BLOM Quick Start Guide

MPC Lab
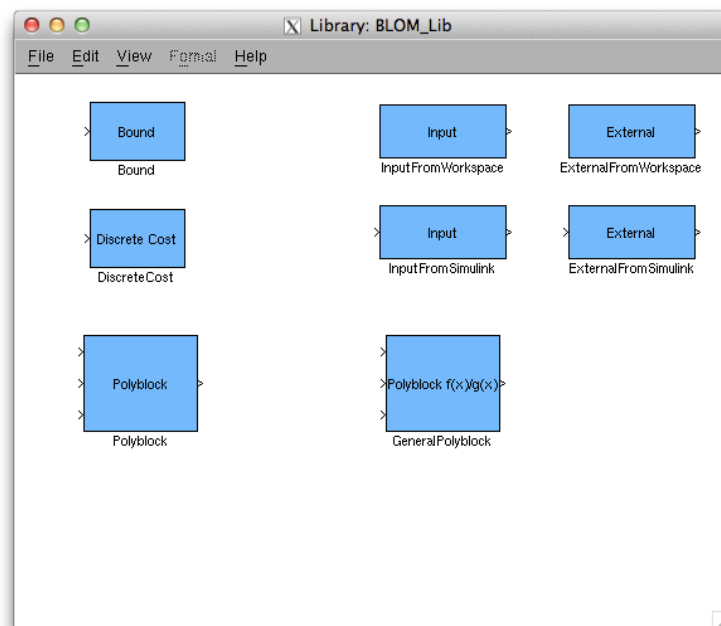
July 1, 2013

What is BLOM?

- Provides a graphical interface to allow users to create optimization problems using Simulink blocks.

- Exports mathematical model to solvers (eg. ipopt)

- Great for optimization problems with "dynamics" that evolve over time

$$\min_x f(x)$$

$$g(x) \leq 0$$

$$h(x) = 0$$

BLOM Library

Externals: Labels External Variables that can be changed via script or command line for different calls of the solver

Inputs: Labels Input Variables to be optimized by solver

Bounds: Sets upper/lower bounds on a variable

Cost: Cost variable to be minimized.

Polyblocks: BLOM's convenient way to create nonlinear functions

Polyblocks

BLOM's internal representation of blocks.

$$y_1 = 2x_1^2 x_2, \quad y_2 = 3x_1 + x_2^4$$

$$P = \begin{bmatrix} 2 & 1 \\ 1 & 0 \\ 0 & 4 \end{bmatrix} \quad K = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 1 \end{bmatrix}$$

$$y_1 = 2x_1 + 3sin(x_2), \quad y_2 = 3x_1^2 e^{x_3} + 0.2tan(x_2)x_4^3$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{BLOM\_FunctionCode('sin')} & 0 & 0 \\ 2 & 0 & \text{BLOM\_FunctionCode('exp')} & 0 \\ 0 & \text{BLOM\_FunctionCode('tan')} & 0 & 3 \end{bmatrix}$$

Setting Up BLOM on Your Computer

1. `http://mpclab.net/Trac/wiki/SVNsetup` Here are instructions on how to get SVN and how to get BLOM running

2. in command line, `svn checkout http://www.mpclab.net/BLOM/``desired_directory`

3. Each time you open up BLOM, make sure to get the latest version by typing `svn update` within that folder (or update through TortoiseSVN)

4. On Mac or Linux machines, you may need to compile IPOPT and then run `BLOM_Setup` (Instructions for doing so at `http://mpclab.net/Trac/wiki/CompilingIpopt`)

Creating Model Example

$$\max f(x) = 3x_1 + x_2 - x_3^2 + 2x_3$$
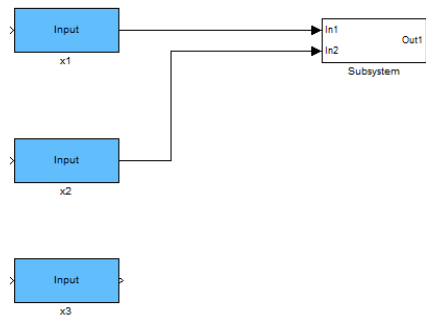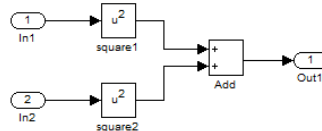
$$x_1^2 + x_2^2 \le 5$$
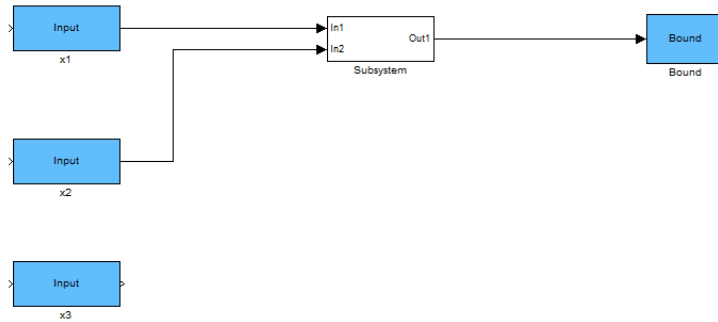
$$x_1 - x_2 \le 1$$

$$x_3 \ge 0$$

Step 1: Place Input and External Blocks for input and external variables

Step 2: For each bound limitation and cost function, drag and drop math blocks to satisfy equations. Use subsystems and/or polyblocks as needed

