MOCAD: A Graphic Predetermined Time Standards Software for the IBM P.C.

Raad A. Dawood

Western Michigan University

Follow this and additional works at: http://scholarworks.wmich.edu/masters_theses

Part of the Computer Sciences Commons, and the Industrial Engineering Commons

Recommended Citation

http://scholarworks.wmich.edu/masters_theses/1048

This Masters Thesis-Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Master's Theses by an authorized administrator of ScholarWorks at WMU. For more information, please contact maira.bundza@wmich.edu.
MODCAD: A GRAPHIC PREDETERMINED TIME STANDARDS SOFTWARE
FOR THE IBM P.C.

by

Raad A. Dawood

A Thesis
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Master of Science
Department of Industrial Engineering

Western Michigan University
Kalamazoo, Michigan
June 1990
MODCAD: A GRAPHIC PREDETERMINED TIME STANDARDS SOFTWARE FOR THE IBM P.C.

Raad A. Dawood, M.S.
Western Michigan University, 1990

MODCAD is a software program designed to give MODAPTS (MODular Arrangement of Predetermined Time Standards) the capability to generate workplace layouts that can be used in conjunction with the time study analysis. Software packages have been developed for MODAPTS such as MODAPTS Plus and Task Master that allow the user to input the predetermined codes of MODAPTS.

AutoCAD (a computer aided design software) is used to combine MODAPTS codes and workplace layout drawings in a single environment by customizing the standard AutoCAD menus and icons to include the MODAPTS codes and routines. AutoLisp (AutoCAD resident language) is used to write the routines to handle the report generation of a MODAPTS analysis.

MODCAD will give the user the tool to generate a workplace layout which can be used to extract exact attributes needed in the MODAPTS analysis.
ACKNOWLEDGEMENTS

I wish to express a special acknowledgement and sincere appreciation to my advisor, Dr. Robert Wygant, for his assistance, direction, and support throughout the development of this software; and to my committee members, Dr. Bob White, Dr. Bruce Fischer, and Dr. Donna Kaminski for their advice and assistance.

Appreciation is expressed also to Mr. Greg Lozeau for his support, and for providing the computer facilities that made this software possible.

Raad A. Dawood
INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road Ann Arbor MI 48106-1346 USA
313 761-4700 800 521-0600

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
MODCAD: A graphic predetermined time standards software for the IBM P.C.

Dawood, Raad A., M.S.
Western Michigan University, 1990
TABLE OF CONTENTS

ACKNOWLEDGEMENTS.......................................................... ii
LIST OF FIGURES................................................................. iv
INTRODUCTION................................................................. 1
STRUCTURE OF THE SOFTWARE............................................. 4
STRUCTURE OF THE MODCAD MENU....................................... 5
STARTING UI: THE SOFTWARE............................................. 11
USING MODCAD............................................................... 12
  Using the Graphics Portion............................................. 12
  A Brief Description of MODAPTS Codes............................. 14
  Using the MODAPTS Portion........................................... 21
  Using Layout Assisted Gets and Puts............................... 29
EDITING THE FILES.......................................................... 30
CONCLUSIONS................................................................. 31
APPENDIX: MODCAD Computer Code Listing............................ 33
BIBLIOGRAPHY............................................................... 73
LIST OF FIGURES

1. MODCAD Menu Setup................................. 6
2. AutoCAD Menu Setup................................. 13
3. Side View of Layout................................. 15
4. Top View of Layout................................. 16
5. Moves With Smaller Articles Display............... 17
6. Moves With Larger Articles Display............... 18
7. Gets for Small Articles Icons.................... 19
8. Gets for Large Articles Icons.................... 20
9. Puts for Small Articles Icons.................... 22
10. Puts for Large Articles Icons.................... 23
11. Mental Tasks Icons................................. 24
12. Finger/Hand/Wrist Use Icons...................... 25
13. Foot Use Icons.................................... 26
14. Body Use Icons.................................... 27
INTRODUCTION

MODAPTS (MOdular Arrangement of Predetermined Time Standards) is a third generation predetermined time system (PTS) used for: a) calculating reliable production standards, b) improving an organization's productivity, c) analyzing departmental efficiency, and d) improving employee relations (Gerber, 1989).

The original design of MODAPTS was developed by Mr. G.C. Heyde (1983). It consisted of predetermined codes for various human activities from moving parts of the body to doing specific operations such as using a typewriter.

There are several computer versions of MODAPTS. One of the first systems was MODAPTS PLUS Professional which consists of three programs which operate on "PC-compatible" computers (Heyde, 1983). The program is written in BASIC computer language. It allows the user to input the codes for various tasks one step at a time until the complete operation is accomplished. The computer then will take these codes and will combine them with allowances and other factors in order to come up with a standard time for that task in a unit called MOD (0.129 Seconds). The software offered a utility to edit the input and to provide an output that contains the steps of the task and the final cycle time.
CAESAR (Computer Assisted, Engineered Standards And Rates) is an interactive, multi-level, standard data system for improving methods and establishing production piece rates in the sewn products industry (Irwin, 1989).

A more advanced version of MODAPTS was developed which combined the use of a mouse with data input in order to ease the operation of manipulating the data and files. It was called "Task Master." The software consisted of pull down menus and simple graphical representations of the various codes. This software as well as the earlier version required the user to memorize the various codes and the way to input them (Shinnick, 1989).

A layout of a workplace is always helpful in order to visualize the locations of the operator, furniture, and tools. In order to use graphics to sketch the workplace, the user can either draw the layout manually or use a CAD software. The MODAPTS analysis then has to be done separately without any direct connection with the graphics.

MODCAD is a software that contains graphics and MODAPTS in a single environment. It is written in AutoLisp (the resident language of AutoCAD). The software has many Lisp routines for both graphics and MODAPTS analysis. They can be used simultaneously, along with predetermined graphic figures, to create a simple or a complex layout that has a direct connection with the MODAPTS analysis. The user can obtain accurate information about distances,
weights, and other attributes either by looking at the model and input them, or by extracting them automatically.

MODCAD offers icons that have symbols of the MODAPTS codes as well as help windows and pull down menus to aid the user to create an analysis with ease and with minimum knowledge of the software.
STRUCTURE OF THE SOFTWARE

The software consists of two portions: the graphics portion and the MODAPTS portion. The graphics portion is mainly AutoLisp routines to create simple geometric shapes with text attributes which can be entered by the user for either actual physical dimensions, weight, or other relevant attributes. The other graphical tools are a multitude of predetermined blocks that contain many shapes such as human figures, tools, and furniture.

The MODAPTS portion consists of AutoLisp routines to prompt the user to input the appropriate code, an explanation of the task, and the frequency. The tasks can be easily identified by the icons that pop up automatically which illustrate every task graphically.

