

# **ADAPT User's Manual: A Data Analysis Tool for Human Performance Evaluation in Dynamic Systems**

**Farzad S. Khan, Elfreda Lau,  
Xinyao Yu, Kim J. Vicente, & Michael W. Carter**

**CEL 97-03**





Director: Kim J. Vicente, B.A.Sc., M.S., Ph.D.

The Cognitive Engineering Laboratory (CEL) at the University of Toronto (U of T) is located in the Department of Mechanical & Industrial Engineering, and is one of three laboratories that comprise the U of T Human Factors Research Group. CEL began in 1992 and is primarily concerned with conducting basic and applied research on how to introduce information technology into complex work environments, with a particular emphasis on power plant control rooms. Professor Vicente's areas of expertise include advanced interface design principles, the study of expertise, and cognitive work analysis. Thus, the general mission of CEL is to conduct principled investigations of the impact of information technology on human work so as to develop research findings that are both relevant and useful to industries in which such issues arise.

### **Current CEL Research Topics**

CEL has been funded by Atomic Energy Control Board of Canada, AECL Research, Alias|Wavefront, Asea Brown Boveri Corporate Research - Heidelberg, Defense and Civil Institute for Environmental Medicine, Honeywell Technology Center, Japan Atomic Energy Research Institute, Natural Sciences and Engineering Research Council of Canada, Rotoflex International, and Westinghouse Science & Technology Center. CEL also has collaborations and close contacts with the Mitsubishi Heavy Industries and Toshiba Nuclear Energy Laboratory. Recent CEL projects include:

- Studying the interaction between interface design and adaptation in process control systems.
- Understanding control strategy differences between people of various levels of expertise within the context of process control systems.
- Developing safer and more efficient interfaces for computer-based medical devices.
- Designing novel computer interfaces to display the status of aircraft engineering systems.
- Developing and evaluating advanced user interfaces (in particular, transparent UI tools) for 3-D modelling, animation and painting systems.

### **CEL Technical Reports**

For more information about CEL, CEL technical reports, or graduate school at the University of Toronto, please contact Dr. Kim J. Vicente at the address printed on the front of this technical report.

## **Abstract**

The purpose of this manual is to describe the ADAPT (Adaptation Data Analysis & Processing Tool) software package. This research tool can be used to assess the behaviour and strategy development of people controlling a dynamic process simulation. This report is a user's manual for ADAPT. It provides with a brief summary of the development of ADAPT, a description of the ADAPT software architecture, and a detailed description of the data analysis modules that comprise ADAPT. The modules or functions are defined by a description of the analysis measure involved, the format required for data input, the mathematical calculations used, and an example of the results. Also included in this manual are installation and data input procedures.



Graphs of Data Analysis Modules	...	...	...	...	31
Variable Names and Descriptions	...	...	...	...	42

## Preface

This report is a user's manual for the Adaptation Data Analysis & Processing Tool (ADAPT). ADAPT is a software package that was developed to analyze data from DURESS (DUal REservoir Simulation System) II - a microworld for cognitive engineering research and teaching (Vicente, 1991; Orchanian, Smahel, Howie, & Vicente 1996). This report describes the data recorded from the operation of DURESS II, the architecture of ADAPT, the different measures included in ADAPT, and the functions that support the calculation of these measures. Although the implementation described here is specific to DURESS II, we believe that ADAPT could be productively generalized to evaluate human performance in a wide variety of dynamic systems.

DURESS II is an updated version of DURESS, a thermal-hydraulic process simulation that has served as a research vehicle for a number of studies on advanced interface design for process control systems (see Vicente, 1997 for a review). The physical structure of DURESS II is illustrated in Figure 1. The system consists of two redundant feed water streams that can be configured to supply water to two reservoirs. The goals are to keep each of the reservoirs at a prescribed temperature (e.g. 40°C and 20°C), and to maintain enough water in each reservoir to satisfy each of the current demand flow rates, which are externally determined. To satisfy these system goals, the operator can act on: eight valves (VA, VA1, VA2, VB, VB1, VB2, VO1, VO2), two pumps (PA, PB), and two heaters (HTR1, HTR2). DURESS II was modeled to be consistent with the laws of physics (e.g., the conservation laws), although several simplifying assumptions were made. It is assumed that the reader of this report is familiar with the contents of the Users' Manual for the DURESS II microworld (Orchanian et al., 1996).

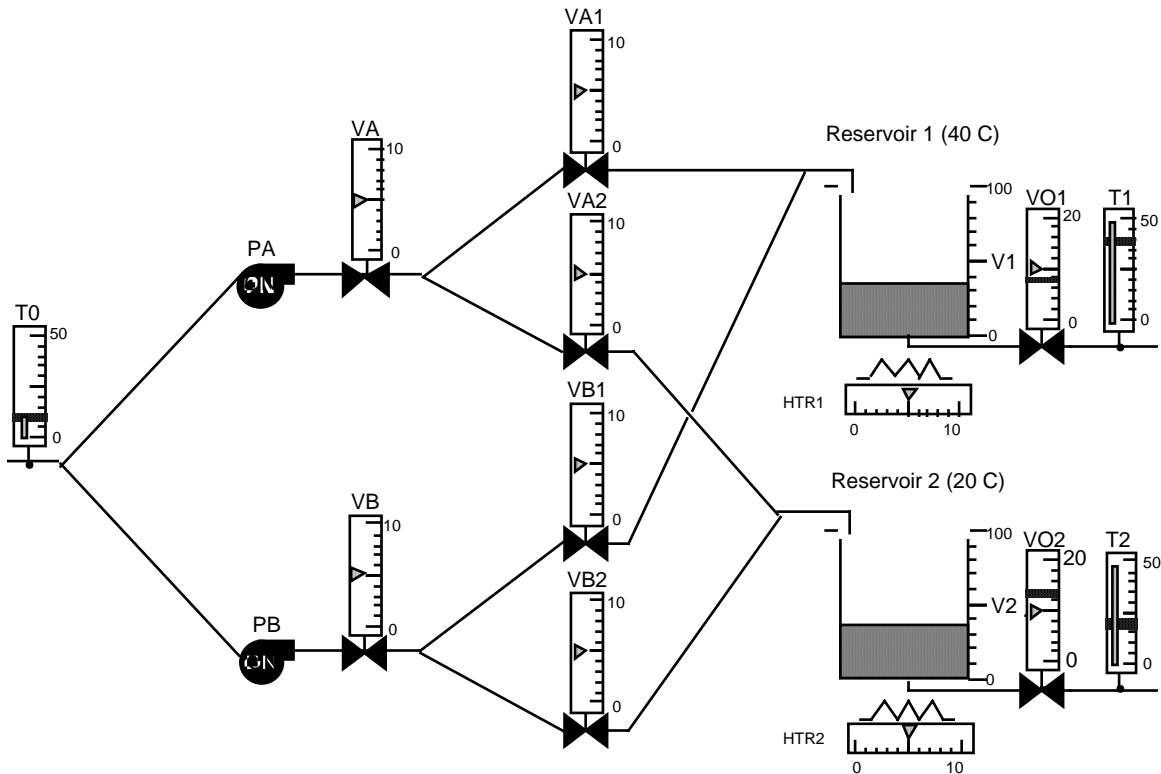


Figure 1: The Physical Structure of DURESS II.

### How to Install ADAPT

ADAPT is designed with the MATLAB programming language. The only software requirement to run ADAPT is that the user's computer must have a copy of MATLAB (a student version is sufficient).

The installation of ADAPT is very simple once the MATLAB Program is installed. All the packages in ADAPT are compressed into a file called ADAPT.tar.gz.

- First, copy this file to a destination directory. For example, copy to directory:

```
/duress/jaeri4/AnalysisTools/
```

- Second, uncompress the files using "gunzip" and "tar". For example:

```
gunzip ADAPT.tar.gz
```

```
tar -xf ADAPT.tar
```

Now, you are ready to run ADAPT if you have DURESS II data files available. For example, the data files for the JAERI 4 project are in the following directories:

```
/duress/jaeri4/JAERI4data/state (this contains the data of state variables of each trial)
```

/duress/jaeri4/JAERI4data/action (this contains the data of the action variables)

### How to Run ADAPT

It is advised you create a directory to save your analysis results. First, go to your own directory and type "matlab" in the UNIX or DOS prompt to enter MATLAB environment. Then add the paths for ADAPT and for the data files by executing the following commands in the MATLAB prompt:

```
path(path, "/duress/jaeri4/AnalysisTools");
path(path, "/duress/jaeri4/JAERI4data/state");
path(path, "/duress/jaeri4/JAERI4data/action");
```

Now, you are ready to run ADAPT by typing the command: "adapt".

ADAPT is user-interactive and menu-driven:

1. Choose the name of the subject whose data files are to be analyzed. For the JAERI 4 project, there are six choices: AS, AV, IS, TL, ML, and WL.
2. Choose one of the trial periods to be analyzed. There are three choices: Start Up, Tune, and Shut Down.
3. Choose the measure that needs to be analyzed. There are 22 different measures available (see below).
4. Input the number of trials or the number of blocks to be analyzed. If you enter number of blocks, then ADAPT will prompt you to input the number of the first and last trials for each block.

Note that the computing time for some of the measures is quite considerable. It is suggested that these analyses be run overnight.

### Architecture

Figure 2 illustrates the software architecture of ADAPT. The five main components are: Input (i.e., Log files), Pre-processing Modules, Data Analysis Modules, and Outputs (i.e., Graphs and Data).

After a subject completes a trial with DURESS II, all numerical data describing the state of the system at the time of the operator's control actions are retained and stored in a binary log file. These log files are the primary input to ADAPT. However, the data in these log files must first



be accessed and transformed by pre-processing modules so that the data are in a form that can be read by the data analysis modules.

The heart of ADAPT consists of various data analysis modules. A module (or a 'function') is a sub-program that is nested within a main program. Each module produces one type of human performance measure. Thus, different modules represent different ways of looking at the same subjects' behavior. The output for each of the modules are graphs and data. Each of the components in Figure 2 will now be described in more detail.

