Abstract

An interactive software package intended for use in conjunction with college classical control courses is presented. The package is menu-based and has many of the organizational features of a modern hand held calculator. Extensive graphical capabilities are included.

Three common types of system representations are included and the menus are organized around these representations. The menus are presented exhaustively and a design example is worked via a step-by-step procedure. The author's experience with the package is related and future enhancements and modifications are outlined.

1.0 Introduction

This paper presents an IBM personal computer based software package intended for use in senior level classical controls courses. Features of the OUCADS package include (1) calculator operations with any of three equivalent representations for single-input/single-output systems, (2) the ability to easily convert from one type of system representation to another, (3) continuous, discrete, and sampled data capability, (4) extensive and flexible graphic capabilities, and (5) linear simulation capability.

A significant barrier to controls education has always been the tedious calculations and graphical constructions required to apply the methods of classical control system design and analysis. Although this problem has been somewhat alleviated by the availability of commercial linear systems analysis software, there exists a need for software which (1) is dedicated to classroom use, (2) is inexpensive, (3) can be used without learning what is effectively a new programming language, and (4) is sufficiently sophisticated and flexible that realistic problems can be solved.

This paper presents a summary of the capabilities of OUCADS by outlining a menu structure which is almost identical for each type of system representation. The concept of calculator-type system manipulations is presented and an exhaustive discussion of the available features for each representation type is given. Finally, a simple design example is presented which involves designing a digital controller to achieve specified phase margin and gain crossover criteria for a sampled data system. The example serves to illustrate most of the features of OUCADS including transfer function manipulation, frequency response generation, root locus calculation, linear simulation, and graphical capabilities.

2.0 Main Menu and Summary of Capabilities

The main menu contained in Figure 1 is displayed immediately after invoking OUCADS and contains four major sections. These sections are quite independent functionally in that, for example, if the user is interested only in transfer function models then the menu items under the "Transfer Function" heading are the only ones required. The menu headings for each of the system representation types contain almost identical menu items.

*** MAIN MENU ***

Transfer Function
1. Create A
2. Create B
3. Manipulation
4. Linear simulation
5. Input/output
10. Create A
11. Create B
12. Manipulation
13. Linear simulation
14. Input/output

Frequency Response
6. Generate A
7. Generate B
8. Manipulation
9. Input/output
15. Root locus analysis
16. Design a compensator
17. QUIT

Figure 1. OUCADS Main Menu

The first two items "Create A" and "Create B" require some explanation. Two
separate sets of data for each system representation type can exist simultaneously. These two sets can be thought of as "registers" which are similar to the familiar data registers in electronic calculators except that in the case of OUCADS, the registers contain system representations (state-space, transfer function, or frequency response data) instead of floating point numbers. The menu items "Create A" and "Create B" for the "State-Space" heading and the "Transfer Function" heading in turn produce menus which are identical, the only difference being into which register the data is placed.

The "Manipulation" items produce submenus which allow for register interchanges and algebraic operations on the appropriate registers. As would be expected, some of the operations use both registers for input while others use only one register. The "Manipulation" menus for the three representation types are very similar.

The "Input/Output" items provide submenus for storing and retrieving registers to disk storage, screen display or hardcopy output, and graphical output in the case of a frequency response representation. Again, these submenus are very similar for each system representation type.

"Linear Simulation" items are also provided for the state-space and transfer function representations. Graphical and other output is also provided for the simulation results via a submenu.

The "Root Locus Analysis" and "Design a Compensator" items are listed separately. The root locus capability is provided for the transfer function data type and includes graphical and other output. The compensation design item is a placeholder for interfacing with student-written modules.

One feature of OUCADS that is unique is the fact that, although the package supports continuous-time, discrete-time, and sampled-data systems, there is no explicit difference in these system types throughout the program. The original motivation for this approach was the very significant savings in code complexity. However, there are also pedagogical advantages, since the student is thereby encouraged to appreciate the essential unity of linear systems analysis.

