

## Component Abstraction

This document described the interfacing of Simulink blocks with component Objects by the use of Simulink S-functions.

The block library can be found as the file BlockLib.slx. Double clicking on a block in the library shows the block's mask

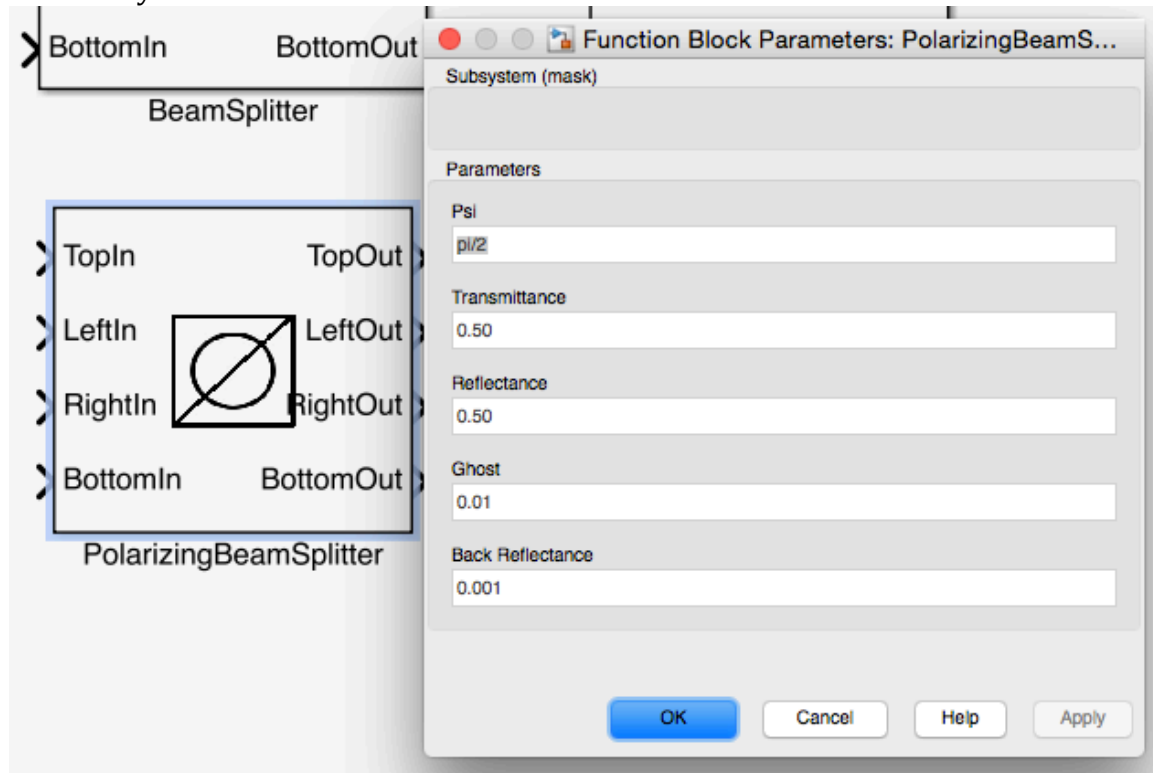


Figure 1: Mask of the PolarizingBeamSplitter block

Component specific parameters can be tweaked using this mask. Looking under the mask (Cmd+U on Mac),