The user can choose the pull down menus to select either a MODAPTS routine or a drawing routine. The layout is to scale, and the user can obtain accurate distances automatically from the layout or manually by specifying two points on the screen. Other AutoLisp routines are written to aid in editing, listing, plotting, and printing the final analysis and layout.
STRUCTURE OF THE MODCAD MENU

The MODCAD menu (Figure 1) consists of standard AutoCAD routines and customized routines for MODAPTS. The functions in the menu are used for two purposes: (1) drawing and inserting figures, and (2) selecting appropriate MODAPTS codes. The menu portion on top of the screen is a pull down type while the menu portion on the right of the screen is a flat menu. Icons that carry the MODAPTS code symbols will be activated via the pull down menu.

The pull down menu structure carries functions for MODAPTS, drawing aids, and file services. They are broken down into the following sub menus:

1. "Tools" sub menu contains drawing aids that allow the user to generate drawings accurately. They help in locating starting points and terminating points for drawing primitives. The user can select from the following functions: (a) center, (b) endpoint, (c) intersection, (d) midpoint, (e) nearest, (f) perpendicular, and (g) tangent.

2. "Obj. cts" sub menu allows the user to insert blocks such as human figure, tools, and furniture. It also allows the user to insert symbols that contain attributes of type and weight of object via a dialogue window. The user can
### MENTAL TASK
- Hand/Finger Use
- Foot Use

### WEIGHT FACTOR
- Extra Force

### ALLOWANCES

**Figure 1. MODCAD Menu Setup.**
store a drawing of any object to be retrieved in the same
drawing session or in later drawing sessions by creating a
block or a wblock (block written to disk). The following
functions can be selected from this sub menu: (a) insert
a block, (b) objects with attributes, (c) create a block,
and (d) create a wblock.

3. "Gets" sub menu contains the functions to select
the GET routine for small or large articles. It also
allows the user to use layout assisted GET. The following
functions can be selected from this sub menu: (a) small
articles, (b) large articles, and (c) form layout.

4. "Puts" sub menu contains the functions to select
the PUT routine for small or large articles. It also
allows the user to use layout assisted PUT. The following
functions can be selected from this sub menu: (a) small
articles, (b) large articles, and (c) form layout.

5. "Other elements" sub menu contains the rest of
MODAPTS codes and special elements. The user can select
the appropriate body part that performs the task.
Allowances can be assigned for the task by using this sub
menu. The following functions can be selected from this
sub menu: (a) mental task, (b) hand/finger use, (c) foot
use, (d) body use, (e) weight factor, (f) extra force, and
(g) allowances.

6. "Text file" sub menu is used to manage the MODAPTS
text report that contains the list of tasks, analysis, and
information about the operation. The user can select from the following functions: (a) start a new file, (b) append an existing file, (c) list a single step, (d) list all steps, (e) edit data file with "edlin," (f) display results, and (g) create an output file.

7. "Drawing file" sub menu is used to manage the workplace layout drawing. The following can be selected: (a) save, and (b) end.

The flat menu structure that is located on the right of the screen is mainly used to enhance the drawing. The user can select any of the menu functions to start a new drawing or to add to an existing drawing. The flat menu structure consists of the following functions:

1. "Dim" sub menu allows the user to scale objects in the drawing. The user can choose appropriate dimension formats to suit the type, location, or alignment of the drawing. The following dimension formats are included: (a) horizontal, (b) vertical, (c) aligned, (d) rotated, (e) radius, (f) diameter, (g) leader.

2. "Text" sub menu allows the user to add text to the drawing in the appropriate format depending on the location of the text relative to objects. The user can also select a text style from 6 available fonts. The following functions are included in this sub menu: (a) dynamic text, (b) text, (c) centered, (d) right, (e) aligned, (f) middle, (g) fit, (h) italic, (i) italic, (j) roman, (k) romans,
(1) romant, (m) standard.

3. "Draw" sub menu allows the user to create lines, arcs, and circles. It also allows the user to hatch a drawing with several standardized AutoCAD hatch patterns.

4. "Grid" sub menu allows the user to create a grid by simply specifying the width and height of the layout in feet. A grid will be created automatically that divides the screen into squares of one foot in area each. The grid can be turned on or off at any time.

5. "Zoom" sub menu allows the user to zoom in and out of a drawing. The following zoom formats can be used: (a) all, (b) center, (c) dynamic, (d) left, (e) previous, and (f) window.

6. "Color" sub menu allows the user to change the color of the primitives to be drawn. The following colors can be selected: (a) red, (b) yellow, (c) green, (d) cyan, (e) blue, (f) magenta, and (g) white.

7. "L-type" sub menu allows the user to select the line type for drawing primitives. The following types of lines can be selected: (a) contin., (b) dashed, (c) hidden, (d) center, (e) phantom, (f) dot, (g) dashdot, (h) border, and (i) divide.

8. "Edit" sub menu allows the user to make modifications in the drawing. The following functions can be selected: (a) erase, (b) move, (c) copy, (d) rotate, (e) array, (f) property, (g) fillet, (h) mirror, (i) trim,
(j) extend, and (k) redraw.

9. "AutoCAD" sub menu allows the user to utilize AutoCAD standard menu to create complex drawings and shapes. The user can return to MODCAD by simply typing "MODCAD" on the Command prompt.
STARTING UP THE SOFTWARE

In order to run the software, the following items must be available:

1. AutoCAD (Release 10) loaded in "ACAD10" directory.
2. IBM AT or compatible with 640k RAM and 20Mb hard drive.
3. Tablet or mouse.

Loading up the software is accomplished by the following steps:

1. Change the directory to ACAD10.
2. Insert the program disk in the floppy drive "A".
3. Type "a:Install".

The software will be installed on the (MODCAD) directory and can be executed by typing "MODCAD" on the C> prompt.
USING MODCAD

Once AutoCAD is activated after typing "MODCAD", the menu will be changed from the standard AutoCAD menu (Figure 2) to the customized MODCAD menu (Figure 1). The menu contains utility options, graphics, and MODAPTS routines as well as graphical blocks that contain predefined shapes.

MODCAD can be used to obtain a MODAPTS analysis without having to create any drawing. The user can ignore the graphic routines and can use the MODAPTS routines only. In this case, a blank drawing must be created at the beginning of the session.

Using the Graphics Portion

Once the graphics "Objects" menu is selected, the user can select to draw shapes and figures. Predetermined figures can be selected by inserting the blocks that contains the shapes. The user can also use the computer assisted figure drawing routine to create symbols that contain descriptions and weights of objects.

Other graphic utilities such as drawing tools can be accessed to assist in drawing accurate shapes. A routine is written to construct a colored grid, scaled in feet, that represents the actual area of the layout.

The user can manipulate the graphics by a utility menu.
Figure 2. AutoCAD Menu Setup.
that allows editing the graphics. Erasing, rotating, moving, copying, and other utilities are available to manipulate the layout. Two examples of layouts are illustrated in Figures 3 and 4.

The "Objects" sub menu contains routines to store user-drawn objects in temporary or permanent files in order to reuse the objects in the present drawing session or in future drawing sessions.