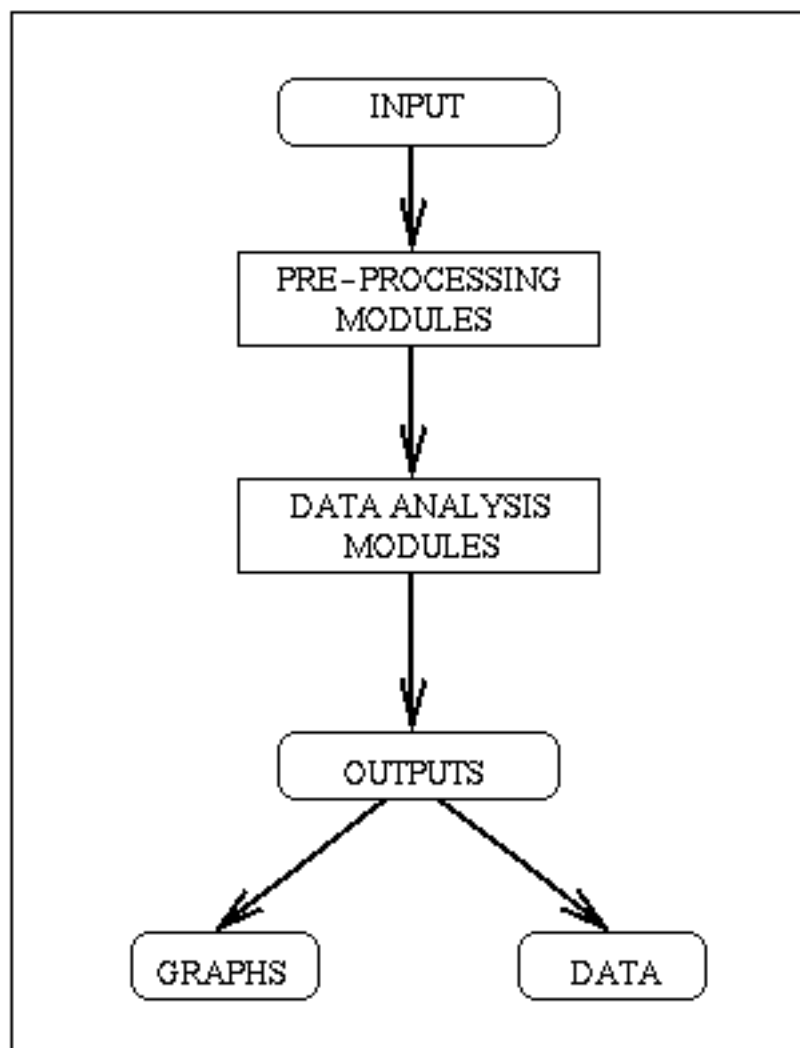


Figure 2: ADAPT Software Architecture.

### **Input: Log Files**

As mentioned previously, whenever a subject acts on any component of DURESS II, the simulation records the control action, the time, and the current values of all of the system variables. These recorded data are stored in a Log File in binary format. All of the analysis tools

described in the following sections use the Log Files as input. It is important to note, however, that the log files cannot be used directly to do the analysis because they are in binary format; a format not recognized by MATLAB. Thus, the log files must first be processed by the Pre-Processing Modules to convert the relevant data into the required format.

### **Pre-Processing Modules**

#### Extract\_Action

Action data are those data obtained when a subject acts on any of the components in DURESS II; whereas process data describe the state of the process. For example, the heater setting (e.g., HTR1) is an action variable because the subject can act directly on that component. In contrast, reservoir temperature (e.g., T1) is a process variable because it describes the state of DURESS II, rather than the state of a component that subjects can directly act on. When doing analysis on "Action data", we extract data from the Extract\_Action. This module is also written in the C Shell programming language. It also converts the data stored in the log files from binary to decimal format.

#### Extract\_All

The function "Extract\_All" is a module written in the C Shell programming language that extracts all of the data from the log files and converts this data stored from binary to decimal format. When doing analysis on either time, components or the settings of the simulation system, we extract data from Extract\_All.

### **Data Analysis Modules**

As shown in Figure 2, after the data from the log files have been pre-processed, they can be fed into the data analysis modules. Before we describe the modules themselves, there are some points that need to be noted about the software. All of the data analysis modules in ADAPT are written in the MATLAB programming language. The output of each module is presented in two formats, a graph and alphanumeric data. The alphanumeric format is represented as a matrix and the name of the matrix, for each function, is given in the description of the data. An example of the output graph created by each module is attached in Appendix 1.

For the description of some functions, the performance analysis can be conducted at three different levels of granularity. We can choose to aggregate the four goal variables, D1, D2, T1, and T2 (see Appendix 2) into a single, unified systems view by representing them into a single four-dimensional state space, where each of the four axes corresponds to one of the goal

variables. This systems level analysis produces one measure for the entire system. Alternatively, we can conduct a reservoir level analysis by using two separate state spaces, each defined by two dimensions, the temperature and demand goal variables for that reservoir. Finally, we can also conduct a variable level analysis by using four separate state spaces, each defined by one of the four goal variables. These three levels of analysis represent a trade-off between detail and ease of interpretation. Higher levels are easier to interpret because they provide a more aggregated view of performance. On the other hand, lower levels provide details that can be obscured by the integration required to conduct higher levels of analysis. Each of the data analysis modules will now be discussed in detail (cf. Howie & Vicente, in press).

### Trajectories in the Goal Space

*Function Name* : tgs.m

*Description of the function* :

There are two target variables for each reservoir, temperature and water outflow demand. These two goals can form a two dimensional state space in which a graph of water outflow vs. temperature can be plotted (cf. Sanderson, Verhage, & Fuld, 1989; Howie & Vicente, in press). For each trial, a trajectory is formed based on information extracted from the data recorded during the study. For a successfully completed startup task, the trajectory begins at the origin (i.e., the system is shut down) and the line traverses within the limits of the graphs until it reaches a goal region depicting the target state. Thus, in order to reach steady state, the line must end in a small goal region defined by the target values for that trial. These goal space trajectory graphs depict the current state of the system in terms of the goal variables, temperature and demand.

*Procedure*:

1. The states of the goal variables are normalized with respect to the goal values for that trial, so that we can compare values across trials (throughout this report, the bar notation signifies a normalized variable).

$$\overline{MO}_1(t) = MO_1(t) / D_1(t)$$

$$\overline{T}_1(t) = T_1(t) / T_{1goal}(t)$$

$$\overline{MO}_2(t) = MO_2(t) / D_2(t)$$

$$\overline{T}_2(t) = T_2(t) / T_{2goal}(t)$$

2. The pair  $(\overline{MO}_i(t), \overline{T}_i(t))$ ,  $i = 1, 2$  are plotted on the two dimensional space (note that time is implicit in the graphs).

For startup trials, the graphs always begin at the origin (0,0) and continue as a trajectory defining the state of the goal variables until it eventually stops at or very close to (1,1), the goal point.

*Outputs :*

Graphs: Two graphs are plotted for each trial for each subject to see how the trajectory in goal space changes within a trial. The first graph is a trajectory in the goal space for the upper reservoir, and the second one is the trajectory in the goal space for the lower reservoir.

Data: The data are represented in the form of a four-column matrix called *traj* :

$$[\bar{T}_1(t) \quad \overline{MO}_1(t) \quad \bar{T}_2(t) \quad \overline{MO}_2(t)]$$

### Lengths of Trajectories in the Goal Space

*Function Name : ltgs.m*

*Description of the function:*

Instead of merely drawing conclusions from the graphs of Trajectories in the Goal Space on the basis of visual inspection (qualitatively), we used the length of the trajectory as a quantitative measure of performance for each trial (Howie & Vicente, in press).

*Procedure:*

The lengths of trajectories in one trial are calculated according to the following two steps:

1. Normalize the four goal variables with respect to goal values.

$$\overline{MO}_1(t) = MO_1(t) / D_1$$

$$\bar{T}_1(t) = T_1(t) / T_{1goal}$$

$$\overline{MO}_2(t) = MO_2(t) / D_2$$

$$\bar{T}_2(t) = T_2(t) / T_{2goal}$$

2. Calculate the lengths of trajectories at the reservoir level (LTGS<sub>1</sub>, LTGS<sub>2</sub>) and system level (LTGS), respectively:

$$LTGS_1 = \sqrt{\sum_{t=0}^{\infty} (\bar{T}_1(t+1) - \bar{T}_1(t))^2 + (\overline{MO}_1(t+1) - \overline{MO}_1(t))^2}$$

$$LTGS_2 = \sqrt{\sum_{t=0}^{\infty} (\bar{T}_2(t+1) - \bar{T}_2(t))^2 + (\overline{MO}_2(t+1) - \overline{MO}_2(t))^2}$$

$$LTGS = \sqrt{LTGS_1^2 + LTGS_2^2}$$

*Outputs:*

Graphs: Three graphs are plotted for each subject to show how the length of trajectories in goal space changes across trials. For all three graphs, the y-axis is the length of the trajectory and the x-axis is trial number. The first two graphs are  $LTGS_1$  and  $LTGS_2$ , and the third graph is  $LTGS$ .

Data: The data for plotting these graphs is a four-column matrix called *lentgs*:

[ trial                       $LTGS_1$                        $LTGS_2$                        $LTGS$  ]

### Distance to the Goals

*Function Name:* dtg.m

*Description of the function :*

The distance to goals (DTG) is also based on a transformation of the data in the goal space trajectory graph (Shaw, Kadar, Sim, & Repperger, 1992; Howie & Vicente, in press). It depicts the distance between the current state of the goal variables to the target values of these variables. Distance to the goals is defined separately for the two reservoirs.

*Procedure:*

The calculation of DTG can be explained in three steps:

1. For each reservoir, the Euclidean distance between the state of the two goal variables and the target states could be calculated as follows:

$$DTG_1 = \sqrt{(T_1 - T_{1goal})^2 + (MO_1 - D_1)^2}$$

$$DTG_2 = \sqrt{(T_2 - T_{2goal})^2 + (MO_2 - D_2)^2}$$

However, this does not allow us to compare data across trials because the target values of the goal variables can change across trials. Furthermore, the two goal variables (temperature and demand) are not given equal weighting in the calculation because their scales differ. Thus, the variable with the larger scale (temperature) is artificially given a higher weighting.