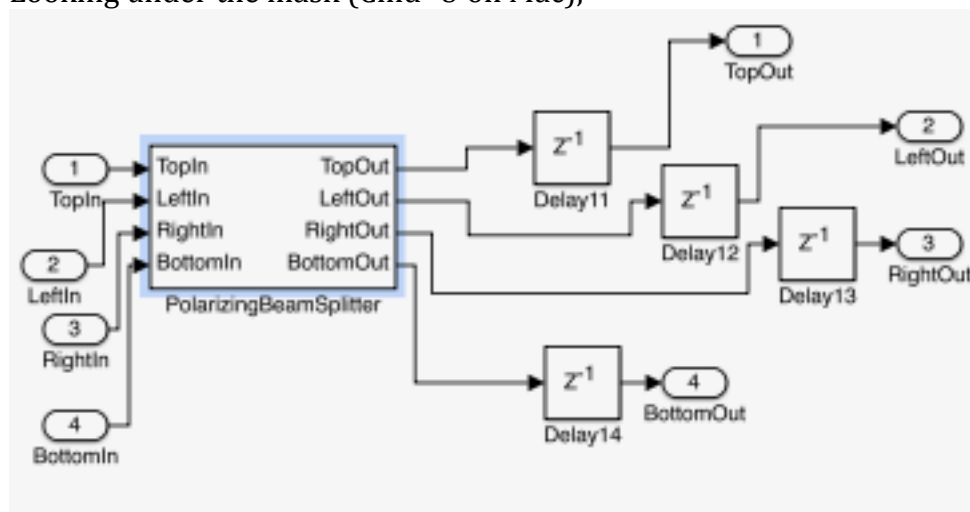


Figure 2: Masked subsystem of PBS block

There is another block called PolarizingBeamSplitter that is connected to a Unit Delay element before outputs are propagated. The purposed of these delay elements are to prevent blocks wired in loops from forming implicit arithmetic equations that are unsolvable by Simulink's engine. Even though Simulink is forced to have the following solver conditions:

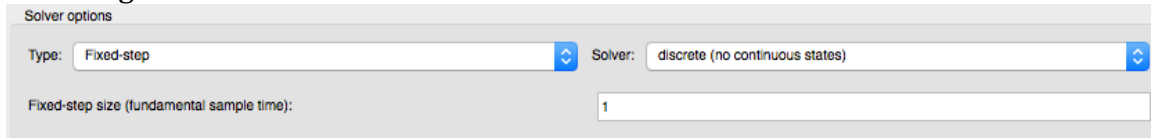


Figure 3: Simulink Model Configurations

And doesn't need to do any internal solving other than moving the output of one block to the input of the next, it still needs delay blocks to ensure that it knows that. Double clicking on the internal PolarizingBeamSplitter shows these mask options:

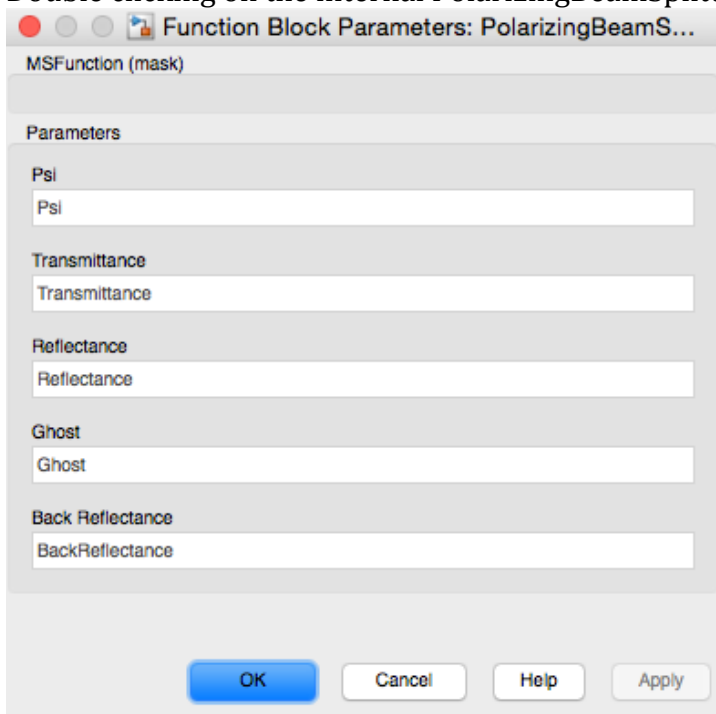


Figure 4: Internal PBS block mask

Which feed in parameters inherited from the subsystem mask shown in the earlier figure. The mask options (“Ctrl+M” on Mac while the block is selected) package these parameters into the “Params” struct as shown:

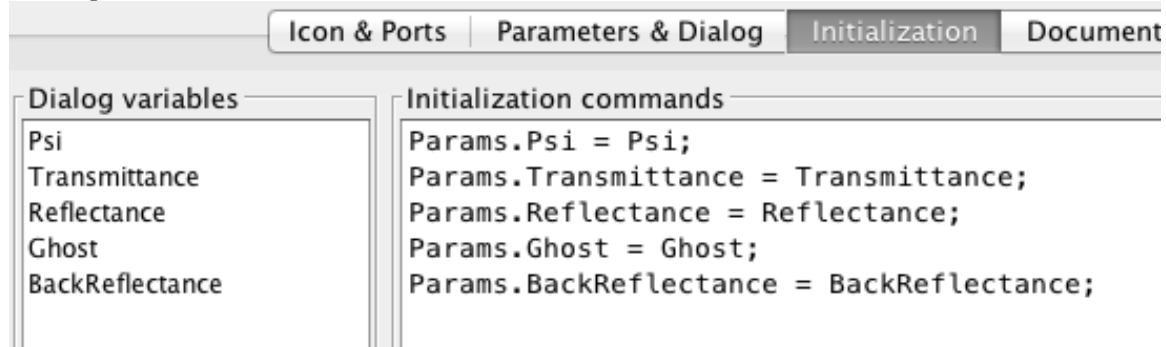


Figure 5: Internal PBS block mask options

This internal block is an S-function, and passes the arguments “PolarizingBeamSplitter”, which is the component descriptor, as well as the struct “Params” to the generic S-function called “fourwaycomponent\_s.m” in the Components folder. The S-function interface can be seen by looking under the mask of the internal PBS block (“Cmd+U” on Mac”):

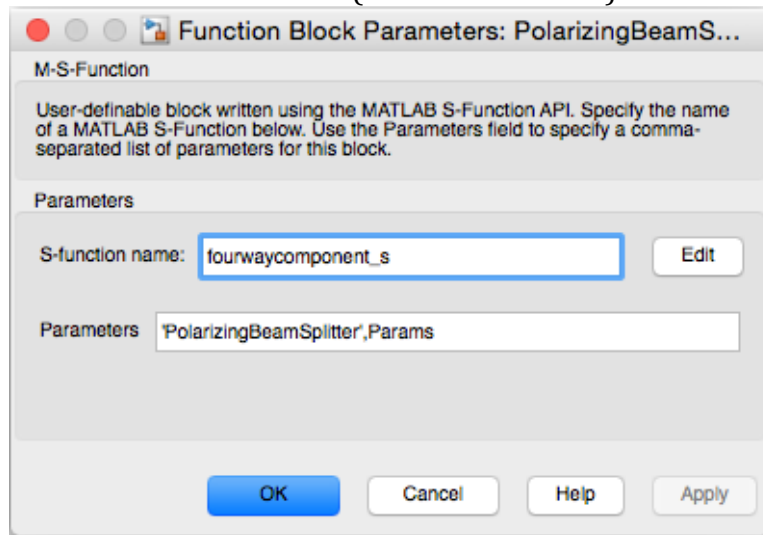


Figure 6: S-function interface of an internal PBS block

Looking through the S-function “fourwaycomponent\_s.m” commented code should describe what happens to this information when it is passed in. The information is passed when the model initializes (before the run starts) and is used to instantiate a PolarizingBeamSplitter object, the code for which is also in the Components folder.

Every component works in this manner, except only Beamsplitters use the “fourwaycomponent\_s.m” S-function. Others use the “component\_s.m” S-function, which is different in no more than the former has four inputs and outputs and the latter has two, and has signals routed slightly differently, which is highlighted in the code.