Carnegie Mellon University
Research Showcase

Department of Electrical and Computer Engineering

Carnegie Institute of Technology

1-1-1995

An ASCEND based column design tool

Robert S. Huss
Carnegie Mellon University

Carnegie Mellon University.Engineering Design Research Center.

Follow this and additional works at: http://repository.cmu.edu/ece

Recommended Citation
Huss, Robert S. and Carnegie Mellon University.Engineering Design Research Center., "An ASCEND based column design tool" (1995). *Department of Electrical and Computer Engineering.* Paper 199.
http://repository.cmu.edu/ece/199

This Technical Report is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase. It has been accepted for inclusion in Department of Electrical and Computer Engineering by an authorized administrator of Research Showcase. For more information, please contact research-showcase@andrew.cmu.edu.

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

An ASCEND Based Column Design Tool

Robert Huss

EDRC 06-199-95

AN ASCEND BASED COLUMN DESIGN TOOL

Robert S. Huss

Abstract

This report is a manual for using the column design tool available for the ASCEND system. With this application the user can design single distillation columns for specific tasks. It can also design a distillation column for flexible operation where a number of feeds are possible.

This work has been partially supported by the Engineering Design Research Center, a NSF Engineering Research Center.

Use of the Application

To run the Flexible Column Creator, the user must be running ASCEND. From the command line in ASCEND, he can start the application by typing,

source main.tcl

assuming he is in the column design directory, which is located in the examples directory in the models directory on the main ASCEND tree. The file can be sourced remotely with the full path name. This should read in the libraries needed and create the application window, shown in Figure 6.1. First we will describe the general structure of the window.

At the top are the menus, under that are entry boxes for choosing a file and model name. Under that are two list boxes for choosing components. Under that are a set of buttons. And under the buttons is a message box, which will contain a description of whatever button or region the mouse is over.

File Menu

Open Settings will bring up a file select box for the user to choose a settings file to source. We use the suffix .col for settings files, and they contain code for regaining any choices the user has made in the application, like the components selected, the feed and recovery specifications, etc.

Save Settings will bring up a file select box for the user to choose a name for the creation of a settings file. If the user chooses an already existing filename, it will be overwritten. We use the suffix .col for settings files, but it is not mandatory.

Disable Message Box will remove the message box from the window.

Enable Message Box will restore the message box to the window.

Write Values will write the current model in ASCEND to a values file, naming it modelname.values, where modelname is the user defined name for the

model. It will overwrite any existing file of that name.