2. In order to address these problems, we normalized the distance to the goals with respect to the goal values for each trial. Normalized distance to goals (NDTG) are defined as follows:

$$NDTG_1 = \sqrt{(T_1 / T_{1goal} - 1)^2 + (MO_1 / D_1 - 1)^2}$$

$$NDTG_2 = \sqrt{(T_2 / T_{2goal} - 1)^2 + (MO_2 / D_2 - 1)^2}$$

These two values can be used to calculate the normalized distance at the overall system goal. In this case, the normalized distance to the goals is defined as:

$$NDTG = \sqrt{NDTG_1^2 + NDTG_2^2}$$

These calculation represent an improvement over those described in step 1. However, there is a remaining problem, namely that the goal variables have a tolerance region associated with them. For example, in one experiment (JAERI I), the tolerance of the temperature goals was  $\pm 2$  degrees C, and the tolerance of the demand goals was  $\pm 1$ kg/s. If the actual temperature is within  $[T_{goal}-2, T_{goal}+2]$ , the temperature goals have been achieved. This applies to demand goals as well. These tolerances are not reflected in the calculations shown above.

3. To address this problem, the formulae can be modified as follows:

$$NDTG_1 = \sqrt{\left(\frac{\max\{|T_1 - T_{1goal}| - 2, 0\}}{T_{1goal}}\right)^2 + \left(\frac{\max\{|MO_1 - D_1| - 2, 0\}}{D_1}\right)^2}$$

$$NDTG_2 = \sqrt{\left(\frac{\max\{|T_2 - T_{2goal}| - 2, 0\}}{T_{2goal}}\right)^2 + \left(\frac{\max\{|MO_2 - D_2| - 2, 0\}}{D_2}\right)^2}$$

$$NDTG = \sqrt{NDTG_1^2 + NDTG_2^2}$$

This final set of formulae are the ones that are used by the module that calculates the distance to goals.

*Outputs:*

Graphs: For each subject, three graphs are plotted for each trial to determine how the distance to goals varies with time within a trial. The first two graphs are NDTG<sub>1</sub> and NDTG<sub>2</sub> vs. time, and the third graph is NDTG vs. time.

Data: The data for plotting these graphs is in *dist* which is a four-column matrix:

[ time            NDTG<sub>1</sub>        NDTG<sub>2</sub>        NDTG ]

### Area under Distance to the Goals

*Name of the function:* adtg.m

*Description of the function:*

The graph of DTGs only provides us with information about what happens within one trial. To track changes across trials, we can calculate the graph of the Area under the DTG graphs.

*Procedure:*

The area under the graph of DTG is calculated by using a built-in Matlab function called *trap*.

*Outputs :*

Graphs: In the graph of the Area under DTG (ADTG), the y-axis is the area under DTG and the x-axis is trial number. Three graphs are plotted for each subject. The first two graphs are the area under the curves of NDTG<sub>1</sub> and NDTG<sub>2</sub> vs. trial number, and the third graph is NDTG vs. trial number.

Data: The data for plotting these graphs is in *area* which is a four-column matrix. The columns consist of:

[time            Area of NDTG<sub>1</sub>                    Area of NDTG<sub>2</sub>        Area of NDTG]

### Rise Time

*Function Name :* risetime.m

*Description of the function:*

Rise time measures how quickly subjects can control DURESS II at the beginning of a trial. We operationally define rise time as the time it takes for each of the four goal variables to reach their respective lower goal boundaries for the first time. The rise time of the whole system is defined as the maximum of the four individual rise times.

*Outputs :*

Graphs: If more than one trial is to be analyzed, then a graph of rise time vs. trial will be plotted.

Otherwise, if only one trial is being analyzed, then the value of rise time will be displayed.

Data: The results are contained in *t\_time* with the first column as the trial number and the second column as the rise time corresponding to each trial.

### Time to Contact the Goals

-

*Function Name:* ttc.m

#### *Description of the function:*

Whereas rise time only gives one value for each subject for each trial, Time to Contact (TTC) is a dynamic measure of the time that is remaining for a goal variable to move from its current state to the lower boundary of the target region, given its current instantaneous velocity (Lee, 1976; van Westrenen, 1996). It is operationally defined for each goal variable as the distance to the respective lower boundary divided by the velocity. The maximum time of TTC is set by default at 300s, which corresponds to the steady state time required to stabilize the system. If the TTC becomes negative, it indicates that the parameter in question is moving away from the goal boundary. In that case, TTC is reset to 300s. TTC is only calculated during the rise time period. After a variable enters the goal region, the TTC measure is no longer meaningful.

#### *Procedure:*

For example, for MO<sub>1</sub>, TTC is calculated as follows:

1.  $TTC_{MO_1}(0) = 300s$
2. For  $0 < t < \text{rise time of } MO_1$ ,

$$TTC_{MO_1}(t) = (MO_1(t) - 1) / V_{MO_1(t)}$$

where  $V_{MO_1}(t)$  is the velocity of MO<sub>1</sub> at time t.



3. if  $TTC(t) > 300$  or  $TTC(t) < 0$ ,  $TTC(t) = 300s$

*Outputs :*

Graphs: Four graphs of TTC vs. time are plotted, one for each of the goal variables.

Data: The results are contained in *timect* with five columns.

[time    TTC(T<sub>1</sub>)    TTC(T<sub>2</sub>)    TTC(MO<sub>1</sub>)    TTC(MO<sub>2</sub>)]

Area under Time to Contact (ATTC)

*Function Name:* attc.m

*Description of the function:*

While useful, the graphs of TTC only provide information for one trial. As before, we can get some insight into what happens across trials by calculating the Area under the TTC graph (ATTC).

*Procedure:*

ATTC is calculated simply by using the MatLab function called *trap* .

*Outputs:*

Graphs: Four graphs of ATTC vs. time are plotted, one each for the four goal variables.

Data: The results are contained in *timearea* with five columns. The columns are represented as:

[trial            T1            T2            MO1            MO2]

Oscillation Measures

The last three modules (rise time, TTC, ATTC) measure subjects' performance in the initial phase of a trial, before they have reached the goal region for the first time. In this subsection, we describe several functions that measure subjects' performance after they have reached the goal

region for the first time but before they have been able to stabilize the goal variable. These functions are referred to as oscillation measures.

*Function Name* : aot.m

*Description of the function*:

There are four oscillation measures that are defined individually for each the four goal variables: oscillation duration, peak, number of oscillation and area of deviation per time unit. For each measure, we identify the maximum of the four goal values (e.g., which goal variable has the largest oscillation duration).

*Procedure*:

The four oscillation measures are defined as follows:

- 1) Oscillation duration is the time between the end of the rise time and the time when each goal variable goes into and stays in the goal region.
- 2) Peak is the percentage deviation with respect to the target value of each goal variable.
- 3) Number of Oscillations, as the term implies, is the number of oscillations (i.e., undershoots plus overshoots) made for each of the four goal variables, until that variable is stabilized in the goal region.
- 4) Area of deviation per unit time is the total area of deviations outside of the goal region divided by the duration of oscillation period for each goal variable. This is calculated by the MatLab function, *trap*.

*Outputs* :

Graphs: If more than one trial is being analyzed, then graphs of these four measures vs. trials will be plotted. Otherwise, then only the values are displayed.

Data: The results are contained in *timeot* with first column as the trial number, second to fifth column as oscillation duration, sixth to ninth column as peaks, tenth to thirteenth columns as number of oscillations, and fourteenth to seventeenth as the area of deviation per unit time.

Time to Goal Boundaries (TTGB)

The next two functions (TTGB, ATTGB) measure subjects' performance during the final phase during which the goal variables are in the target regions.

*Function Name:* ttgb.m

*Description of the function:*

Time to Goal Boundary (TTGB) is a dynamic measure of a subject's ability to control stability during steady state (van Westrenen, 1996). It is defined individually for each output variable as the distance to the upper or lower target boundaries divided by the current instantaneous velocity. The maximum value of TTGB is defined to be 300s since this is the duration that subjects are required to keep the system in the goal region.

*Procedure:*

1. Suppose the velocity of  $MO_1$  at time  $t$  is  $V_{MO_1}(t)$ , if  $V_{MO_1}(t) > 0$ , then

$$TTGB_{MO_1}(t) = (MO_1 + 1 - MO_1(t)) / V_{MO_1}(t)$$

and if  $V_{MO_1}(t) < 0$ , then

$$TTGB_{MO_1}(t) = (MO_1(t) - D_1 - 1) / V_{MO_1}(t)$$

and if  $V_{MO_1}(t) = 0$ , then

$$TTGB_{MO_1}(t) = 300 \text{ s.}$$

2. If  $TTGB_{MO_1}(t) > 300 \text{ s}$ , then set  $TTGB_{MO_1}(t) = 300$ .

*Outputs :*

Graphs: Four graphs of TTGB vs. time are plotted for one trial. In order to provide a common baseline for all four goal variables, only the last 300s for each variable are plotted.

Data: The results are contained in *timecb* with five columns.

[time      TTGB(T1)      TTGB(T2)      TTGB(MO1)      TTGB(MO2)]

Area under the Time to Contact Goal Boundaries

*Name of the function:* attgb.m

*Description of the function:*

Whereas TTGB is a dynamic measure for a single trial, Area under the graphs of TTGB (ATTGB) is an aggregate measure that can be calculated for each trial. It thereby allows us to evaluate performance across a set of trials. Because ATTGB depends on both TTGB and the

duration of the stability period, ATTGB is normalized by the largest possible values of the time duration of TTGB multiplied by 300. The multiplication by 300 corresponds to the steady state time required to stabilize the system. This normalization allows comparison of trials that have different time periods.

*Procedure:*

ATTGB is calculated simply by using the MatLab function called *trap* .

*Outputs:*

Graphs: Four graphs of ATTGB vs. trial number are plotted, each graph representing one of the goal variables.

Data: The results are contained in *timearea* with five columns.

[trial    ATTGB(T<sub>1</sub>)    ATTGB(T<sub>2</sub>)    ATTGB(MO<sub>1</sub>)    ATTGB(MO<sub>2</sub>)]

Variance of Goal Variables (Outputs)

The next four measures are similar in that they are all measures of variability and are all based on an abstraction hierarchy representation of DURESS II (see Bisantz & Vicente, 1994). Each measure calculates the variance in system behavior (or subjects' actions for the Component Settings level) at one of four levels of the abstraction hierarchy (Goals, Mass & Energy, Flows & Heat Transfer, and Component Settings).

*Function Name:* var\_output.m

*Description of the function:*

This measure shows the consistency in subjects' performance across trials, at the level of goal variables (outputs). The state of the four goal variables,  $(\overline{MO_1}(t), \overline{MO_2}(t), \overline{T_1}(t), \overline{T_2}(t))$  can be plotted against time, creating one trajectory in five-dimensional space for each trial. The variance in these trajectories within a block of trials is then calculated.