A Brief Description of MODAPTS Codes

MODAPTS codes may be classified by two types: (1) Gets and Puts, and (2) Other elements. The difference between the two is that Gets and Puts must have a MOVE code on the left in order to be used as complete codes. For example, G1 (simple grasp get), or P2 (put with feedback) can not be used without a MOVE code such as M1 (finger move) on the left. The appropriate code for Gets and Puts must look like M1G1 or M1P2. The other elements such as mental tasks, or foot use can be used directly.

MOVE codes are of two types: (1) Moves with smaller articles, and (2) Moves with larger articles. The moves with larger articles may utilize a single hand or two hands. Figures 5 and 6 show the different MOVE codes in relation to distance moved and to mass of object.

Gets and Puts codes as illustrated in Figures 7, 8, 9, and 10 represent the type of the move and the time.
Figure 3. Side View of Layout.
Figure 4. Top View of Layout.
<table>
<thead>
<tr>
<th>Elements</th>
<th>Body Part Moved</th>
<th>Usual Distance (in.)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M1</strong>=FINGER 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>M2</strong>=HAND PALM 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>M3</strong>=FOREARM WRIST 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>M4</strong>=FULL ARM FORWARD ELBOW 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>M5</strong>=FULL ARM OUTWARD SHOULDER 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>M7</strong>=BODY TRUNK MOVEMENT BODY 30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Moves With Smaller Articles**

Command:

**Figure 5.** Moves With Smaller Articles Display.
### Elements

<table>
<thead>
<tr>
<th>Elements</th>
<th>Maximum Distance (in.)</th>
<th>Limit Mass (lb.)</th>
<th>Associated with Puts</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2=Hand</td>
<td>2 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3=Forearm</td>
<td>6 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M4=Arm</td>
<td>12 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M5=Extended Arm</td>
<td>18 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M4=Arms</td>
<td>24 35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M5=Extended Arms</td>
<td>35 35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M7=Also Trunk</td>
<td>39 136</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M7=Also Trunk</td>
<td>39 136</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M9=Trunk &amp; Feet</td>
<td>79 57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M9=Trunk</td>
<td>79 57</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Moves with Larger Articles

<table>
<thead>
<tr>
<th>Moves with Larger Articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
</tr>
<tr>
<td>M3</td>
</tr>
<tr>
<td>M4</td>
</tr>
<tr>
<td>M5</td>
</tr>
<tr>
<td>M7</td>
</tr>
<tr>
<td>M9</td>
</tr>
</tbody>
</table>

**Command:**

Figure 6. Moves With Larger Articles Display.
Figure 7. Gets for Small Articles Icons.
Figure 8. Gets for Large Articles Icons.
needed (in Mods) for each one. For example, M4G4 represents the code of moving the full arm forward a distance of about 12 inches and grasping an article heavier than two lbs. with both hands. The total number of Mods for this task is 8 Mods which is equal to 1.032 seconds (8 Mods X 0.129 seconds). Layout assisted Gets and Puts use a logic that determines the Mod value of the MOVE code.

Other elements are illustrated in Figures 11, 12, 13, and 14. These represent most of MODAPTS codes that can be used directly by choosing the appropriate icon. Most of these codes such as K1 (little depression) have more than one mod value. For example, K has the following range of Mods: (a) K1, (b) K1.5, (c) K2, (d) K2.5, (e) K3, (f) K3.5, (g) K4, and (h) K4.5. This range will appear on the right of the screen upon selecting the K1 icon. The user can pick any of the K values depending on the operation.

Using the MODAPTS Portion

Before attempting to select any icon, the user must create a new text file or must retrieve an old text file. The computer will prompt the user to choose the drive and the file name. Once a file has been assigned, the user must assign allowances by selecting the allowances function. The computer will prompt the user to input personal allowance, fatigue allowance, and delay allowance for the job.
Figure 9. Puts for Small Articles Icons.
Figure 10. Puts for Large Articles Icons.
Figure 11. Mental Tasks Icons.
Figure 12. Finger/Hand/Wrist Use Icons.
Figure 13. Foot Use Icons.
Figure 14. Body Use Icons.
Allowances can be changed, and the results will reflect that change.

The user can select Gets, Puts, and other MODAPTS codes by pull down menus which in turn activate icon menus that contain graphical representation of the codes. Upon selecting Gets or Puts, "Moves" menus will appear (Figures 5 and 6) to choose an appropriate MOVE command. The computer will activate the Puts or Gets icons (Figures 7, 8, 9, and 10) for either light or heavy articles. The user then selects the appropriate Get or Put icon which is followed by a prompt to input a brief explanation of the task and the frequency of the task. Other prompts such as activity number, load, and other factors are displayed in order to complete the information needed for the analysis.

Once all the relevant information about the task is complete, the computer will use a formula to calculate the Mods (units of time) for that task and will create a text line in a file which is stored. The file can be printed, edited, and appended.

For icons other than Puts or Gets, the user can access codes such as using a Typewriter, Walking, and other activities (Figures 11, 12, 13, and 14). Once the icon is selected, prompts will be displayed to get other information from the user about the task.

MODCAD will create a total of three files for each job. The first file contains the drawing which can be
plotted or edited for later uses. The second file contains all the steps for that job which consists of: (a) activity number, (b) MODAPTS code, (c) brief explanation of the task, (d) the frequency of the task, and (e) the MODS for that task (calculated by the computer). The third file contains the information in the second file along with the following: (a) total Mods; (b) operation name; (c) description of operation; (d) station number; (e) operator name; (f) cycle time in seconds, minutes, and hours with and without rest allowance; (g) cycles/hr; and (h) rest allowance. The third file can be created after the analysis is complete. The purpose of this file is to provide a hard copy of the analysis and may not be used as a data file.

Using Layout Assisted Gets and Puts

The user can obtain an accurate move command prior to selecting Gets or Puts by interacting directly with the graphics area. When Put or Get from layout option is selected, the user will be prompted to digitize two points on the screen that represent the motion of body. Once the path has been digitized, the user will be prompted to enter the mass of the object being manipulated. Based upon the distance, mass, body control, and other factors, the computer will recommend an appropriate move command to be used with Gets and Puts.
EDITING THE FILES

The drawing file can be easily edited by using the graphics portion of MODCAD. The user can choose functions such as erase, copy, and move to manipulate the objects in the drawing.

The text file that contains the tasks can be edited by pulling down the edit menu which will allow the user to edit the file using edlin which resides in the DOS directory. Once corrections have been made, command will return to MODCAD and the user can continue to enter data to the task.
CONCLUSIONS

For many users of PTS the need or benefits may not justify the use of computer or graphics in the analysis of work methods. In those applications where the layout may be an important part of the study a computer aided design program such as MODCAD provides the user with a tool that may be modified to fit his particular operations and allows the analysts to develop consistent standards with a minimum amount of training.

A second benefit of using MODCAD is the ability to modify existing layouts and standards quickly and to transfer these changes directly into an existing data base. This will enable companies to maintain their time standards and process sheets as changes are made rather than allowing them to deteriorate and become obsolete.

MODCAD can be made to handle 3-D drawings and full interface between the graphics option and the analysis. The reason this is not possible in this software are the limitations of the memory and speed of the computer used (286 IBM compatible).