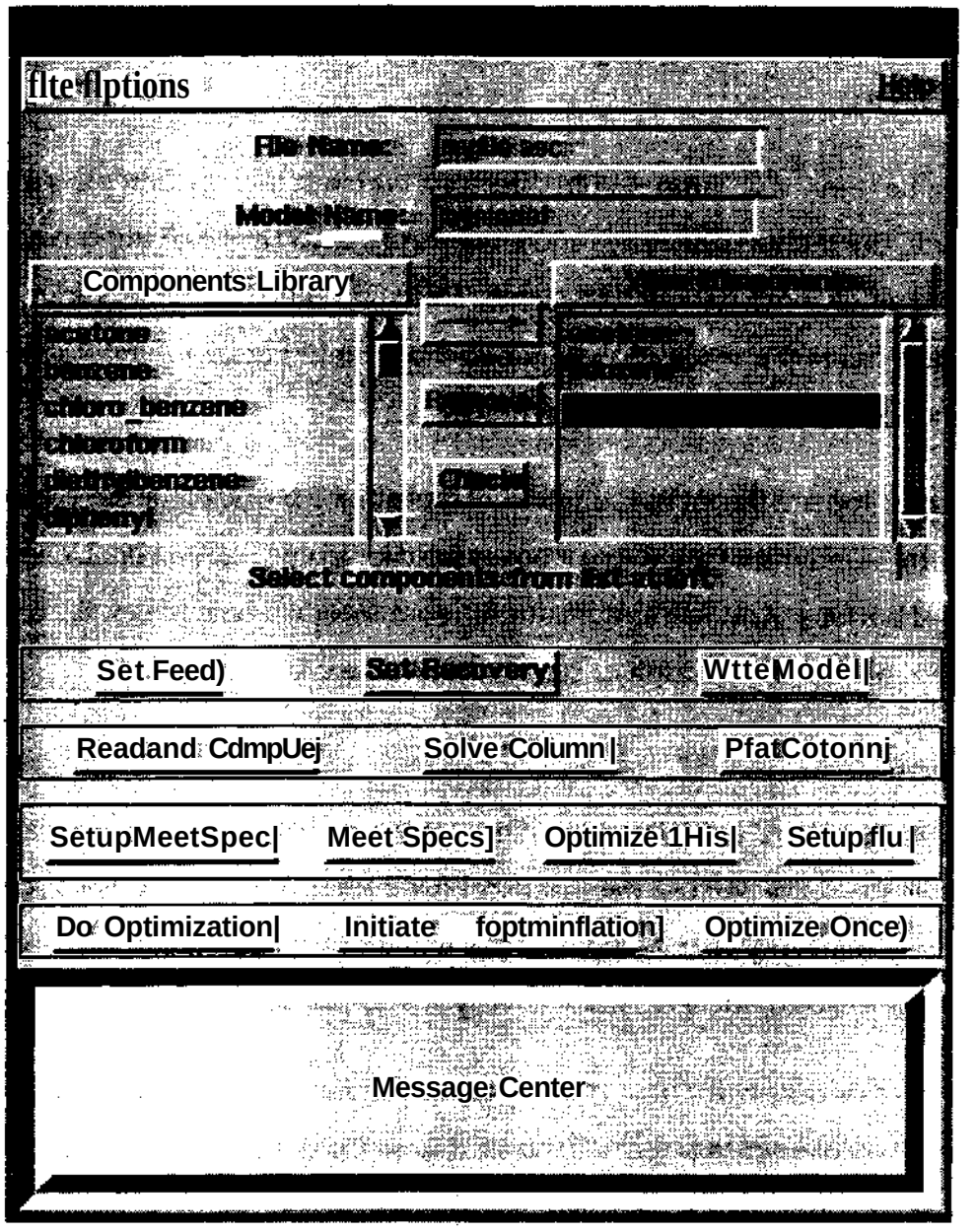

Figure 6.1. Column Design Interface

Read Values will read in the values file created by **Write Values.**

Save All Values is for saving **the current** set of **solutions for the flexible**

design problem. The flexible design problem will solve the column for the different potential feeds and create virtual saves of each solution. This will read each virtual save and create a separate values file for it.

Update Values is also for the flexible design problem. It will read in each of the values files created by **Save All Values** and create a virtual save for each of them. This is useful for continuing with a flexible design model on a different day. The user can read in the settings from the other day, read and compile the model, and then update the values.

Quit destroys the window. It does not quit ASCEND, destroy any models, nor clear any tcl variables.

Options Menu

List components alphabetically reorders the components library list to be alphabetical.

List components by boiling point reorders the components library list to be in the order of increasing boiling point.

Set globals creates an input window where the user can view or change the important variables that the application uses. Most of these can be changed through the interface, or are changed when different parts of the algorithm are executed. The variables plot_dir, values_dir, and logfile set the destination directory for plots, values and log files. The variable delta sets how big of a step the meet specifications procedure will take. Also max and min set the range of the approximation for the flexible design problem.

Color Settings creates a window for changing the color settings of the application. Standard names of colors can be input for each entry, or, by pressing Control-Mousel, the user will bring up a color select box.

Procedure Menus will create extra menus for the procedures used by the application. In most cases, clicking these procedures in their menus will not work, because they require arguments, but it is useful for the more advanced user to have a list of the procedures available.

Entry Boxes

The first thing a user should do is enter their own file and model name in the two entry boxes in the top of the window.

File Name is the entry box for the name of the file to which the model will be written.