The only points in the OUCADS analysis process where the distinction between continuous and discrete systems is important algorithmically are (1) frequency response generation, (2) linear simulation, (3) root locus analysis (and there only in order to correctly interpret the results), and (4) compensator design. Of course, the responsibility for correctly manipulating the component continuous and discrete subsystems of a sampled data system is thus placed on the user.

3.0 Transfer Function Menus

The first two main menu items dealing with transfer function representations are denoted "Create A" and "Create B". These functions are accomplished with concise "question and answer" sessions. The major main menu items are the "Linear Simulation", "Manipulation", and "Input/Output" items. The choice of each of these menus in turn invokes a submenu. The structure of the submenu is indicated in Figure 2.

```
Figure 2. Structure of Menus Accessible from Main Menu Items under the Heading "Transfer Function"
```

The transfer function manipulation menu is given in Figure 3.

```
TRANSFER FUNCTION MANIPULATION MENU
1. Interchange A and B
2. Set A = B
3. Set B = A
4. Reciprocate A
5. Reciprocate B
6. Add A to B, store in A
7. Multiply A by B, store in A
8. Divide A by B, store in A
9. Add a constant to A
10. Add a constant to B
11. Multiply A by a constant
12. Multiply B by a constant
13. Subtract A-B, add to A
14. Return to MAIN MENU
```

Please enter your choice now.

```
Figure 3. Transfer Function Manipulation Menu
```
The most important feature of this menu is what it indicates regarding the register/calculator structure of OUCADS. For example, many of the features associated with modern hand-held calculators are present such as register interchanges and pushing onto and popping from the stack \((B=A\) and \(A=B\)). Also, operations requiring only one register as an operand modify only that register. The register \(A\) performs an accumulator function for dual operand operations, \(i.e.,\) the results are always placed in the \(A\) register. An important point to note is that registers \(A\) and \(B\) can both contain improper rational functions. This feature is useful during intermediate calculations; for example, when closing a feedback loop.

The linear simulation menu of Figure 4 offers provision for the time domain simulation of the step and ramp response of a system whose transfer function representation is contained in either register \(A\) or register \(B\). Also, depending on the menu option selected, the transfer function is interpreted as representing either a discrete-time or continuous-time system.

**Figure 4. Linear Simulation Menu**

The Input/Output menu item in turn invokes the menu of Figure 5. This menu provides for storage (in sequential files) and retrieval of simulation results.

**Figure 5. Simulation Input/Output Menu**

There is also provision for graphical output via the "Plot of \(A\)" and "Plot of \(B\)" options, each of which generates the menu of Figure 6.

**Simulation Response Plot Menu**

1. Set Manual Scaling
2. Display Plot
3. Print Plot
4. Generate HPGL Meta File
5. Return to Simulation I/O Menu

Please enter your response now

**Figure 6. Simulation Plot Menu**

There are three main features of this particular plot menu. First, the simulation data can be displayed on the installed monitor display. An autoscaling algorithm is provided so that the manual scaling option can be used more intelligently. Second, the simulation results can be plotted on the installed graphics hardcopy device. Finally, an HPGL graphics metafile can be produced. The HPGL file can then be used to generate plots using HPGL compatible pen plotters or imported to certain word processors to provide presentation quality graphics. For example, many of the figures in this document were generated as HPGL metafiles, imported to Wordperfect, incorporated into the document, and printed using an HP Laserjet Series II printer. The need to "cut and paste" is thereby eliminated and due to the nature of pen plotter commands, the full resolution of the final graphics output device can be preserved.

The remaining transfer function menu is the "Input/Output" item in the main menu. Its invocation results in the menu of Figure 7. The major features of this menu are the ability to store and retrieve transfer functions to and from disk and the ability to examine and print the contents of transfer function registers.