A mainframe computer with a high processing speed can be used instead of a personal computer. AutoCAD can be used on some mainframe computers such as VAX computers. This will allow for more interface between the graphics and
the analysis which is limited now by memory constraints.

MODCAD can be modified for a mainframe application to produce shaded 3-D images and simulated motion of parts and human subjects. This will give the user the tool for highly accurate analysis.
Appendix

MODCAD Computer Code Listing
This routine will be read by AutoCAD automatically. MODCAD Lisp files will be activated by the system.

(defun run ()
  (load "c:/modcad/grids.lsp")
  (load "c:/modcad/mvs.lsp")
  (load "c:/modcad/filer.lsp")
  (load "c:/modcad/reader.lsp")
  (load "c:/modcad/readone.lsp")
  (load "c:/modcad/allow.lsp")
  (load "c:/modcad/old.lsp")
  (load "c:/modcad/new.lsp")
  (load "c:/modcad/display.lsp")
  (load "c:/modcad/fincont.lsp")
  (load "c:/modcad/layout.lsp")
  (load "c:/modcad/hancont.lsp")
  (load "c:/modcad/layout1.lsp")
  (command "menu" "modcad")
  (command "redraw")
  (terpri)
  (prompt "MODCAD IS ACTIVE NOW")
)
(defun c:modcad ()
  (run)
)
This routine will display the results on the screen

(defun display ()
  (setq accum 0)
  (textscr)
    (setq ff (open data "r"))
    (setq z 0)
      (while (<= z counter)
        (setq j (read-line ff))
        ;
        (print z)
        ;
        (print j)
        ;
        (setq one (substr j 2 3))
        (setq two (substr j 4 9))
        (setq three (substr j 13 70))
      )
    (setq accum (+ accum (atof two)))
    ;(print accum)
      (setq z (+ 1 z))
    (setq total "TOTAL MODS : ")
    (setq acc (rtos accum))
    (setq total (strcat total acc))
    (setq gap " ")
    (setq bg " ")
    (setq cy "1 Cycle time in: Secs. Mins. Hrs. 
    Cycle/hr")
    (setq nr "No Rest ")
    (setq wr "W/ Rest ")
    (setq ra "TOTAL REST ALLOWANCES: ")

    (setq total (+ personal fatigue delay))

      (SETQ ANS1 (* ACCUM 0.129))
      (SETQ ANS2 (/ ANS1 60))
      (SETQ ANS3 (/ ANS2 60))
      (SETQ ANS4 (/ 1 ANS3))

      (SETQ ANS5 (* ANS1 (/ TOTAL 100)))
      (SETQ ANS5 (+ ANS5 ANS1))
      (SETQ ANS6 (/ ANS5 60))
      (SETQ ANS7 (/ ANS6 60))
      (SETQ ANS8 (/ 1 ANS7))
    (TERPRI)
    (PROMPT "TOTAL MODS: ")(PRIN1 ACCUM)
    (TERPRI)
    (PROMPT "--------------------------")
    (TERPRI)
(PRINT CY)
(print nr)
(PRIN1 GAP)
(PRIN1 ANS1)
(PRIN1 GAP)
(PRIN1 ANS2)
(PRIN1 GAP)
(PRIN1 ANS3)
(PRIN1 GAP)
(PRIN1 ANS4)

(print wr)
(PRIN1 GAP)
(PRIN1 ANS5)
(PRIN1 GAP)
(PRIN1 ANS6)
(PRIN1 GAP)
(PRIN1 ANS7)
(PRIN1 GAP)
(PRIN1 ANS8)

(TERPRI)
(PROMPT"

(TERPRI)
(PROMPT"

(TERPRI)
(TERPRI)
(TERPRI)
(PRINT "FATIGUE ALLOWANCE: ")
(PRIN1 FATIGUE)

(PRINT "PERSONAL ALLOWANCE: ")
(PRIN1 PERSONAL)

(PRINT "DELAY ALLOWANCE: ")
(PRIN1 DELAY)
(terpri)
(prompt "___________________________")

(print ra) (PRIN1 TOTAL)

(close ff)
(TERPRI)
(terpri)
(prompt "
--------------------------")
(terpri)
(prompt "Press F1 to return to graphic
screen")
(terpri)
(prompt "
--------------------------------------")
(terpri)

(defun c:disp ()
  (display)
)
^Z^Z
This routine is for hand controlled tasks

(DEFUN han ()

(SETQ M (GETREAL "Please enter mass in Lbs.: "))

(setq res (getstring "Is the person man who is suitable for the task? (Y or N) : ")
(setq res (strcase res))
(if (= res "N")
   (setq M (* M 2))
)

(setq res (getstring "Are you sliding an article along a surface? (Y or N) : ")
(setq res (strcase res))
(if (= res "Y")
   (setq M (/ M 3))
)

;;(print m)
(if (and (> m 0) (<= m 35)) (Prompt "Mass is too low for a loadfactor"))

(if (and (> m 35) (<= m 44)) (setq code "L1")
(if (and (> m 35) (<= m 44)) (setq mods 1))

(if (and (> m 44) (<= m 53)) (setq code "L2")
(if (and (> m 44) (<= m 53)) (setq mods 2))
;;;;
(if (and (> m 53) (<= m 62)) (setq code "L3")
(if (and (> m 53) (<= m 62)) (setq mods 3))

(if (and (> m 62) (<= m 71)) (setq code "L4")
(if (and (> m 62) (<= m 71)) (setq mods 4))

(if (and (> m 71) (<= m 79)) (setq code "L5")
(if (and (> m 71) (<= m 79)) (setq mods 5))

(if (and (> m 79) (<= m 88)) (setq code "L6")
(if (and (> m 79) (<= m 88)) (setq mods 6))

(if (and (> m 88) (<= m 97)) (setq code "L7")
(if (and (> m 88) (<= m 97)) (setq mods 7))

(if (and (> m 97) (<= m 106)) (setq code "L8")
(if (and (> m 97) (<= m 106)) (setq mods 8))
(if (and (> m 106)(<= m 115))(setq code "L9"))
(if (and (> m 106)(<= m 115))(setq mods 9))
(if (and (> m 115)(<= m 123))(setq code "L10"))
(if (and (> m 115)(<= m 123))(setq mods 10))
(if (and (> m 123)(<= m 132))(setq code "L11"))
(if (and (> m 123)(<= m 132))(setq mods 11))
(if (and (> m 132)(<= m 97))(setq code "L12"))
(if (and (> m 88)(<= m 97))(setq mods 12))

(if (> m 132)(Prompt "Mass is too heavy for human operator
..... "))
(terpri)
(prompt "Select DONE to confirm or CANCEL to abort
...........")

