Notes on Simulink

Issue: Using functions that have already been made.

* Need to use the matlab function block and call the previously made function. If that function requires the use of function handles then special care must be taken.
  + Using coder.extrinsic allows a matlab code to be called from Simulink but doesn’t get compiled like Simulink code does. This allows the use of function handles. When using this, the variables must be preallocated before calling the matlab function so the data type will be recognized by Simulink.

Issue: Passing Structs into function calls

* If a struct exists in the matlab workspace, Simulink requires special care. When a function needs the struct
  + Create the function
  + Go into Model Explorer -> ‘function name’ -> Add Data
  + The new variable added needs to have the same name as the struct in the workspace, Scope needs to be changed to parameter, and Tuneable needs to be deselected

Issue: Passing function handles to matlab equations

* Function handles are an invalid data type for Simulink. To pass these to a matlab function, they must first be converted into strings func2str()
* Next, inside the matlab function block, the extrinsic function that is called is not directly the matlab function you want, instead it is a function that converts all of the strings back into function handles str2func(). After that is accomplished, the matlab function can be called from the extrinsic function.

Issue: Change solver tolerance

* Simulation -> Model Settings -> Solver -> solver details

Issue: Saving variables to be output from the solution

* To Workspace block
  + Can change the sampling time per block
  + Can use a MUX to combine the signals to one Workspace block
  + All To Workspace blocks save as the variable name ‘out’
  + Can save data as time series, struct, or array
    - Arrays may be the cleanest to use in terms of the dot operator
    - Each variable is saved to a column array

Issue: fsolve objects are not a valid data type

* Use rmfield to remove that object from the SIM struct

Issue: Setting initial conditions

* I had to make a block that determined the initial conditions
* If I had a previous simulation that I outputted the variables from, I could set that state to be my initial state. (Don’t think this would work well for my model)

Issue: Matrix multiplication

* When multiplying two matricies, the orientation (row vs column) matters. If using a constant block to define a matrix, the box that says ‘interprete as 1D’ needs to be deselected. Then you will have the proper orientation