**Figure 7. Transfer Function Input/Output Menu**

4.0 Frequency Response Menus

Menu items 6 through 9 are intended for use with frequency response representations of systems. As with the transfer function representations, a
register approach is used. Also, the frequency responses of registers A and B are generated based on the transfer functions of transfer function registers A and B, respectively. In other words, the selection of item 6, "Generate A", takes the transfer function in register A, say \( G(s) \), and calculates \( G(j\omega) \) for a user specified range of frequencies and places the result in the frequency response register A. The structure of the frequency response menus is illustrated by Figure 8.

![Figure 8. Structure of Menus Accessible from Main Menu Items under the Heading "Frequency Response"](image)

There are three ways to calculate a frequency response from a transfer function as is indicated by the menu invoked by "Generate A" or "Generate B" and reproduced in Figure 9.

**FREQUENCY RESPONSE GENERATION MENU**

1. Z-domain frequency response
2. Sampled-data frequency response
3. Continuous frequency response
4. Return to MAIN MENU

Please enter your choice now.

**Figure 9. Frequency Response Generation Menu**

The first menu item in Figure 9 is the "Z-domain frequency response" selection. This selection interprets the transfer function as that of a discrete-time system \( G(z) \) and calculates \( G(e^{j\omega}) \) for a user-specified range of frequencies. The number of frequency points is user-specified and limited to 100 points. The frequency points are logarithmically spaced. This menu item is most commonly used to calculate the frequency response of digital filters used for compensation purposes.

The second menu item, "Sampled-data frequency response", calculates the frequency response of the continuous-time block \( G(s) \) preceded by an ideal sampler and zero order hold device \( G_{OH}(s) \) by a truncation of the infinite series

\[
G'(j\omega) = \frac{1}{\pi} \sum_{n=-\infty}^{\infty} G_{OH}(j\omega+j\pi n)G(j\omega+j\pi n) \quad (1)
\]

\[ + \frac{g(\omega)}{2} \]

The number of terms retained in the truncation is adaptive and dependent on the relative sizes of consecutive partial sums. The number of terms retained is generally different for each frequency point. The series method is an attractive alternative when a pulse transfer function representation is not available.

The third and last calculation option is a straightforward calculation of \( G(j\omega) \) for an appropriate frequency range.

The frequency response manipulation menu is given in Figure 10 and is almost identical to the corresponding menu for transfer function manipulations.

**FREQUENCY RESPONSE MANIPULATION MENU**

1. Interchange A and B
2. Set \( A = B \)
3. Set \( B = A \)
4. Reciprocate A
5. Reciprocate B
6. Add A to B, store in A
7. Multiply A by B, store in A
8. Divide A by B, store in A
9. Add a constant to A
10. Add a constant to B
11. Multiply A by a constant
12. Multiply B by a constant
13. Return to main menu

Please enter your choice now.

**Figure 10. Frequency Response Manipulation Menu**

The "Input/Output" item under the "Frequency Response" heading leads to the menu of Figure 11. Again, responses may be stored from frequency response registers A and B and retrieved from disk into registers A and B. These registers may also be examined and printed.

**FREQUENCY RESPONSE I/O MENU**

1. Retrieve A
2. Retrieve B
3. Store A
4. Store B
5. View or Print A
6. View or Print B
7. Semilog plot of A
8. Semilog plot of B
9. Polar plot of A
10. Polar plot of B
11. Return to MAIN MENU

Please enter your response now.

**Figure 11. Frequency Response Input/Output Menu**
There are two major plot submenus generated under the "Input/Output" menu. The first of these is the semilog plot menu given in Figure 12.

**SEMILOG (BODE) PLOT MENU**
1. Set manual scaling
2. Display plot
3. Print plot
4. Toggle frequency units
5. Generate HPGL Meta File
6. Return to I/O MENU

Please enter your choice now

**Figure 12. Semilog Plot Menu**

The semilog plot routines are highly automated. Manual scaling is provided although it is seldom necessary. The default frequency units are radians/second. Menu item 4 can be used to change the displayed frequency units.