; (print code)
; (print mods)

(defun c:hancont ()
  (han)
)

*Z
This routine will install MODCAD on the hard drive

```
echo off
if exist acad.exe goto next
echo Please change directory to (ACAD10) where acad.exe resides
echo and try again  (A:install)
goto end

:next
echo Copying MODCAD Menu and Slide Libraries into AutoCAD directory
copy a:modcad.mnu
copy a:*.sib
copy a:acad.lsp
cd\
copy a:modcad.bat
echo Making \modcad directory
md modcad
cd modcad
echo Copying files into \MODCAD directory
copy a:.*
echo off
echo MODCAD has been installed successfully!
```

:end

`^Z`^Z`^Z`^Z`
This routine will activate MODCAD from the c: prompt

```plaintext
echo off
cls
echo Initializing system and activating AutoCAD
echo off
set path=c:\;c:\acad10;c:\modcad
set lispheap=40000
set lispstack=5000
set acadfreeram=24
set acad=\modcad
acad
cd\
^Z^Z^Z
```
This routine will create a new text file for analysis

(defun newfile ()
  (setq step 0)
  (setq order 0)
  (setq counter 0)
  (prompt "This option will create a new file")
  (setq drive (getstring "\nEnter drive for data file : ")
  (setq name (getstring "\nEnter file name + extention : ")
  (setq col ":")
  (setq data (strcat drive col name))
  (list data)
)
(defun c:new ()
  (newfile)
)
^Z^Z
This routine will read all the steps and will display them on the screen.

(defun reader ()
  (setq accum 0)
  (textscr)
    (setq ff (open data "r"))
    (setq z 0)
    (while (<= z counter)
      (setq j (read-line ff))
        (print z)
        (print j)
      (setq one (substr j 2 3))
      (setq two (substr j 4 9))
      (setq three (substr j 13 70))
    (setq output (strcat one three))
    (setq output (strcat output two))
    (print output)
      (setq z (+ 1 z)))
  (close ff)
  (terpri)
  (prompt "--------------------------------------")
  (terpri)
  (prompt "Press F1 to return to graphic screen")
  (terpri)
  (prompt "--------------------------------------")
  (terpri)
)
(defun c:read ()
  (reader)
)
This routine will set the allowances required for analysis.

(defun all ()
  (prompt "Default value for personal allowance = 5%")
  (setq personal (getreal "\nInput personal allowance % : "))
  (if (= personal nil) (setq personal 5))

  (prompt "Default value for fatigue allowance = 5%")
  (setq fatigue (getreal "\nInput fatigue allowance % : "))
  (if (= fatigue nil) (setq fatigue 5))

  (prompt "Default value for delay allowance = 5%")
  (setq delay (getreal "\nInput delay allowance % : "))
  (if (= delay nil) (setq delay 5))
)
(defun c:allow ()
  (all)
)
This routine will create the steps for MODAPTS routine that do not involve Puts and Gets.

(defun filer ()
  (setq counter (+ step 1))
  (setq talk "STARTING THE NEW STEP NUMBER :")
  (setq TR (list talk counter))
  (print TR)

  ; gettin the description of the task
  (setq desc (getstring T 5 "\nDescription [40 letters] : "))

  ; increasing the step count by one
  (setq step (+ step 1))

  ; getting the string out of the step integer to be displayed
  (setq temp step)
  (setq xstep (rtos temp 2 0))

  ; getting the frequency
  (setq freq (getint "\nPlease enter frequency: "))

  ; getting the frequency * mods result for that step
  (setq prod (* freq mods))

  ; preparing the data to be in STRING form with separations
  (setq space " ")
  (setq prod (rtos prod 2 2))
  (setq prod (strcat space prod))
  (setq prod (strcat prod space))
  (setq prod (strcat prod space))
  (setq desc (strcat desc space))
  (setq freq (rtos freq 2 0))
  (setq desc (strcat desc freq))
  (setq desc (strcat desc space))
  (setq desc (strcat desc code desc))
  (setq desc (strcat prod desc))
  (setq desc (strcat xstep desc))

  ; opening the data file
  (setq f (open data "a"))

  ; writing the string into the open file
  (print desc f)
  ; closing the open file
(close f)
)

; automatic program function
(defun c:file ()
(filer)
)
^Z^Z
This routine will construct a grid.

(defun gr ()
  (command "layer" "n" "gridline" "")
  (command "layer" "s" "gridline" "")
  (setq h (getreal "$\text{Please enter width of layout: }$"))
  (setq w (getreal "$\text{Please enter height of layout: }$"))
  (setq ord (fix h))
  (setq obs (fix w))
  (setq pt (list ord obs))
  (command "limits" "" pt)
  (command "zoom" "a")
  (setq e 0)
  (while (<= e obs)
    (setq west (list 0 e))
    (setq east (list ord e))
    (command "line" west east "")
    (setq e (+ e 1)))
  (setq z 0)
  (while (<= z ord)
    (setq south (list z 0))
    (setq north (list z obs))
    (command "line" north south "")
    (setq z (+ z 1)))
  (command "layer" "s" "0" "")
)
(defun c:grids ()
  (gr))
This routine will extract information for layout assisted MOVES

(defun lay ()
  (setq d (getdist "\nDigitize first and second point of MOVE or enter a distance: ")
  (print d)
  (setq w (getreal "\nEnter weight of articles 2 lbs. and under: "))
  (print w)
  (if (and (> d 0)(<= d 1))(setq r 1))
  (if (and (> d 1)(<= d 2))(setq r 2))
  (if (and (> d 2)(<= d 6))(setq r 3))
  (if (and (> d 6)(<= d 12))(setq r 4))
  (if (and (> d 12)(<= d 18))(setq r 5))
  (if (and (> d 18)(<= d 30))(setq r 6))
  (if (> d 30)(setq r 7))
  (print r)
  (if (= 1 r)(setq co "M1")(setq mo 1))
  (if (= 2 r)(setq co "M2")(setq mo 1))
  (if (= 3 r)(setq co "M3")(setq mo 1))
  (if (= 4 r)(setq co "M4")(setq mo 1))
  (if (= 5 r)(setq co "M5")(setq mo 1))
  (if (= 6 r)(setq co "M7")(setq mo 1))
  (if (= 7 r)(setq co "M1")(setq mo 1))

(PROMPT "THE RECOMMENDED MOVE CODE FOR THIS ARTICLE IS: ")
(PRIN1 CO)
(TERPRI)
(PROMPT "SELECT DONE TO CONFIRM OR CANCEL TO ABORT
...............")
)

(defun c:layout ()
  (lay)
)
\^Z\^Z

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
This routine will create steps that involve Puts and Gets
(defun moves ()
  (setq counter (+ step 1))
  (setq talk "STARTING THE NEW STEP NUMBER :")
  (setq TR (list talk counter))
  (print TR)

; getting the description of the task
  (setq desc (getstring T 5 "\nDescription [40 letters] : "))

; Extracting input from the MOVE-PUT and MOVE-GET statements
  (setq code (strcat code))
  (setq mods (+ mods))

; increasing the step count by one
  (setq step (+ step 1))

; getting the string out of the step integer to be displayed
  (setq temp step)
  (setq xstep (rtos temp 2 0))

; getting the frequency
  (setq freq (getint "\nPlease enter frequency: "))

; getting the frequency * mods result for that step
  (setq prod (* freq mods))

; preparing the data to be in STRING form with separations
  (setq space " ")
  (setq prod (rtos prod 2 2))
  (setq prod (strcat space prod))
  (setq prod (strcat prod space))
  (setq prod (strcat prod space))
  (setq desc (strcat desc space))
  (setq freq (rtos freq 2 0))
  (setq desc (strcat desc freq))
  (setq desc (strcat space desc))
  (setq desc (strcat code desc))
  (setq desc (strcat prod desc))
  (setq desc (strcat xstep desc))

; opening the data file
  (setq f (open data "a"))

; writing the string into the open file
  (print desc f)
; closing the open file
(close f)
)

; automatic program function
(defun c: MV ()
(moves)
)
^Z
This routine will read an old file for editing.