*Procedure:*

The variance of goal variables over a block of trials is defined as follows:

1. **Time Shift:** Generally, there is a delay between the beginning of a trial and the time of the first action by the subject. The magnitude of this delay varies across subjects, and within-subjects across trials. For the purposes of this analysis, this subject response delay is noise, so it should be removed. If there is a time delay,  $\tau$  for the trial, then the four goal variables  $T_1$ ,  $T_2$ ,  $MO_1$ ,  $MO_2$  should be shifted by  $\tau$  so that they all start at time at 0. This leads to functions  $T_1(t-\tau)$ ,  $T_2(t-\tau)$ ,  $MO_1(t-\tau)$ , and  $MO_2(t-\tau)$ , respectively.
2. **Normalization:** For each trial, the four goal variables:  $MO_1(t-\tau)$ ,  $MO_2(t-\tau)$ ,  $T_1(t-\tau)$ ,  $T_2(t-\tau)$  are normalized with respect to their set points, which leads to  $\overline{MO}_1(t-\tau)$ ,  $\overline{MO}_2(t-\tau)$ ,  $\overline{T}_1(t-\tau)$ , and  $\overline{T}_2(t-\tau)$ , respectively. Normalization allows us to compare all of the variables across trials from a common reference scale.
3. **Linear Interpolation:** The DURESS II simulation only logs the system state at the time of a subject action rather than at a constant sampling interval (Orchanian et al., 1996). Thus, if there is a long time between actions, then the state of the system during this time will be known and must be derived. To recover these data,  $\overline{MO}_1(t-\tau)$ ,  $\overline{MO}_2(t-\tau)$ ,  $\overline{T}_1(t-\tau)$ ,  $\overline{T}_2(t-\tau)$  are linearly interpolated at a rate of 3 seconds over the first 300 s of a trial. This interpolation interval was chosen based on knowledge of the bandwidth of the DURESS II dynamics. Thus, we get  $\overline{MO}_1(t_i)$ ,  $\overline{MO}_2(t_i)$ ,  $\overline{T}_1(t_i)$ ,  $\overline{T}_2(t_i)$ , with  $t_1 = 0$ ,  $t_2 = 3$ ,  $t_3 = 6$ , ...,  $t_{101} = 300$ .
4. The multi-dimensional, time-wise variance at each  $t_i$  is calculated by:

$$\text{var}(t_i) = \frac{\sum_{j=1}^n (\overline{MO}_1^j(t_i) - \text{ave}\overline{MO}_1(t_i))^2 + (\overline{MO}_2^j(t_i) - \text{ave}\overline{MO}_2(t_i))^2 + (\overline{T}_1^j(t_i) - \text{ave}\overline{T}_1(t_i))^2 + (\overline{T}_2^j(t_i) - \text{ave}\overline{T}_2(t_i))^2}{n-1}$$

where  $\overline{MO}_1^j(t_i)$ ,  $\overline{MO}_2^j(t_i)$ ,  $\overline{T}_1^j(t_i)$ ,  $\overline{T}_2^j(t_i)$  are normalized water outflow rates and temperatures at time  $t_i$  of trial  $j$  within the block of sampled trials.  $\text{ave}\overline{MO}_1(t_i)$ ,  $\text{ave}\overline{MO}_2(t_i)$ ,  $\text{ave}\overline{T}_1(t_i)$ ,  $\text{ave}\overline{T}_2(t_i)$  are the average values of  $\overline{MO}_1^j(t_i)$ ,  $\overline{MO}_2^j(t_i)$ ,  $\overline{T}_1^j(t_i)$ ,  $\overline{T}_2^j(t_i)$ , respectively, over the same block of trials. For example,

$$\text{ave}\overline{MO}_1(t_i) = \sum_{j=1}^n \overline{MO}_1^j(t_i) / n$$

5. The multi-dimensional variance over the entire 300 s span can then be calculated as follows:

$$\text{variance} = \frac{\int_0^{300} \text{var}(t) dt}{300} \approx \frac{\sum_{i=0}^{99} \text{var}(3i) \times 3}{300}$$

*Outputs:*

Graphs: A graph of Variance vs. block number is plotted.

Data: The results are contained in *variance* with two columns.

[trial number    variance]

### Variance of Mass and Energy

*Function Name:* var\_me1.m and var\_me2.m

*Description of the function:*

At this level of the abstraction hierarchy, there are twelve variables that describe the state of DURESS II: MO1, EI1, EO1, M1, E1, MI1, MO2, EI2, EO2, M2, E2, and MI2 (see Bisantz & Vicente, 1994). With the addition of time, they form a thirteen-dimensional space. Multi-variance is defined in the same way as variance at the goal level, except that the variables are normalized with respect to their maximum possible values, which are pre-defined in the configuration files. This normalization process removes any artificial, differential-weighting effects caused by heterogeneous numerical scales across variables.

*Procedure:*

There are two ways to calculate variance of Mass and Energy which is why there are two sub-functions for this function.

1. The first way is by normalization w.r.t. the goal variables (D1, T1, D2 and T2), as well as scale:

For Reservoir 1:	For Reservoir 2:
------------------	------------------

$\overline{MO1} = MO1/D1$ $\overline{EI1} = EI1/D1 \times T1 \times 2,090,000$ $\overline{EO1} = EO1/D1 \times T1 \times 2,090,000$ $\overline{M1} = M1$ $\overline{E1} = E1/168,000,000$ $\overline{MI1} = MI1/D1$	$\overline{MO2} = MO2/D2$ $\overline{EI2} = EI2/D2 \times T2 \times 2,090,000$ $\overline{EO2} = EO2/D2 \times T2 \times 2,090,000$ $\overline{M2} = M2$ $\overline{E2} = E2/168,000,000$ $\overline{MI2} = MI2/D2$
---	---

2. The second method is by normalization w.r.t. scale only:

For Reservoir 1:	For Reservoir 2:
$\overline{MO1} = MO1$ $\overline{EI1} = EI1/2,090,000$ $\overline{EO1} = EO1/2,090,000$ $\overline{M1} = M1$ $\overline{E2} = E1/168,000,000$ $\overline{MI1} = MI1$	$\overline{MO2} = MO2$ $\overline{EI2} = EI2/2,090,000$ $\overline{EO2} = EO2/2,090,000$ $\overline{M2} = M2$ $\overline{E2} = E2/168,000,000$ $\overline{MI2} = MI2$

*Outputs:*

Graphs: In the menu list under Action of Mass and Energy, there is an option of getting the output of either of the sub-functions. The user can choose the required menu item desired for analysis.

For both the outputs, a graph is displayed for variance vs. block number.

Data: The results are contained in *variance* with two columns:

[trial number      variance of the corresponding block]

Variance of Flows and Heat Transfer

*Function Name:* var\_flow.m

*Description of the function:*

At this level of the abstraction hierarchy, there are 10 variables that describe the state of DURESS II: FA, FA<sub>1</sub>, FA<sub>2</sub>, FB, FB<sub>1</sub>, FB<sub>2</sub>, FO<sub>1</sub>, FO<sub>2</sub>, HTR<sub>1</sub>, HTR<sub>2</sub>. Including time, they form an eleven-dimensional space. As before, the variables are normalized with respect to their maximum values before calculating the variance in trajectories.

*Outputs:*

Graphs: A graph of variance vs. block number is plotted.

Data: The results are contained in *variance* with two columns.

[trial number      variance of corresponding block]

Variance of Component Settings (Actions)

*Function Name:* var\_com.m

*Description of the function:*

In the fourth level of the abstraction hierarchy for DURESS II, there are twelve different components that subjects can act on: PA, PB, VA, VA<sub>1</sub>, VA<sub>2</sub>, VB, VB<sub>1</sub>, VB<sub>2</sub>, VO<sub>1</sub>, VO<sub>2</sub>, HTR<sub>1</sub>, HTR<sub>2</sub>. With time, these variables form a thirteen-dimensional action state space. Multi-variance is defined in the same way as variance at the goal level, except that: a) the variables are normalized with respect to their maximum settings; and b) time is represented on an ordinal scale (e.g., time of first action, time of second action, etc.) rather than on an interval scale.

*Outputs:*

Graphs: Variance is plotted vs. block number.

Data: The results are contained in *variance* with two columns:

[trial number                      variance of the corresponding block]

Actions Frequency Distributions

*Function Name:* afd.m



*Description of the function:*

Another way of looking at subjects' actions is with the use of frequency distribution diagrams. These allow us to see if subjects tend to set components to particular settings consistently by the use of a 'recipe' or a procedure. To test for this consistency, this function calculates the relative frequency of settings for each component for a block of trials (e.g., how often did a subject set HTR 1 between 6.5 and 7.5). The frequencies are normalized with respect to the largest one.

*Procedure:*

The range of each component was divided into 11 equally-spaced bins, except for the pumps which have only two bins (on or off). Given this discretization, the formula for calculating the Action Frequency Distribution for any component is:

$$afd(k) = \frac{action(k)}{\max(action(k))}$$

for  $k = 1, 2, \dots, 11$  (2 for the pumps)

$k$  = bin number

$action(k)$  = number of times that the component is set to bin  $i$

*Outputs:*

Graphs: Twelve graphs of relative frequency vs. setting are plotted, one for each component.

Data: The action frequency of the twelve components are contained in *ppa*, *ppb*, *pva*, *pva1*, *pva2*, *pvb*, *pvb1*, *pvb2*, *phtr1*, *phtr2*, *pvo1*, *pvo2*.

Entropy

*Function Name:* et.m

*Description of the function:*

Information theory was created in the context of communication systems, where the interaction between a *transmitter* and a *receiver* is of high interest (Shannon & Weaver, 1947). The general aim of communication is to make sure that the receiver accurately receives the messages from the transmitter; in other words, to create a strong correlation between the messages from the transmitter and receiver.

*Procedure:*

The quantity that is used in information theory as a measure of non-parametric correlation between transmitter and receiver is called entropy. Let X and Y be discrete variables, ranging over the set  $\gamma_x = \{y_1, y_2, \dots, y_n\}$  and  $\gamma_y = \{y_1, y_2, \dots, y_n\}$ . The entropy of X and Y are defined in terms of their probability distributions:

$$H(X) = \sum_{x \in \gamma_x} p(x) \log_2(p(x))$$

$$H(Y) = \sum_{y \in \gamma_y} p(y) \log_2(p(y))$$

H(X) is a measure of the non constancy or variability of X over its possible range  $\gamma_x$ : H(X) = 0 if and only if X definitely takes one particular value of  $\gamma_x$ , i.e., some x in  $\gamma_x$  has probability 1 while others are 0, and H(X) is maximum if and only if all x are equally probable. The same applies to H(Y).