Display, hardcopy, and HPGL metafile capabilities are provided. A convenient feature of all plot routines in OUCADS is that any plot can be configured with regard to scaling, frequency units, etc., using the graphics display and then printed or stored as a metafile on disk exactly as the plot was last displayed.

The polar plot menu is given in Figure 13. The polar plot produced is somewhat unusual in that the frequency response points are not simply plotted in polar coordinates. Rather, the coordinates are a modified form of polar coordinates in which the axes are linearly spaced in decibels, resulting in a log-log set of axes. The origin is chosen to correspond to the smallest magnitude appearing in the frequency response data. The overall result is a plot which contains the entire frequency response.

**POLAR (NYQUIST) PLOT MENU**
1. Set manual scaling
2. Set constant magnitude circle
3. Set constant phase line
4. Set constant M-contours
5. Set special frequency
6. Display plot
7. Print plot
8. Generate HPGL Meta File
9. Return to I/O MENU

Please enter your choice now

**Figure 13. Polar Plot Menu**

There are a relatively large number of options available for polar plots. In addition to the options provided for the semilog plot, there are provisions for drawing constant magnitude circles which are useful for determining gain margins. Constant phase angle radial lines can also be drawn and these are useful for determining phase margins. User specified frequencies can also be marked and labeled. This feature is useful in determining phase and gain crossover frequencies. For unity gain feedback configurations, constant closed loop magnitude contours can be displayed. These contours can be used to find the closed loop bandwidth and the peak frequency response value. They can also give indications as to the response complexity.

5.0 **State Space Menus**

Menu items 10 through 14 are intended for state space system representations. The inclusion of the state space representation is motivated by the need for sampled-data simulation capability. By combining the manipulation capabilities and the ability to calculate a discrete state representation of a continuous block preceded by an ideal sampler and zero order hold, a discrete state space representation can be calculated for an interconnected system for which a pulse transfer function exists.

The main menu items "Create A" and "Create B" both invoke the menu of Figure 14.

**CREATE STATE SPACE MATRICES MENU**
1. Create A from user input
2. Create A from transfer function A
3. Create A from transfer function B
4. Return to MAIN MENU

Please enter your response now

**Figure 14. Menu for Creating State Space System Representations**

Realizations can be created either manually with menu item 1 or from the transfer functions stored in either of the transfer function registers with menu items 2 or 3.

The state space manipulation menu invoked by main menu item 12 is given in Figure 15. The only difference between this and the transfer function manipulation menu of Figure 3 is the ability to calculate discrete state space representations from continuous time representations. Of course, it is assumed that the continuous time block represented by the continuous time representation is preceded by an ideal sampler and zero order hold.
STATE SPACE REALIZATION MANIPULATION MENU

1. Interchange A and B
2. Set A = B
3. Set B = A
4. Reciprocate A
5. Reciprocate B
6. Add A to B, store in A
7. Multiply A by B, store in A
8. Divide A by B, store in A
9. Add a constant to A
10. Add a constant to B
11. Multiply A by a constant
12. Multiply B by a constant
13. Subtract A - B -> A
14. Discretize A
15. Discretize B
16. Return to MAIN MENU

Figure 15. State Space Realization Manipulation Menu

Although the state space manipulation menu is similar in appearance to the transfer function manipulation menu, there is no way to represent systems with improper transfer functions in state space form. Therefore, there are some limitations on the use of the state space manipulation capabilities that are not present for transfer function representations. In particular, any operation which requires calculating a realization corresponding to the inverse of a strictly proper (low-pass) transfer function will result in an error. Such a situation may be encountered in any of menu items 4, 5, and 8 of Figure 15.

The simulation option of main menu item 13 invokes simulation and simulation Input/Output menus that are identical to those invoked by main menu item 4. The only difference is algorithmic, since here the state space realizations are used to calculate the various responses, instead of the transfer functions.

