# A Brief Guide to Optimization with SIMsystem

Version 2.2
December 2004

Computer-based design optimization is growing rapidly in usage. As examples, it is used to refine the design of engine intake manifolds, and is used to calibrate controllers.

To conduct an optimization, the optimizer program is given a set of variables that can be modified, and a performance criterion, described mathematically in what is called the "objective function." It then uses repeated evaluations of the objective function, over a range of variable values, to determine the best choices for the variables.

A SIMsystem simulation can be a part of an optimization process. SIMsystem users are well positioned to benefit from optimization methods, because developing a simulation is the biggest part of any optimization job. Whenever the optimizer needs an evaluation of the objective function, it calls for another simulation. The suggested approach for performing optimization with SIMsystem differs between UNIX SIMsystem and PC SIMsystem, (as described below).

The following example describes an optimization of a control algorithm for an automobile anti-lock brake system:

- Minimize: Stopping Distance (This is termed the objective function.)

- Subject to constraints upon:
    - Actuator Reversals
    - Vehicle Jerk
    - Controller Robustness

- Minimize by adjusting five parameters in the controller. These are called the design variables.

# A Brief Guide to Optimization Using SIMsystem

## Six-Step Procedure for Optimization with SIMsystem

Step 1: Develop a reliable simulation of the system of interest.

Use any of the modeling environments available with ADvantage.

Step 2: Select the objective function, the constraints, and the design variables.

The objective function can be a combination of goals. In the anti-lock example, the line of code could be written as the next example:

| |
|---|
| Objective function = .8 * (stopping distance) + .2 * (vehicle yaw index). |

Note that some optimization programs try to minimize the objective function, while some try to maximize it. In the following example, the intent is to reduce stopping distance and vehicle yaw index. Therefore, if the optimization program is formatted to maximize the objective function then the objective function should be multiplied by minus one, as shown in below.

| |
|---|
| Objective function = -0.8 * (stopping distance) - 0 .2 * (vehicle yaw index). |

Some optimization programs will allow constraints upon only the design variables. This enables a test of the constraint before a simulation is run. Other optimization programs allow constraints upon anything that can be calculated during the simulation.

It is probably wisest to begin the development of an optimization using just two or three design variables. When that works the number can be increased.

Step 3: Prepare the simulation.

The list below shows the optimization sequence:

1. Load the simulation
2. Optimizer sends design variable values to the simulation
3. Run the simulation
4. Simulation returns to the optimizer a value for the objective function
5. Optimizer sends revised design variable values to simulation
6. Run the simulation
7. Repeat this sequence

# A Brief Guide to Optimization Using SIMsystem

Because the design variables are assigned values by the optimization program, any assignment statements within the simulation program should be outside the main loop. If the design variables are set within the main loop of the simulation, the values that were received from the optimizer will be overwritten. So the relevant part of a C code simulation should look something like this:

```
/* design variable defined above main loop */
Double design1 = 4.0

void rt_main ()
```

The objective function can be calculated at the end of each simulation. For the ABS example, you could write a line of code like this:

```
Objfun = .8 * distx + .2 * yaw
```

<u>Step 4 for PC users, only: Obtain and prepare the Optimizer with PC SIMsystem.</u>

PC users use the Visual Interact COM interface. One option is to obtain an optimizer in Visual Basic or Visual C++, and have it control simulations. Build the simulation in Advantage, exactly as usual. Compile the optimization program, if that is required. Run the optimization program. Whenever it needs an evaluation of the objective function, it must send updated values for the design values to SIMsystem, and then run a simulation and collect results.

Another option is to use Microsoft Excel; many PC users already have and it contains a solver package. Excel's Solver minimizes the function value within a cell, subject to constraints contained in other cells.

Still other cells in the spreadsheet contain the design variable values. As Solver performs its minimization, these design variable values are modified by Solver. The cell to be minimized can call a Visual Basic function. The Visual Basic function can run a SIMsystem simulation, attaching to SIMsystem through the COM interface. You can contact support@adi.com to request an example of such a Visual Basic function.

The standard version of Excel contains optimizers. You may a need more powerful optimizer for calibrating controllers. Frontline Systems (www.frontsys.com) makes both the standard and the more powerful optimizers. The more powerful optimizers provide global, and integer, optimization. Free trial copies of the optimizers are available for download from the Frontline Systems web site.

Next, run the optimization and then go on to Step 5.

Step 4 for UNIX users only: Obtain and prepare the Optimizer with the UNIX SIMsystem.