Similarly, we can define the joint entropy between X and Y by using the two-dimensional, joint probability distribution:

$$H(X, Y) = \sum_{x \in \gamma_x, y \in \gamma_y} p(x, y) \log_2(p(x, y))$$

Mutual entropy allows us to define another important quantity defined in information theory, the information transmitted between X and Y:

$$T(X : Y) = H(X) + H(Y) - H(X, Y)$$

T(X, Y) is a non-parametric measure of the strength of the relation (or interaction) between X and Y: T(X : Y) = 0 if and only if X and Y are statistically independent, and T(X : Y) is a maximum (equal to  $\min \{H(X), H(Y)\}$ ) if and only if X is completely dependent upon Y or vice versa.

In the context of DURESS II, we define entropy for components and settings:  $H_C$  and  $H_S$ . There are twelve components that an operator can act on. Settings are continuous so we discretize them into five fuzzy sets: E, L, M, H, F. The membership functions are defined as:

$$\mu_E(x) = \begin{cases} -0.40x + 1.00 & 0 < x \leq 2.50 \\ 0 & \textit{otherwise} \end{cases}$$

$$\mu_L(x) = \begin{cases} -0.40x + 2.00 & 2.50 < x \leq 5.00 \\ 0.4x & 0.00 < x \leq 2.50 \\ 0 & \textit{otherwise} \end{cases}$$

$$\mu_M(x) = \begin{cases} -0.40x + 3.00 & 5.00 < x \leq 7.50 \\ 0.4x - 1.00 & 2.50 < x \leq 5.00 \\ 0 & \textit{otherwise} \end{cases}$$

$$\mu_H(x) = \begin{cases} -0.40x + 4.00 & 7.50 < x \leq 10.00 \\ 0.4x - 2.00 & 5.00 < x \leq 7.50 \\ 0 & \textit{otherwise} \end{cases}$$

$$\mu(x) = \begin{cases} 0.40x - 3.00 & 7.50 < x \\ 0 & \textit{otherwise} \end{cases}$$

The main advantage of this group of membership function is that

$$\sum_{\sigma \in \{E, L, M, H, F\}} (x) = 1$$

for all  $x \in [0, 10]$  (for VO1 and VO2, whose setting ranges are  $[0, 20]$ , we divide it by 2 first).

The maximum of  $H_C$  is  $\log_2(12)$ , the maximum of  $H_S$  is  $\log_2(5)$ , and the maximum of  $T_{CS}$  is  $\min\{H_C, H_S\}$ . The entropy and transmission values are then normalized with respect to these corresponding values:

$$\bar{H}_c = H_c / \log_2(12), \quad \bar{H}_s = H_s / \log_2(5), \quad \bar{T}_{cs} = T_{cs} / \min\{H_c, H_s\}$$

For a block of trials, the frequency of each component and setting can be determined, and then entropy and transmission measures can be calculated.

*Output:*

Graph: Three graphs are displayed, the first one shows the normalized entropy of components vs. block number, the second shows the normalized entropy of settings vs. block number, and the third one shows the normalized transmission vs. block number.

Data: The results are contained in *entropy* with seven columns.

[trial number	$H_C$	$H_S$	$T_{CS}$	$\bar{H}_c$	$\bar{H}_s$	$\bar{T}_{cs}$ ]
---------------	-------	-------	----------	-------------	-------------	------------------

Contingency Table

*Function Name:* ct.m

*Description of the function:*

Contingency table analyses can also be used to indicate the interaction between two elements (Siegel, 1956). For each block of trials, four contingency tables can be created:

1. subject  $\times$  component
2. interface  $\times$  component
3. subject  $\times$  setting
4. interface  $\times$  setting

A  $\chi^2$  value can be calculated for each of these tables to show the strength of the correlation for a particular block of trials.

*Output:*

Graphs: Four graphs are displayed, showing the  $\chi^2$  value vs. block number for the four contingency tables.

Data: The  $\chi^2$  value for each block is stored in four variables respectively:

*chi\_sa* : subject  $\times$  component

*chi\_sa* : interface  $\times$  component

*chi\_sa* : subject  $\times$  setting

*chi\_sa* : interface  $\times$  setting

Actions vs. TTGB, TTC, and TTFB

The remainder of the measures described in this report are all measures of adaptation. They evaluate, in different ways, the extent to which subject behavior is tailored to the structure or state of DURESS II. In this subsection, we describe measures of adaptation to system state. In the following subsections, we describe measures of adaptation to system structure.

*Function Name:* action\_tt.m

*Description of the function:*

Subjects' actions can be classified, by intent, according to the following categories:

1. Fine-tuning actions: The purpose of these actions is to keep the system within the goal boundaries. Those actions during the steady state period of each trial (i.e., the last five

minutes) are believed to be of this type. During this period, one might expect that actions may be driven by the current value of TTGB.

A graph of action frequency vs. TTGB for the starting period is plotted. For a block of trials, actions during the last five minutes are discretized into different bins, according to the values of TTGB during the time of the action. The maximum TTGB is pre set at 300s, and TTGB is divided into 20 bins: 0-15, 15-30, ..., 285-300. For example, if an action is made when the value of TTGB is 289, then that action will be put into the last bin (285-300).

2. Goal oriented actions: The aim of these actions is to bring the system into the goal boundaries. We consider actions during the transient period of each trial (i.e. the period not including the last five minutes) to be of this type. During this period, one might expect that actions may be driven by the current value of TTC.

A graph of actions frequency vs. TTC is plotted. The maximum TTC is pre-set as 300s, and TTC is divided into 20 bins: 0-15, 15-30, ....., 285-300.

3. Failure avoiding actions: While the subjects try to control the system to meet the goal variables, they also need to avoid safety boundaries of the system. There are five failure boundaries that should be taken into consideration:

- (a) boiling of the two reservoirs,
- (b) overflow of the two reservoirs,
- (c) overheating of the two reservoirs,
- (d) pump damage due to close of down stream valves and open of pumps,
- (e) time-out (45 minutes)

Failure avoiding actions are those actions that are intended to avoid failures. It is believed that most of such actions occur in the transient period of each trial (i.e., the period not including the last five minutes). During this period, one might expect that actions may be driven by the current value of the time to failure boundary (TTFB).

TTFB is defined as the minimum time to reach the above five failure boundaries (the maximum is 300s ). Each action is classified into different bins, as above, according to TTFB when the action has taken place.

*Output:*

Graphs: Three graphs are displayed, revealing the actions distribution over TTC, TTGB, and TTFB.

Data: The variables needed to plot the graphs are contained in *bin*, which is a three-column matrix.

[TTGB TTC TTFB]

### Reservoir Volumes at Steady State

*Function Name:* vol.m

#### *Description of the function:*

The reservoir volumes at steady state (i.e., the end of the startup period) are a partial indication of subjects' control strategies (see Pawlak & Vicente, 1996 for details). This function plots the final reservoir volume vs. trial number, for each reservoir. These graphs show if any subjects had preferences for higher or lower volumes, and whether these preferences were stable across trials.

#### *Outputs:*

Graphs: If the trial number is greater than one, then two graphs are plotted for steady state volume vs. trial number, one for each reservoir. Otherwise, if one trial is being analyzed, the values of the volumes are the only outputs.

Data: The results are contained in *volume* with three columns.

[trial number  $V_{\text{upper reservoir}}$   $V_{\text{lower reservoir}}$ ]

### Mass vs. Energy

*Function Name:* me.m

#### *Description of the function:*

This graph is a high-level, dynamic description of the behavior of DURESS II within a trial (Howie & Vicente, in press). The current mass of the water in a reservoir is plotted with respect to the current total energy in the same reservoir. Temperature, defined as the amount of energy per unit mass, is represented by lines of constant temperature (isotherm) on this graph. One of these isotherms represents the target temperature for a particular reservoir. This two-dimensional space is used as a frame of reference for plotting system behavior. For a successful start-up trial, the trajectory begins at the origin, and eventually ends on (or within tolerance of) the target

temperature isotherm. The graphs of these trajectories represent how well subjects are able to coordinate the control of mass and energy.

*Outputs:*

Graphs: Two graphs are plotted, one for upper reservoir and the other for the lower reservoir.

Data: No data needs to be saved.

Actions on Heaters

*Function Name:* afd\_heaters.m

*Description of the function:*

It can be seen from the equations governing DURESS II that the ratio of HTR1 to D1 should be 1:1 to stabilize T1 at the target value. Similarly, the ratio of HTR2 to D2 should be 1:3 to stabilize T2 at the target value. These specific ratios are due to the system constraints at steady state. We can see that to what extent a subject is aware of these constraints by creating action frequency diagrams as a function of the ration between heater setting and demand.

*Procedure:*

The formula to calculate Actions on Heaters is:

$$afd(k) = \frac{action(k)}{\max(action(k))}$$

$k = 1, 2, \dots n$

$k = \text{bin number}$

$action(k) = \text{the number of times the heaters are set at bin } k$

bin size for Heater 1:  $[0, \frac{1}{3}), [\frac{1}{3}, \frac{2}{3}), [\frac{2}{3}, \frac{3}{3}), \dots$

bin size for Heater 2:  $[0, \frac{1}{8}), [\frac{1}{8}, \frac{2}{8}), [\frac{2}{8}, \frac{3}{8}), \dots$

*Output:*

Graphs: Two graphs of frequency vs. discrete ratio of heater setting to demand goal are plotted, one for each reservoir.

Data: The action frequency of the two heaters are contained in *phtr1* , *phtr2* respectively.

## Constraint Surfaces:

*Name of the function:* landscape.m

*Description of the function:*

Like any other simulation, the variables in DURESS II are subject to mathematical constraints (see Vicente, 1991). These constraints can be visualized as hyper-surfaces on which trajectories of subject behavior or process behavior can be plotted (cf. Gibson & Crooks, 1938). The location of the trajectories on the hyper-surfaces gives some insights into the control strategies a subject used to cope with the constraint represented by that surface.

We identified seven qualitatively different constraints for DURESS II that could be used in this manner. (An eighth constraint relating mass level, energy level, and reservoir temperature has already been discussed above). Because of the symmetries in the system, each of these constraints can be instantiated in two ways (i.e., for both feedwater streams, or for both reservoirs), leading to 14 quantitatively different constraints.