Figure 16. Structure of Menus Accessible from Main Menu Items under the Heading "State-Space Realization"

The state space "Input/Output" option of item 14 in the main menu results in a menu identical to that of Figure 7. Again, disk storage and retrieval is provided, as well as display and hardcopy capability.

The structure of the menus reachable from the main menu items under the heading "State-Space Realization" is indicated by Figure 16.

6.0 Root Locus and Compensator Design Menus

The root locus and compensator design menus belong most properly under the "Transfer Function" heading of the main menu. They appear as separate menu items for two reasons. First, the compensator design item is not actually an active menu. It is a placeholder for a student written module whose functions can be specified by the instructor. The root locus menu stands alone in the interest of standardization of the "Transfer Function" and "State Space Realization" headings.

The root locus menu invoked by main menu item 15 is given by Figure 17. The first two items of the root locus menu select which of the transfer function registers are to be used as \( P(s) \) in the standard root locus formulation in which the solutions of

\[
1 + KP(s) = 0
\]

are calculated for a range of values of \( K \).

Figure 17. Root Locus Analysis Menu

The third and fourth items of Figure 17 allow the user to specify the values of \( K \) for which the solutions of Equation 2 are calculated. The fifth and sixth items provide display or hardcopy output of the roots.

The seventh item invokes the plot submenu of Figure 18. In addition to providing a basic plot of the roots and the display, hardcopy, and metafile capabilities of the frequency response and
simulation plots, several post-analysis tools are also provided. These tools include the ability to display constant damping ratio, constant frequency, and constant damping factor lines for both \(z\)-plane and \(s\)-plane interpretations.

**PLOT MENU**

1. Manual scaling
2. Constant zeta line
3. Unit circle
4. Constant frequency
5. Constant damping factor
6. \(z\)-plane interpretation
7. Display plot
8. Print plot
9. Generate HPGL Meta File
10. Return to ROOT LOUIS MENU

Please enter your choice now.

**Figure 18. Root Locus Plot Menu**

### 7.0 Design Example

The most concise way to illustrate the capabilities of OUCADS is to present an example of the type of design problem that is typically encountered in a senior level classical controls course.

The system feedback configuration is that of the error-sampled unity feedback system of Figure 19. The continuous plant transfer function is given by

\[ G(s) = \frac{1}{s^2}. \]  

The second order compensator is a two stage (lag-lead) compensator designed to provide a 40 degree phase margin at a gain crossover frequency of 1.0 radians/second.

\[ D(z) = D_1(z)D_2(z) \]  

where

\[ D_1(z) = -0.3192z + 0.311 \]  
\[ -1.012z + 1 \]  

\[ D_2(z) = -10.42z + 9.78 \]  
\[ -1.622z + 1 \]  

A semilog plot of the compensated open loop frequency response is given by Figure 20. The corresponding polar plot is given by Figure 21, where the phase margin is indicated by the intersection of the \(-140^\circ\) radial line, the frequency response, and the 0 decibel circle. The gain crossover frequency \(\omega_c\) is also denoted by a symbol and labeled.

Constant closed loop magnitude contours are used in Figures 22 and 23 to denote the bandwidth \(w_B\) and the frequency of peak closed loop magnitude \(\omega_p\), respectively.

**Figure 20. Design Problem Open Loop Compensated Frequency Response**

**Figure 21. Design Problem Open Loop Compensated Frequency Response**

416
Figure 22. Magnitude Contours with \( \omega_p \) Denoted

Figure 23. Magnitude Contours with \( \omega_p \) Denoted

Figure 24 is a semilog plot of the closed loop frequency response from which the bandwidth and peak frequency response values can be verified. Although the closed loop system is not second order, the designer might expect the settling time in response to a step input to be on the order of 10 seconds. This is verified by the simulation results of Figure 25 where it can be seen that the settling time is actually less than the 10 seconds estimated. Also of interest is the root locus plot of Figure 26, where the closed loop poles are plotted for a range of \( K \) of \(.1 \) to \(10\). At low \( K \) values (roughly \(.1\)) two of the roots are very close to the unit circle and at high values (roughly \(10\)) two poles are actually outside the unit circle. This serves to verify the upper and lower gain margins of roughly 20 decibels and -20 decibels, respectively, that can be obtained from the polar plots of Figures 21-23.