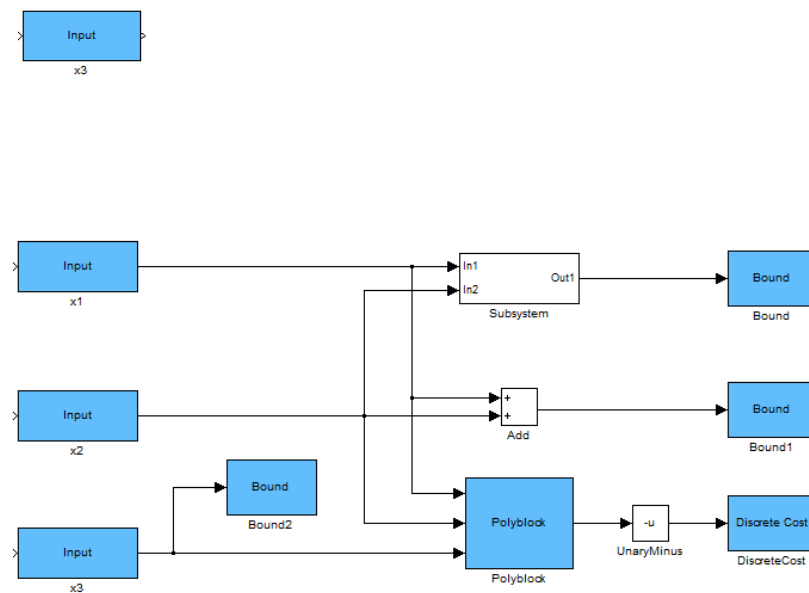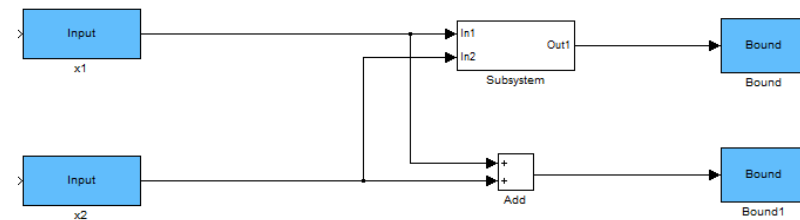**Step 3:** Attach bound and cost blocks and set limits/time relevances

Repeat steps 2 and 3 as necessary

Calling BLOM

1. Always remember to run `BLOM_addpath` to add all the BLOM related files into your path

2. Create model in simulink

3. `BLOM_SetDataLogging(`‘ModelName’`)`

4. `ModelSpec = BLOM_ExtractModel(`‘ModelName’`, `#timesteps`)`

5. `[RunResults ResultsVec] = BLOM_RunModel(ModelSpec)`

6. `[OptGuess ExtVars InitialStates ] = BLOM_SplitResults(ModelSpec,RunResults)`

7. `SolverStruct = BLOM_ExportToSolver(ModelSpec,`‘Solver’`)`

8. `SolverStructData = BLOM_SetProblemData(SolverStruct,ModelSpec,OptGuess, ExtVars, InitialStates)`

9. `SolverResult = BLOM_RunSolver(SolverStructData,ModelSpec)`

Using Externals: Items 7-9 can be run in a loop using outputs from `SolverResult` to populate `OptGuess` and `InitialStates` in subsequent iterations. `OptGuess`, `ExtVars`, and `InitialStates` can all be filled in from the command line (e.g. `ExtVars.x1=5`, and `OptGuess=SolverResult`)

Tips and Tricks

-

Optimizing Your Model

- Use fewer blocks. Outputs of blocks (with the exception of subsystems, from/goto tags, mux/demux) represent variables. Having fewer blocks and therefore fewer variables allows for faster computation.

- Switch to polyblocks. Converting groups of mathematical operations into polyblocks can also reduce the number of variables

- For polyblocks with sparse entries, create matrices using Matlab function sparse. This reduces memory storage.

Currently Supported Simulink Blocks

- Sum, Add, Subtract
- Product Multiply Divide
- Gain
- Unary Minus
- Bias
- Math

    - square
    - sqrt
    - reciprocal
    - exp
    - 10^u
    - log
    - log10
    - magnitude^2 (Reals only)
    - 1/sqrt, rsqrt
    - hypot

- Trigonometry

    - sin
    - cos
    - tan
    - asin
    - acos
    - atan
    - sincos

- Polynomial
- Constant
- Unit Delay
- Subsystem
- From, Goto
- Mux, Demux