First, you need the ADI COSIM API product. It allows a simulation to be controlled with code in a C program rather than with COSIM Interact. Also you must obtain an optimizer.

It is not necessary to create an original optimizer program. Many excellent, robust and easy to use optimizer programs are available. Some optimizer programs are included in other packages, such as the IMSI library.

At ADI the "simulated annealing" style optimizer called "Hide & Seek" has been used. It is of a type termed "global optimizer." Global optimizers find the best design no matter how far it is from the engineer's original guess. This is advantageous, but the cost is that more simulations are required. It is available from Professor Robert L. Smith, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109-2117, (734) 936-0715 or (734) 763-2060, or rlsmith@umich.edu

Another well-regarded optimizer is CFSQP, which can be obtained from Judy Guzewich, Office of Technology Liaison, University of Maryland, (301) 405-1340, jg247@umail.umd.edu. It is a local, rather than a global, optimizer.

Any optimizer you choose should be capable of calling a function to obtain objective function values.

Place the COSIM API commands into the optimization program. The API enables the optimization program to be in charge of the SIMsystem simulation.

Typically an optimizer program has a basic structure like this:

| |
|---|
| INITIALIZATION SECTION<br><br>MAIN LOOP {Set design variables, get objective function value, calculate new design variables values}<br><br>TERMINAL SECTION |

Then a typical set of COSIM API commands for optimization would be placed like this:

```
In INITIALIZATION SECTION
COSAPIinit()
COSAPIattach()
COSAPIprogramLoad()



In MAIN LOOP
COSAPIput()
COSAPIgo()
COSAPIwait()
COSAPIget()

In TERMINAL SECTION

COSAPIdetach()
```

A call to COSAPIprogramReset()may be necessary before the put command in the main loop, if hardware i/o is involved.

The arguments for each function call are described in the COSIM API on-line documentation. A working example can be obtained by request to support@adi.com.

An example of a COSIM API program can be found in this directory: $ADI_HOME/examples/rts/cosapi.

During development, you may want to include print statements in this code. Additionally, you could include COSIM API command equivalents for the COSIM Interact commands that are commonly used to monitor a simulation, such as COSAPIdeviceStatus() and COSAPIcaptureAdd().

Next build the simulation in ADvantage, as usual. Compile the optimization program, if that is required. Run the optimization program and go on to step 5.

Step 5. Review results and revise the optimization.

Typically the first optimization delivers an unrealistic solution because an important design constraint was not included. Optimization work imposes a rigorous, mathematical approach to design requirements.

If a number of goals were combined in the objective function, repeating the optimization with different relative weighting can provide insights into goal tradeoffs. This is the focus of the field of Multi-criteria Optimization.

Most optimizers allow user control over how thoroughly the search for a solution is conducted. This may be in the form of a limit to the number of iterations, or in a specification of search tolerances. Repeated optimizations can find the proper settings that will allow rapid optimization to a reasonable solution.

Step 6: Consider using stochastic optimization.

Variability is a fact of life in the real world. Optimization does a better job of designing for real-world usage if variability is included in the simulations.

In stochastic optimization, estimates of parameter variability must be mathematically represented. In the anti-lock brake system example, brake lining output and tire-to-road surface adhesion are variable. A distribution could be estimated for each. Then for a stochastic optimization, every evaluation of the objective function requires not one simulation but many, each with different values for lining output and surface adhesion, selected in a manner that represents the estimated distributions and which preserves an independent relationship between the two parameters.

For example, if lining output is distributed in a Gaussian distribution, values near the mean will be used more frequently in simulations than will the extreme values. Perhaps ten simulations would be conducted for each objective function evaluation. The objective function could be a sum across the set of simulations.

Stochastic optimization will obviously take many times the amount of time required for a standard optimization. Yet before long, conducting an optimization run will be routine, and so unattended batch runs can be a viable approach to managing long optimizations.

# A Brief Guide to Optimization Using SIMsystem

Step 7: Don't give up!

There may be a bit of work up front, getting accustomed to the optimization program and the means for connecting it to the SIMsystem simulation. Yet once the optimization is set-up it is a very useful technology. After set-up, it can probably serve with minimal modification for a very long time. Each time the dynamic system is changed, or a requirement upon the system modified, the optimization of the new system can be almost immediate.

Use ADI resources to help you. Questions can be sent support@adi.com. Also, optimization services can be contracted from ADI to help get an optimization system in place.