8.0 Classroom Experience with OUCADS

OUCADS has been used in various versions for parts of three academic years in the Electrical and Computer Engineering department at Ohio University. The experience of the author is that its use has helped to maintain the interest of the students in design at a fairly high level. Unfortunately, their motivation for
completing assignments that require hand calculations has suffered. Since Ohio University does not have sufficient facilities to provide a computer to every student during testing sessions, either students must develop and maintain their skills at hand calculations or their tests must be designed to account for the fact that most of their problem solving experience has been in front of an OUCADS menu. The author's solution to this problem has been a combination of methods. First, a fair number of hand calculations are required before students are allowed to turn in completed assignments using OUCADS. For example, students are still required to construct Bode plots, Nyquist diagrams, root loci, and to perform design calculations by hand. In addition, they are required to write their own computer codes for linear simulation and compensator design. These codes must be compatible with OUCADS. Finally, after students have demonstrated their proficiency in the various constructions and calculations, they are supplied with data and plots from OUCADS during subsequent examinations.

Overall, the author's experience with OUCADS in the classroom has been positive. Not only can more ambitious analysis and design problems be assigned, but the use of OUCADS with an overhead projector saves a great deal of the lecture time previously used to construct graphs on the board or to perform routine calculations.

9.0 Future Enhancements and Modifications

Although OUCADS is not intended to be a general purpose analysis and design tool, there are several minor capabilities and modifications that would enhance its usefulness as a teaching tool. Probably the most important of these is a general re-design of the root locus capability. Current plans are to include the root locus as two separate items, "Root Locus P(s) = A" and "Root Locus P(s) = B", under the "Transfer Function" heading of the main menu. Some work on the graphical capability is also necessary. For example, it is awkward to use the manual scaling capability of the root locus plot menu to "zoom up" a section of the root loci plot for closer inspection. The ability to label roots corresponding to a specified K value would also be desirable.

Future plans also call for more extensive file handling capability. Currently, each type of stored data carries a unique suffix. However, no provision for examining say, all of the transfer functions in the working directory is included. Possible enhancements would include this ability and the ability to delete files without terminating execution.

Another modification involves providing an option in the state space and transfer function manipulation menus for directly calculating closed loop transfer functions and state space realizations. Problems can occur when the user attempts to simulate responses for non-coprime transfer functions or non-minimal state space realizations, especially when the noncontrollable and/or nonobservable modes are unstable. These problems can arise when using the present algebraic manipulation features.

Finally, the present limits on system order will be increased. These limits were maintained at a low value during the development of OUCADS since the total memory requirements were unknown. The maximum order of a transfer function or state space representation will be increased from 10 to 30 and the number of points allowed in a frequency response will be increased from 100 to 300.

10.0 Conclusions

An interactive, menu-driven software package intended for use in classical control courses has been presented. The menus have been listed exhaustively and examples of the graphical capabilities have been presented via a design example.

The package is based on the concepts of calculator type operations on "registers" which contain any of three types of system representations. This type of data organization allows the user to easily enter a system representation, operate on that representation, and display the results in a very concise manner.

The package is capable of handling transfer function, frequency response, and state space representations and can produce root loci plots, frequency response plots in both polar and semilog coordinates, and step and ramp time response plots. Other display and hardcopy capabilities are also included in OUCADS, as well as the ability to store each of the data types on disk for later retrieval and use.

Various enhancements and modifications are planned. In particular, the maximum size of the system representations will be increased and the root locus plot capabilities will be expanded.