(defun oldfile ()
  (setq step 0)
  (setq order 0)
  (setq counter 0)
  (prompt "This option will retrieve an old file")
  (setq drive (getstring "\nEnter drive for data file :  "))
  (setq name (getstring "\nEnter file name + extention :  "))
  (setq col ":")
  (setq data (strcat drive col name))
  (list data)
)
(defun c:old ()
  (oldfile)
)
^Z
This routine will read and display one step

(defun single ()
;;;;textscr
(setq ord (getint "Please enter step number to display : "))

(setq ff (open data "r"))

; corrector
(setq order (+ ord 1))
(setq z 1)
; the while loop begins
(while (<= z order)
(setq j (read-line ff))
(setq z (+ Z 1))
)
; The while loop ends

; reading the last file after the count
;;
(list j)
(setq one (substr j 2 3))
(setq two (substr j 4 9))
(setq three (substr j 13 70))
(setq output (strcat one three))
(setq output (strcat output two))
(list output)

)
; main program terminates

(defun c:readone ()
(singe)
)
^Z^Z
This routine will produce a hard copy of MODAPTS analysis

(defun pap ()
  (prompt "This option will create an output file")

  ; getting the output file drive
  (setq drive (getstring "\nEnter drive for output file: "))
  ; getting the output file name
  (setq name (getstring "\nEnter name+ext for output file: "))
  (setq col ";")
  (setq TLR (open output "w"))
  ; getting the name of the operation
  (setq opname (getstring T 5 "\nOperation name [40 letters]: "))
  ; getting the description of operation.
  (setq opdesc (getstring T 5 "\nDescription [40 letters]: "))
  ; getting the station number.
  (setq station (getstring T 5 "\nStation number [40 letters]: "))
  ; getting the operator name.
  (setq operator (getstring T 5 "\nOperator name [40 letters]: "))

  ; ******************
  (setq accum 0)
  (textscr)
    (setq z 0)
      (while (<= z counter)
        (setq j (read-line TLR))
        (setq one (substr j 2 3))
        (setq two (substr j 4 9))
        (setq three substr j 13 70))

      (setq output (strcat one three))
    (setq output (strcat output two))
    ; writing the string into the open file
    (print desc TLR)
      (setq z (+ z 1))
  ; ******************
  (setq accun 0)

    (setq z 0)
      (while (<= j (read-line TLR))
        (setq one substr j 2 3))
      (setq two (substr j 4 9))
(setq three substr j 13 70))
(setq accum (+ accum (atof two)))
(setq z (+ z 1)))

(setq "TOTAL MODS : ")
(setq acc (rtos accum))
(setq total (strcat total acc))
(setq gap " ")
(setq bg " ")

(setq cy "1 Cycle time in:  Sec  Min  Hrs  
Cycle/hr")
(setq nr "NO REST ")
(setq wr "W/ REST ")
(setq ra "Total rest allowances: ")

(setq total (+ personal fatigue delay))

(setq ANS1 (* accum 0.129))
(setq ans2 (/ans1 60))
(setq ans3 (/ans2 60))
(setq ans4 (1 ans3))

(setq ans5 (* ans1 (/total 100)))
(setq ans5 (+ ans5 ans1))
(setq ans6 (/ ans5 60))
(setq ans7 (/ ans6 60))
(setq ans8 (/ 1 ans7))

(print "TOTAL MODS: :TLR)(Printl accum TLR)

(print "---------------------------"TLR)

(print cy TLR)
(print nr TLR)
(print1 gap TLR)
(print1 ans1 TLR)
(print1 ans2 TLR)
(print1 gap TLR)
(print1 ans3 TLR)
(print1 gap TLR)
(print1 ans4 TLR)

(print wr TLR)
(print1 gap TLR)
(print1 ANS5 TLR)
(print1 gap TLR)
(print1 ans6 TLR)
(print1 gap TLR)
(print1 ans7 TLR)
(print1 gap TLR)
(print ans8 TLR)

(print "Fatigue Allowance: " TLR)
(print fatigue TLR)

(print "Personal Allowance: " TLR)
(print personal TLR)

(print "Delay Allowance: " TLR)
(print delay TLR)
(print ra TLR) (print total TLR)
(close TLR)

(defun c:PAPER ()
  (PAP)
)
This is a listing for the MODCAD custom menu.

***BUTTONS
;
$P1=* ^C^C
^B

***pop1
[TOOLS]
CENTER
ENDPOINT
[INTERsection]INT
MIDPOINT
NEAREST
[PERpendicular]PER
TANgent

***POP2
[OBJECTS]
[INSERT A BLOCK]^C^Cinsert;
[OBJECTS WITH ATTRIBUTES]^C^Cinsert;
[-------------]
[CREATE A BLOCK]^C^CblocK;

***pop3
[GETS]
[SMALL ARTICLES]$S=GTS VSLIDE C:/MODCAD/MOVES
[-------------]
[LARGE ARTICLES]$S=GTL VSLIDE C:/MODCAD/MOVEL
[-------------]
[FROM LAYOUT]$S=LGET

***pop4
[PUTS]
[SMALL ARTICLES]$S=PTS VSLIDE C:/MODCAD/MOVES
[-------------]
[LARGE ARTICLES]$S=PTL VSLIDE C:/MODCAD/MOVEL
[-------------]
[FROM LAYOUT]$S=LPUT
***pop5
[OTHER ELEMENTS]
[MENTAL TASK]$I=MENTAL $I=* 
[HAND/FINGER USE]$I=HANDUSE $I=* 
[FOOT USE]$I=FOOTUSE $I=* 
[BODY USE]$I=BODYUSE $I=* 
[-------------]
[WEIGHT FACTOR]$C^C$P(setq code "L1") (setq mods 1) file;  
[EXTRA FORCE]$C^C$P(setq code "X4") (setq mods 4) file; 
[-------------]
[ALLOWANCES]allow;

***pop6
[TEXT FILE]
[START A NEW FILE]new; 
[APPEND AN EXISTING FILE]old; 
[------------]
[LIST A SINGLE STEP]readone; 
[LIST ALL STEPS]read; 
[-------------]
[EDIT DATA FILE WITH EDLIN]EDIT; 
[------------]
[DISPLAY RESULTS]disp; 
[-------------]
[CREATE AN OUTPUT FILE]PAPER;