1. Relation between  $VA_1$ ,  $VA_2$ ,  $VA$ , and  $FA$  ( $VB_1$ ,  $VB_2$ ,  $VB$ , and  $FB$ ):

$$FA = \begin{cases} VA & VA_1 + VA_2 > VA \\ VA_1 + VA_2 & otherwise \end{cases}$$

$$FB = \begin{cases} VB & VB_1 + VB_2 > VB \\ VB_1 + VB_2 & otherwise \end{cases}$$

$VA_1+VA_2$ ,  $VA$ , and  $FA$  ( $VB_1$ ,  $VB_2$ ,  $VB$ , and  $FB$ ) are used as the three axes, receptively.

2. Relation between  $VA_1$ ,  $VA_2$ ,  $FA$ , and  $FA_2$  ( $VB_1$ ,  $VB_2$ ,  $FB$ , and  $FB_2$ ):

$$FA_1 = (FA * VA_1) / (VA_1 + VA_2) = FA / (1 + x),$$

$$FB_1 = (FB * VB_1) / (VB_1 + VB_2) = FB / (1 + y),$$

where:  $x = VA_2 / VA_1$ ,  $y = VB_2 / VB_1$

$x$  ( $y$ ),  $FA$  ( $FB$ ), and  $FA_1$  ( $FB_1$ ) are used as the three axes, respectively.

3. Relation between  $FA_1$ ,  $FA_2$ , and  $FA$  ( $FB_1$ ,  $FB_2$ , and  $FB$ ):

$$FA_2 = FA - FA_1 \quad \text{for } FA \geq FA_1$$



$$FB_2 = FB - FB_1 \quad \text{for } FB \geq FB_1$$

FA<sub>1</sub>, FA<sub>2</sub>, and FA (FB<sub>1</sub>, FB<sub>2</sub>, and FB) are used as the three axes, respectively.

4. Relation between FA<sub>1</sub>, FB<sub>1</sub>, and MI<sub>1</sub> (FA<sub>2</sub>, FB<sub>2</sub>, and MI<sub>2</sub>):

$$FA_1 + FB_1 = MI_1$$

$$FA_2 + FB_2 = MI_2$$

FA<sub>1</sub>, FB<sub>1</sub>, and MI<sub>1</sub> (FA<sub>2</sub>, FB<sub>2</sub>, and MI<sub>2</sub>) are used as the three axes, respectively.

5. Relation between EO<sub>1</sub>, MO<sub>1</sub>, and T<sub>1</sub> (EO<sub>2</sub>, MO<sub>2</sub>, and T<sub>2</sub>):

$$EO_1 = D_1 * C_p * T_1$$

$$EO_2 = D_2 * C_p * T_2$$

where  $C_p = 4.200 \text{ Jkg}^{-1} \text{ K}^{-1}$  is the specific heat capacity of water.

EO<sub>1</sub>, MO<sub>1</sub>, and T<sub>1</sub> (EO<sub>2</sub>, MO<sub>2</sub>, and T<sub>2</sub>) are used as the three axes, respectively.

6. Relation between EI<sub>1</sub>, FH<sub>1</sub>, and MI<sub>1</sub> (EI<sub>2</sub>, FH<sub>2</sub>, and MI<sub>2</sub>):

$$EI_1^Y = FH_1 + C_p * T_i * MI_1,$$

$$EI_2^Y = FH_2 + C_p * T_i * MI_2$$

EI<sub>1</sub>, FH<sub>1</sub>, and MI<sub>1</sub> (EI<sub>2</sub>, FH<sub>2</sub>, and MI<sub>2</sub>) are used as the three axes, respectively.

7. Relation between MI<sub>1</sub>, MO<sub>1</sub>, and  $\check{V}_1$  (MI<sub>2</sub>, MO<sub>2</sub>, and  $\check{V}_2$ ):

$$\check{V}_1 = (MI_1 - MO_1) / \rho,$$

$$\check{V}_2 = (MI_2 - MO_2) / \rho$$

where  $\rho$  is the density of water =  $1 \text{ gm/cm}^3$ .

MI<sub>1</sub>, MO<sub>1</sub>, and  $\check{V}_1$  (MI<sub>2</sub>, MO<sub>2</sub>, and  $\check{V}_2$ ) are used as the three axes, respectively.

*Procedure:*

Since we can only view at most a three dimensional space, we have separated some of the constraints which contain more than three variables into sub-constraints which only consist of

three variables. For each graph, first we draw the constraint surface which is defined by the sub-constraints, representing the physical constraint that an actual trajectory must obey. Then we draw the actual trajectory on the surface to show the behavior of the subject or the process for a particular trial.

*Output :*

Graphs: Graphs are displayed to show the trajectory for a particular subject for a particular trial on of the constraint surface of interest.

Data: No data needs to be saved.

### **Conclusions**

This report is a user's manual for the software package, ADAPT, developed as part of the JAERI IV research project. Although it was developed for this project, ADAPT can be used in any subsequent research that requires evaluation of human performance in the DURESS II microworld. Moreover, it seems that many of the measures comprising ADAPT can perhaps be used to measure human performance in other dynamic systems, including other microworlds, full-scope simulators, and perhaps even operational systems in industry. ADAPT retains some features from MatLab, which is its development environment. ADAPT is also flexible in the sense that users can add more modules to the package, and can use it for other purposes with a minimum of modification effort.

### **Acknowledgments**

This research was sponsored by a research contract from the Japan Atomic Energy Research Institute (Dr. Fumiya Tanabe, contract monitor), and by grants from the Natural Sciences and Engineering Research Council of Canada. We would like to thank Greg Jamieson for commenting on an earlier draft, and Dr. Tanabe, Dr. Robert Shaw and the other members of the Intentional Dynamics Laboratory at the University of Connecticut for their contributions.

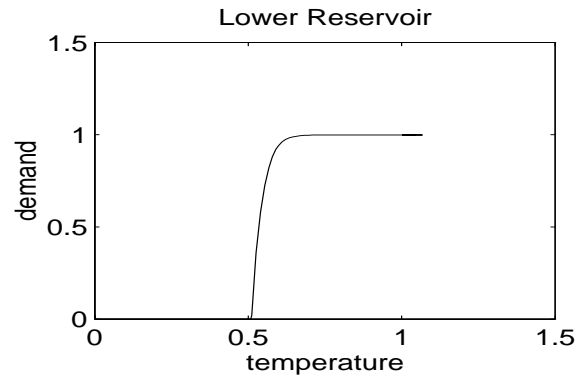
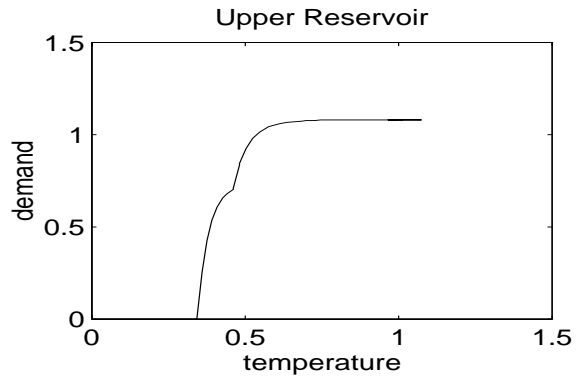
### **References**

- Bisantz, A. M., & Vicente, K. J. (1994). Making the abstraction hierarchy concrete. International Journal of Human-Computer Studies, 40, 83-117.
- Gibson, J. J., & Crooks, L. E. (1938). A theoretical field-analysis of automobile-driving. American Journal of Psychology, 51, 453-471.

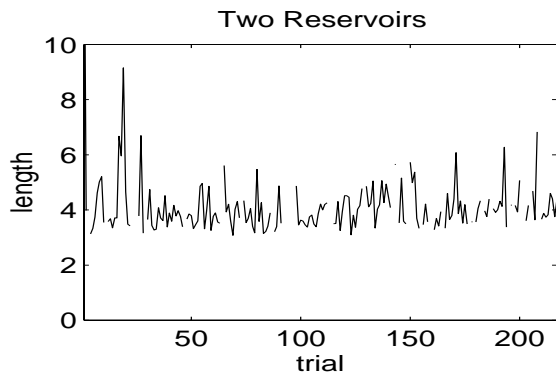
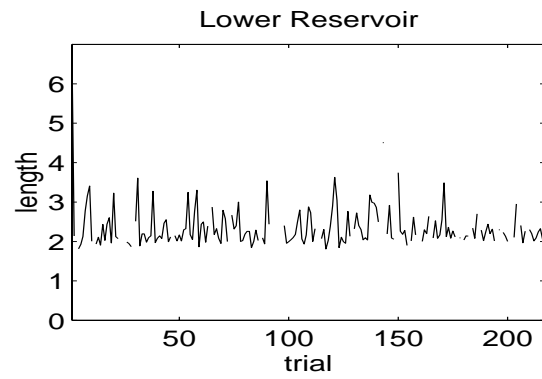
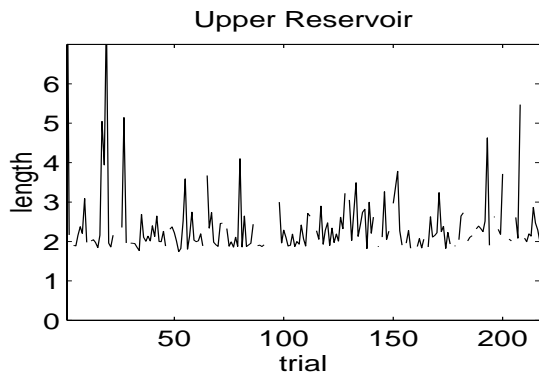
- Howie, D. E., & Vicente, K. J. (in press). Measures of operator performance in complex, dynamic microworlds: Advancing the state of the art. Ergonomics.
- Lee, D. N. (1976). A theory of visual control of braking based on information about time-to-collision. Perception, 5, 437-459.
- Orchanian, L. C., Smahel, T. P., Howie, D. E., & Vicente, K. J. (1996). DURESS II user's manual: A thermal-hydraulic process simulator for research and teaching (CEL 96-05). Toronto: University of Toronto, Cognitive Engineering Laboratory.
- Pawlak, W. S., & Vicente, K. J. (1996). Inducing effective operator control through ecological interface design. International Journal of Human-Computer Studies, 44, 653 - 688.
- Sanderson, P. M., Verhage, A. G., & Fuld, R. B. (1989). State-space and verbal protocol methods for studying the human operator in process control. Ergonomics, 32, 1343-1372.
- Shannon, C. Z., & Weaver, W. (1947). The mathematical theory of communication. Urbana, IL: University of Illinois Press.
- Shaw, R. E., Kadar, E., Sim, M. & Repperger, D. W. (1992). The intentional spring: A strategy for modeling systems that learn to perform intentional acts. Journal of Motor Behavior, 24, 3-28.
- Siegel, S. (1956). Nonparametric statistics for the behavioural sciences. New York: McGraw-Hill.
- van Westrenen, F. (1996). Process stability and workload: Control behaviour during manual control of a simulated slowly responding process. Manuscript submitted for publication.
- Vicente, K. J. (1991). Supporting knowledge-based behavior through ecological interface design (Unpublished doctoral dissertation). Urbana, IL: University of Illinois at Urbana-Champaign.
- Vicente, K. J. (1997). Operator adaptation in process control: A three-year research program. Control Engineering Practice, 5, 407-416.