Model Name is the entry box for the name of the model.

Component Lists

Components Library is the list of all the components currently in the ASCEND components library. The library listing can be shown either in alphabetical order or by boiling point order. The options menu allows the user to change this.

Your Components is the list of the components the user has chosen.

The user can select components from the left list box by clicking on them and then hitting the —> key. Any number of components can be selected.

The user can remove components from his own list by selecting a component in the right list and hitting the **Remove** button.

The Check button checks the components for possible azeotropes and puts them in boiling point order. This check is done against a database we created by doing infinite dilution equilibrium calculations for each binary pair. From this database we can guess if binary azeotropes or heterogeneous behavior will occur. If heterogeneous behavior is expected, we warn the user that the models will be

inaccurate, since we do not consider liquid-liquid equilibrium yet.

Settings Buttons

The **Set Feed** button brings up an entry window, asking the user first for how many feeds he wants. This is for the flexible design problem, not for multiple feeds to a single column. If a standard design is desired, the user should input 1. Then a dialog box will pop up, asking for the molar flowrate of each component in each feed. If the user wants to do a flexible design problem, he can input how every many feeds he wishes.

The **Set Recovery** button brings up a dialog box asking the user to pick the light and heavy key and set the recovery of each. The recovery is the fraction leaving the top of the column, so the light key recovery must be higher than the heavy key recovery.

The **Write Model** button will create an ASCEND model named whatever the user entered in the **Model Name** entry line in the file the user entered on the File **Name** line. If the file already exists, the model will be appended to it. If it does not already exist, it will create the file.

Creating Buttons

The **Read and** Compile button will read the file into ASCEND and compile it. It will also run the values and specify procedures and export it to the Browser. At this point, or at any later point, the user can explore and manipulate this model like any other in ASCEND.

The **Solve Column** button will run the SolvColl procedure described above. The I/O window will show the progress of this procedure.

The Plot Column button can be used at any time once the model is read and compiled in ASCEND. It will create a plot file with the same name as the model with the suffix plot, e.g. mymodel.plot. It will also try to plot it with

whatever plotting command the ASCEND user has set in the Utilities window.

Running Buttons

The **SetupMeetSpec** button gives the user 3 choices about how to approach the recovery specification/We suggest option 1 be tried first. It will attempt to meet the recovery specifications while trying to move the average slopes of the key components in the top and bottom to 0.01. This is not always achievable, but it is a reasonable starting point and will increase the number of trays in each column section. Option 2 has a similar effect, but assumes that the initial slopes are good ones. The third option is suggested as a second attempt option, after doing 1 or 2. Once a fairly large number of trays are modeled, it is reasonable to fix them and just go for the recoveries.

The **Meet** Specs button will run the MultipleMeetSpec procedure with the settings made with the **Setup MeetSpec** button. If the multiple specifications cannot be met, it will prompt the user to ask if he wants to try for each key recovery separately. Note that the settings given to MultipleMeetSpec can be accessed with the Set globals choice in the **Options** menu. They are keys (full ASCEND name of the key components), specs (specifications for the key components), and delta (initial step size towards specification).

The Optimize This button will use our gradient based optimization procedure to try to minimize the cost of the column by adjusting the number of trays and the feed tray location, while holding the recovery specifications and solving for the reflux ratio and distillate flowrate.

The Setup All button will attempt to get starting solutions for each possible feed for the flexible design problem. If it has problems, it will prompt the user for choices on meeting the specifications.

Flexible Optimization

Do **Optimization** will try to do the full flexible design optimization routine. First it will create the initial approximation and locally optimize this. Then it will move the approximation as needed and re-optimize iteratively until the local optimum is within the borders of the approximation. Then it will reduce the range of the approximation and initiate the approximation again and resolve the optimization problem.

Initiate Approximation will do only do the approximation for each of the columns and one optimization, setting up the optimization problem.

Optimize Once will find the local minimum for a particular level of approximation, moving the approximation region and re-optimizing until the minimum does not occur on a boundary of the approximation region.