***POP7
[DRAWING FILE]
SAVE 
[PRINT]PRPLOT

***ICON
**GSMALL 
[GETS FOR SMALL ARTICLES] 
[GET(G0)]$C$CREDRAW (setq de "G0") (setq ds 0) MV; $s=root 
[GET(G1)]$C$CREDRAW (setq de "G1") (setq ds 1) MV; $S=ROOT 
[GET(G3)]$C$CREDRAW (setq de "G3") (setq ds 3) MV; $S=ROOT

**GLARGE 
[GETS FOR LARGE ARTICLES] 
[GET1(G2)]$C$CREDRAW (setq de "G2") (setq ds 2) MV; $s=root
**P SMALL**

[PUTS FOR SMALL ARTICLES]

[PUT(P0)]**P**^C CREDRAW (setq de "P0") (setq ds 0) MV; $S=ROOT

[PUT(P2)]**P2**^C CREDRAW (setq de "P2") (setq ds 2) MV; $S=ROOT

[PUT(P5)]**P5**^C CREDRAW (setq de "P5") (setq ds 5) MV; $S=ROOT

**PLARGE**

[PUTS FOR LARGE ARTICLES]

[PUT1(P01)]**P01**^C CREDRAW (setq de "P0") (setq ds 0) MV; $S=ROOT

[PUT1(P21)]**P21**^C CREDRAW (setq de "P2") (setq ds 2) MV; $S=ROOT

[PUT1(P51)]**P51**^C CREDRAW (setq de "P5") (setq ds 5) MV; $S=ROOT

[PUT1(P10)]**P10**^C CREDRAW (setq de "P10") (setq ds 10) MV; $S=ROOT

**MENTAL**

[MENTAL TASKS]

[MENTAL(D3)]**D3**^C$S=XD3 $S=*

[MENTAL(E2)]**E2**^C$P(setq code "E2") (setq mods 2) file;

[MENTAL(R3)]**R3**^C$P(setq code "R3") (setq mods 3) file;

[MENTAL(E4)]**E4**^C$P(setq code "E4") (setq mods 4) file;

[MENTAL(Q6)]**Q6**^C$S=XQ6 $S=*

[MENTAL(A5)]**A5**^C$S=XA5 $S=*

[MENTAL(H4)]**H4**^C$S=XH4 $S=*

**HANDUSE**

[FINGER/HAND/WRIST USE]

[HAND(K1)]$S=XK1 $S=*

[HAND(T18)]$S=XT18 $S=*

[HAND(J2)]^C$P(setq code "J2") (setq mods 2) file;

[HAND(C4)]$S=XC4 $S=*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**FOOTUSE
[FOOT USE]
[FOOT(F3)]C^P(setq code "F3") (setq mods 3) file;
[FOOT(W5)]$S=XW5 $S=* 

**BODYUSE
[BODY USE]
[BODY(B17)]$S=XB17 $S=* 
[BODY(S30)]$S=XS30 $S=* 

***SCREEN
**ROOT
[MODCAD]
[*****]$S=aid 
[ ] [DIM]$s=specs $s=*
[ ] [TEXT]$s=tex $s=*
[ ] [DRAW]$s=pr $s=*
[ ] [GRID]$s=gr $s=*
[ ] [ZOOM]^C^ZOOM$s=ZM $s=*
[ ] [COLOR]^C^Cvslide;c:/modcad/colors;COLOR;$S=COL $S=*
[ ] [L-TYPE]^C^CLINETYPE;$S=LT $S=*
[ ] [EDIT]$S=CORRECT $S=*
[----------]
[ ] [AutoCAD]^C^Cmenu;acad; 

**LT
[LINE]
[ MENU]
[ _____]
[ contin.]CONTINUOUS;
[ dashed]dashed;
[ hidden]hidden;
[ center]center;
[ phantom]phantom;
**ZM**

[ZOOM:] zoom

[All]

[Center]

Dynamic

Extents

Left

Previous

Window

[OFF]^C^layer;freez;gridline;;

[ON]^C^layer;thaw;gridline;;

[NEW]^C^GRIDDS;
MODCAD] $s=root $s=* 

**GTS
[ MOVES]
[ FOR]
[ A]
[ SMALL]
[ ARTICLE]
[--------]
[]
[M1](setq co "M1") (setq mo 1) $I=GSMALL $I=* []
[M2](setq co "M2") (setq mo 2) $I=GSMALL $I=* []
[M3](setq co "M3") (setq mo 3) $I=GSMALL $I=* []
[M4](setq co "M4") (setq mo 4) $I=GSMALL $I=* []
[M5](setq co "M5") (setq mo 5) $I=GSMALL $I=* []
[M7](setq co "M7") (setq mo 7) $I=GSMALL $I=* []
[CANCEL] $S=root $S=* 

**GTL
[ MOVES]
[ FOR]
[ A]
[ LARGE]
[ ARTICLE]
[--------]
[]
[M2](setq co "M2") (setq mo 2) $I=GLARGE $I=* []
[M3](setq co "M3") (setq mo 3) $I=GLARGE $I=* []
[M4](setq co "M4") (setq mo 4) $I=GLARGE $I=* []
[M5](setq co "M5") (setq mo 5) $I=GLARGE $I=* 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**PTS**

[ MOVES]
[ FOR]
[ A]
[ SMALL]
[ ARTICLE]
[--------]

[ ]

[M1](setq co "M1") (setq mo 1) $I=PSMALL $I=* 

[M2](setq co "M2") (setq mo 2) $I=PSMALL $I=* 

[M3](setq co "M3") (setq mo 3) $I=PSMALL $I=* 

[M4](setq co "M4") (setq mo 4) $I=PSMALL $I=* 

[M5](setq co "M5") (setq mo 5) $I=PSMALL $I=* 

[M7](setq co "M7") (setq mo 7) $I=PSMALL $I=* 

[CANCEL]$s=root $s=* 

**PTL**

[ MOVES]
[ FOR]
[ A]
[ LARGE]
[ ARTICLE]
[--------]

[ ]

[M2](setq co "M2") (setq mo 2) $I=PLARGE $I=* 

[M3](setq co "M3") (setq mo 3) $I=PLARGE $I=* 

[M4](setq co "M4") (setq mo 4) $I=PLARGE $I=* 

[M5](setq co "M5") (setq mo 5) $I=PLARGE $I=* 

[CANCEL]$s=root $s=*
**specs
[ DIMENSIN ]
[  MENU ]
[-------]
[ DIM ] ^C^CDIM
[ ]
[ ]
HORIZ
VERTICAL
ALIGNED
ROTATED
[-------]
RADIUS
DIAMETER
[-------]
LEADER
[ ]
[ ]
[MODCAD]$S=ROOT $S=* 

**CORRECT
[ EDITING ]
[  MENU ]
[ ]
[ ]
[ ]
[ ]
ERASE
MOVE
COPY
ROTATE
ARRAY
[PROPERTY] CHANGE
FILLET
MIRROR
TRIM
EXTEND
[ ]
[ ]
[------]
[MODCAD]$S=ROOT $S=* 
[------]