## Appendix 1: Graphs of Data Analysis Modules

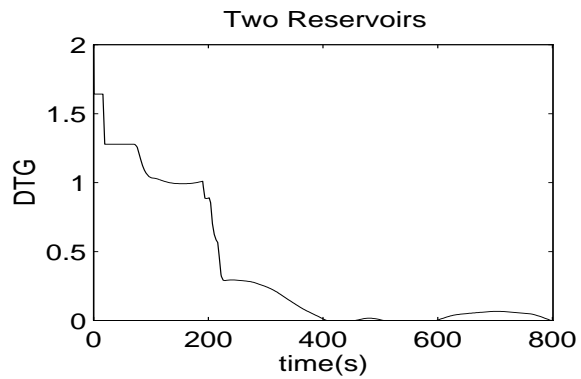
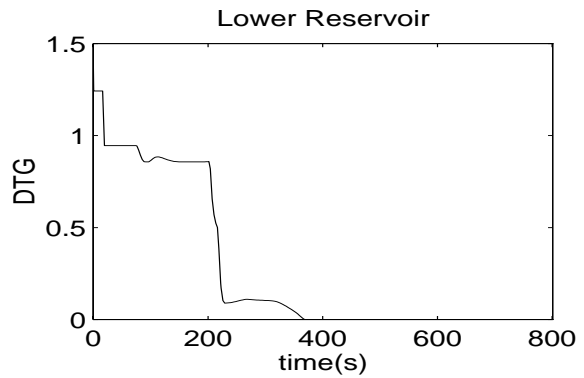
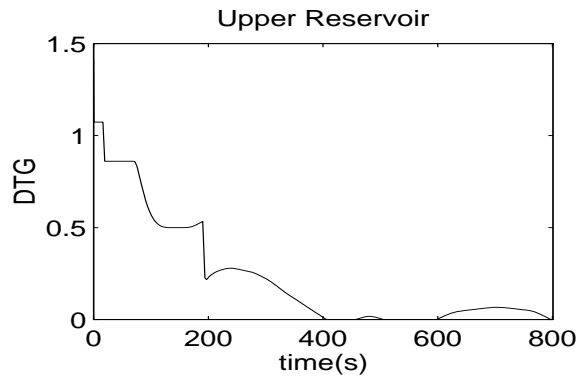
### Trajectories in the Goal Space



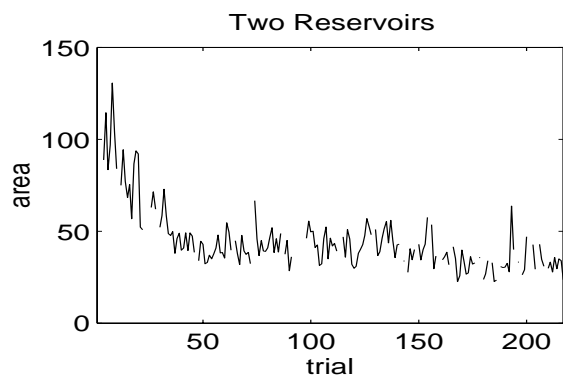
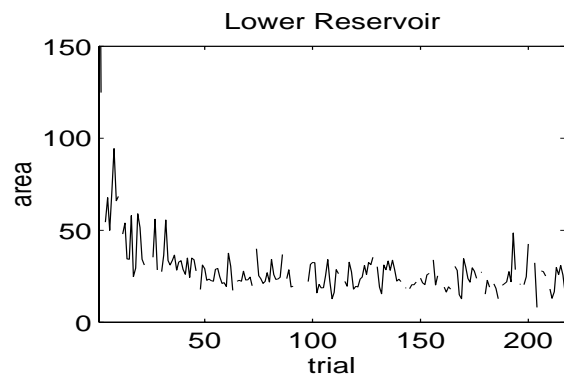
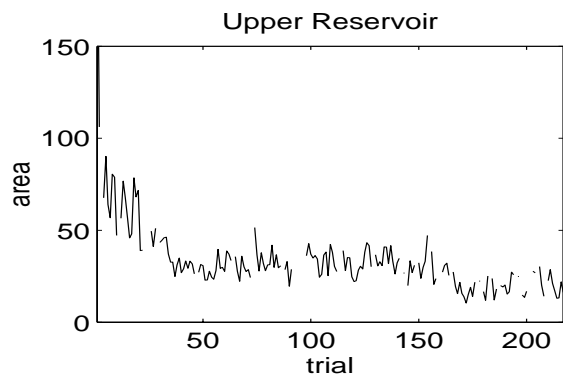
### Lengths of Trajectories in the Goal Space



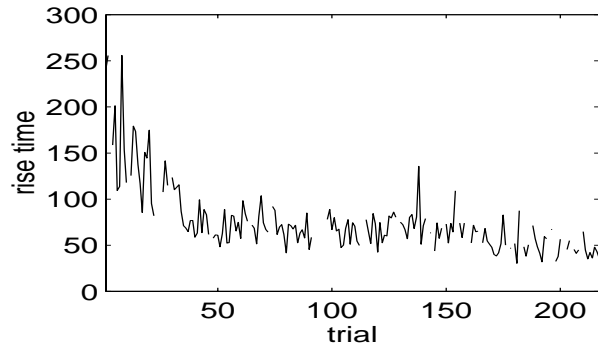
### Distance to the Goals



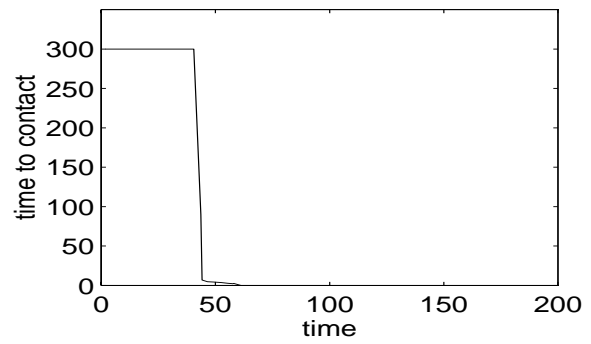
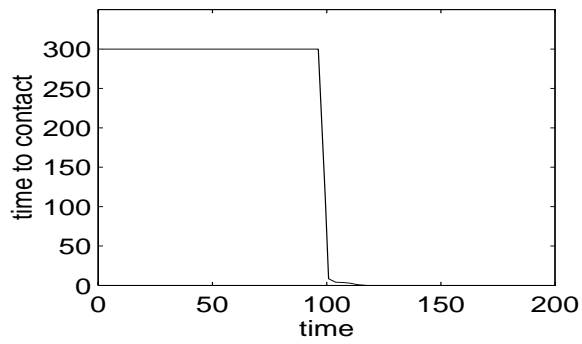
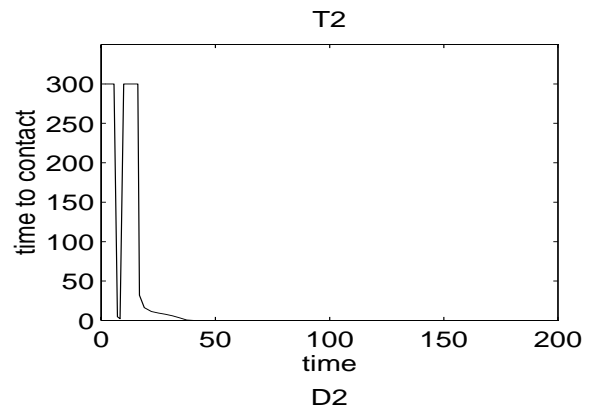
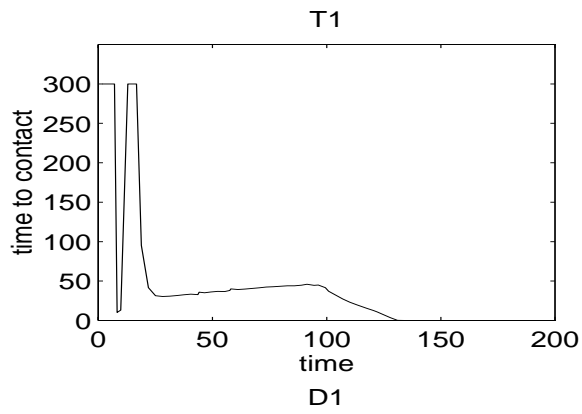
### Area under Distance to the Goals