**XG12
[GETS]
[G12]^C^CREDRAW (setq de "G12") (setq ds 12) MV; $S=ROOT

[G16]^C^CREDRAW (setq de "G16") (setq ds 16) MV; $S=ROOT

[CANCEL]^C^C$S=ROOT

**XK1
[ KEY]
[KEYBOARD]
[-------]
[K1]^C^C^P(setq code "K1") (setq mods 1) file;$S=ROOT

[K1.5]^C^C^P(setq code "K1.5") (setq mods 1.5) file;$S=ROOT

[K2]^C^C^P(setq code "K2") (setq mods 2) file;$S=ROOT

[K2.5]^C^C^P(setq code "K2.5") (setq mods 2.5) file;$S=ROOT

[K3]^C^C^P(setq code "K3") (setq mods 3) file;$S=ROOT

[K3.5]^C^C^P(setq code "K3.5") (setq mods 3.5) file;$S=ROOT

[K4]^C^C^P(setq code "K4") (setq mods 4) file;$S=ROOT

[K4.5]^C^C^P(setq code "K4.5") (setq mods 4.5) file;$S=ROOT

[CANCEL]^C^C$S=ROOT
**XD3

[]

[DECIDE]
[D3]^C^C^P(setq code "D3") (setq mods 3) file; $S=ROOT
[]

[DETERM.]
[NUMBER]
[N3]^C^C^P(setq code "N3") (setq mods 3) file; $S=ROOT
[N6]^C^C^P(setq code "N6") (setq mods 6) file; $S=ROOT
[]

[VOCALISE]
[V3]^C^C^P(setq code "V3") (setq mods 3) file; $S=ROOT
[]

[CANCEL]^C^C$S=ROOT

**XH4

[HAND-]
[WRITE]
[]

[H4]^C^C^P(setq code "H4") (setq mods 4) file; $S=ROOT
[]
[H5]^C^C^P(setq code "H5") (setq mods 5) file; $S=ROOT
[]
[H6]^C^C^P(setq code "H6") (setq mods 6) file; $S=ROOT
[]
[H7]^C^C^P(setq code "H7") (setq mods 7) file; $S=ROOT
[]
[H12]^C^C^P(setq code "H12") (setq mods 12) file; $S=ROOT
[]
[H26]^C^C^P(setq code "H26") (setq mods 26) file; $S=ROOT
[]
[H35]^C^C^P(setq code "H35") (setq mods 35) file; $S=ROOT
[]

[CANCEL]^C^C$S=ROOT

**XA5

[ARITH-]
[METIC]
[
[
[
[
[
[
[[A5]^C^P(setq code "A5") (setq mods 5) file; $S=ROOT]
[
[A18]^C^P(setq code "A18") (setq mods 18) file; $S=ROOT]
[
[A24]^C^P(setq code "A24") (setq mods 24) file; $S=ROOT]
[
[
[
[
[
[CANCEL]^C^S=ROOT

**XT18**
[TYPE-]
[WRITE]
[
[
[
[
[
[[T13]^C^P(setq code "T13") (setq mods 13) file; $S=ROOT]
[
[T18]^C^P(setq code "T18") (setq mods 18) file; $S=ROOT]
[
[T24]^C^P(setq code "T24") (setq mods 24) file; $S=ROOT]
[
[T36]^C^P(setq code "T36") (setq mods 36) file; $S=ROOT]
[
[T40]^C^P(setq code "T40") (setq mods 40) file; $S=ROOT]
[
[T60]^C^P(setq code "T60") (setq mods 60) file; $S=ROOT]
[
[
[
[
[
[CANCEL]^C^S=ROOT

**XQ6**
[QUEST]
[
[
[
[
[
[[Q6]^C^P(setq code "Q6") (setq mods 6) file; $S=ROOT]
[
]
[Q9]`C^C^P(setq code "Q9") (setq mods 9) file; $S=ROOT

[Q10]`C^C^P(setq code "Q10") (setq mods 10) file; $S=ROOT

[Q12]`C^C^P(setq code "Q12") (setq mods 12) file; $S=ROOT

[Q15]`C^C^P(setq code "Q15") (setq mods 15) file; $S=ROOT

[Q18]`C^C^P(setq code "Q18") (setq mods 18) file; $S=ROOT

[CANCEL]`C^C^S=$S=ROOT

**XC4

[CRANK]

[C3]`C^C^P(setq code "C3") (setq mods 3) file; $S=ROOT

[C4]`C^C^P(setq code "C4") (setq mods 4) file; $S=ROOT

[CANCEL]`C^C^S=$S=ROOT

**WX5

[WALK]

[W1.83]`C^C^P(setq code "W1.83") (setq mods 1.83) file; $S=ROOT

[W2.36]`C^C^P(setq code "W2.36") (setq mods 2.36) file; $S=ROOT

[W4.5]`C^C^P(setq code "W4.5") (setq mods 4.5) file; $S=ROOT

[W5]`C^C^P(setq code "W5") (setq mods 5) file; $S=ROOT

[W6]`C^C^P(setq code "W6") (setq mods 6) file; $S=ROOT

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
[W7.75]^C^CP(setq code "W7.75") (setq mods 7.75) file;$S=ROOT

[CANCEL]^C^CS=ROOT

**XB17
[BEND &]
[ARISE]

[B17]^C^CP(setq code "B17") (setq mods 17) file;$S=ROOT

[B18]^C^CP(setq code "B18") (setq mods 18) file;$S=ROOT

[CANCEL]^C^CS=ROOT

**XS30
[SIT &]
[STAND]

[S30]^C^CP(setq code "S30") (setq mods 30) file;$S=ROOT

[S36]^C^CP(setq code "S36") (setq mods 36) file;$S=ROOT

[S48]^C^CP(setq code "S48") (setq mods 48) file;$S=ROOT

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**XL1**

[LOAD]

[FACTOR]

[FACTOR]

[FINGER]^C^CFINCONT;

[CONTROL]^C^CFINCONT;

[HAND]^C^CANCONT;

[CONTROL]^C^CANCONT;

[CANCEL]^C^CSS=ROOT

***LGET

[ LAYOUT]

[ASSISTED]

[ GETS]

[SMALL]^C^CLAYOUT;

[ARTICLES]^C^CLAYOUT;

[DONE]$I=GSMALL $I=* 

[--------]

[LARGER]^C^CLAYOUT1;

[ARTICLES]^C^CLAYOUT1;

[DONE]$I=GLARGE $I=* 

[CANCEL]$S=ROOT

***LPUT

[ LAYOUT]
[ASSISTED]
[  PUTS]
[--------]

[ SMALL] ^C^CLAYOUT;
[ ARTICLES] ^C^CLAYOUT;
[ ]
[ DONE] $I=PSMALL $I=* 

[--------]

[ LARGER] ^C^CLAYOUT1;
[ ARTICLES] ^C^CLAYOUT1;
[ ]
[ DONE] $I=PLARGE $I=* 
[ ]
[ ]
[ ]
[ ]
[ CANCEL] $S=ROOT

^Z
BIBLIOGRAPHY