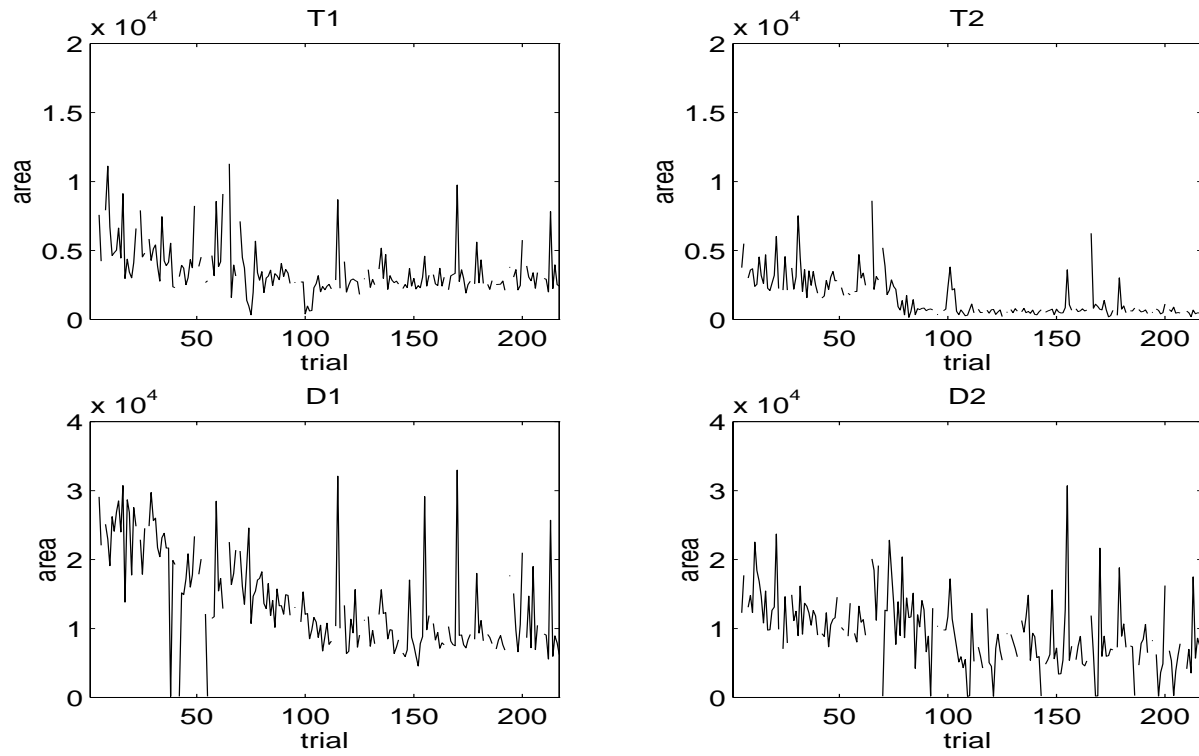
## Rise Time



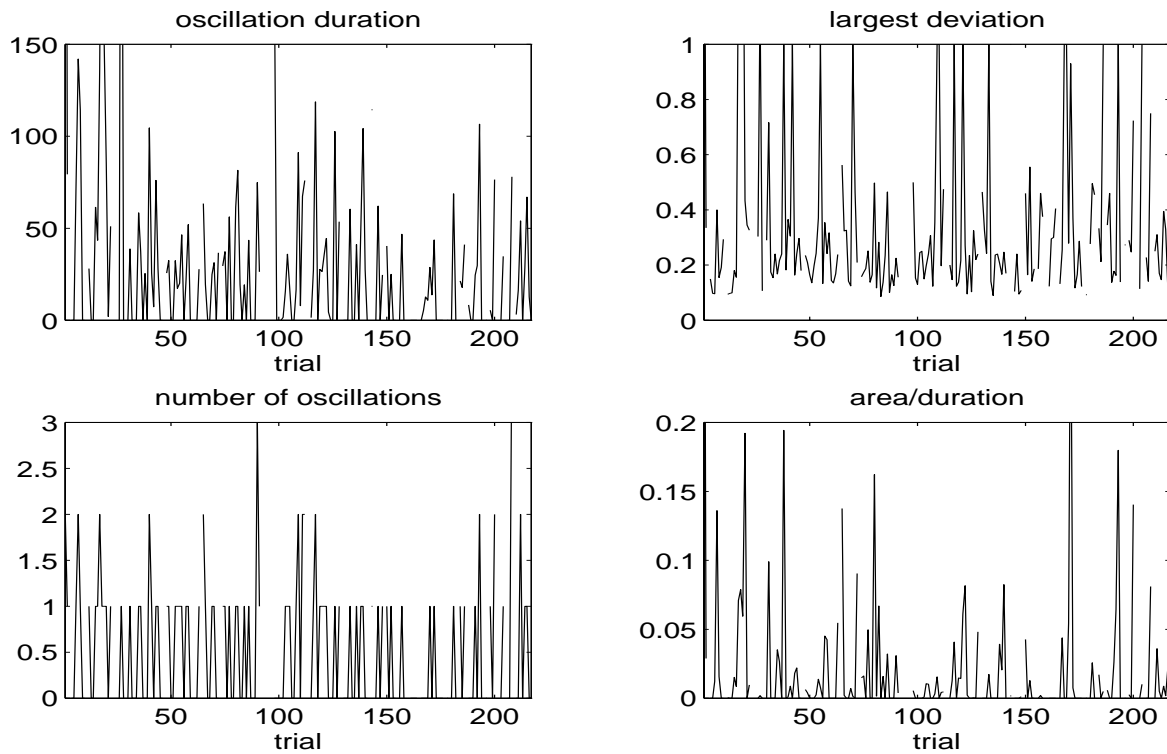
## Time to Contact the Goals



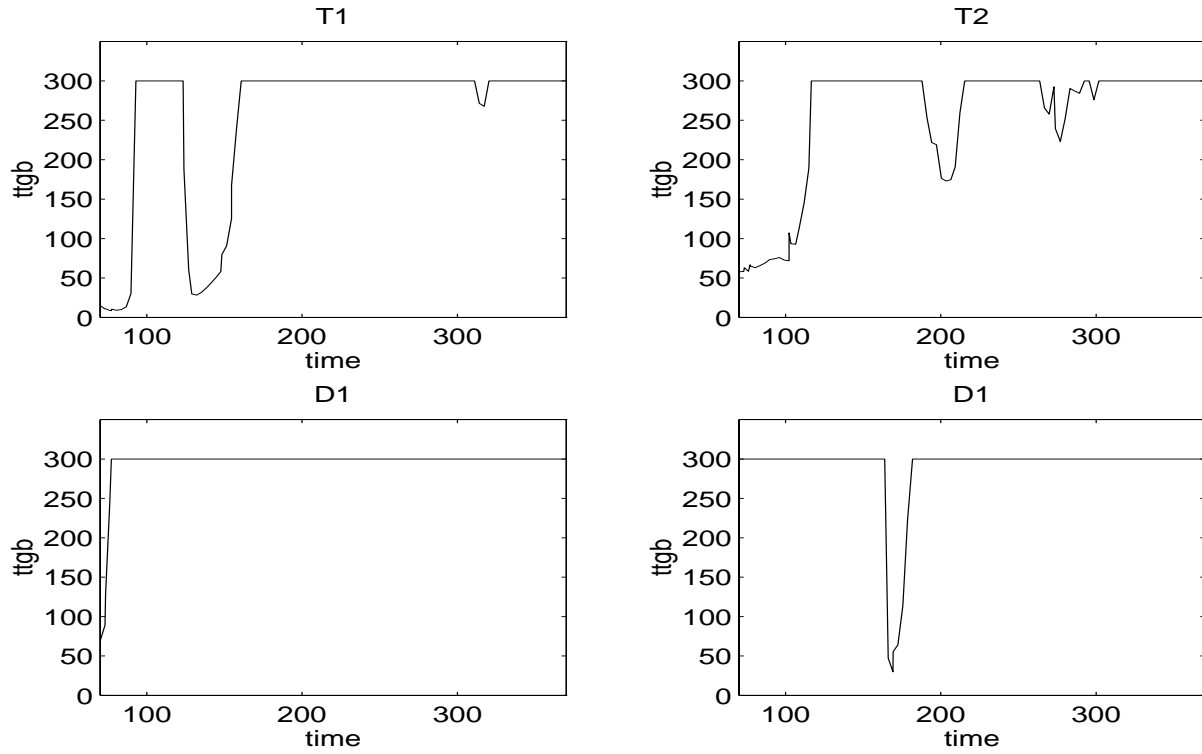
### Area under Time to Contact



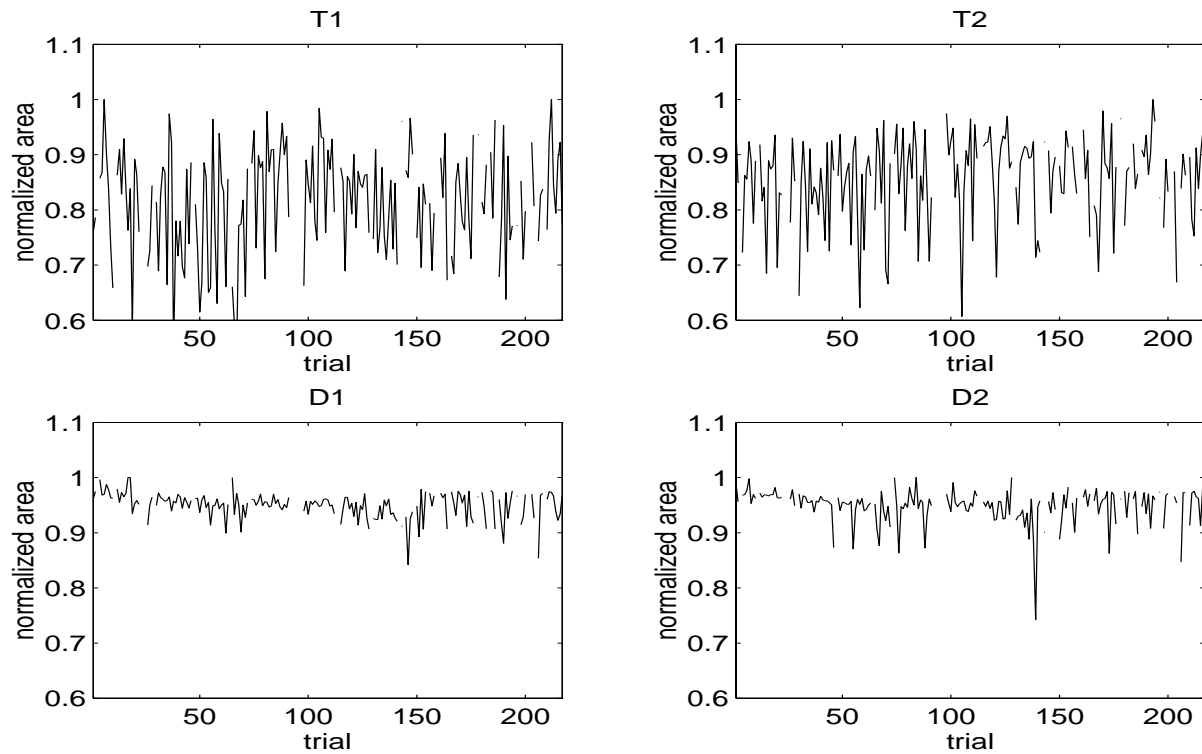
### Oscillation Measures



### Time to Goal Boundaries

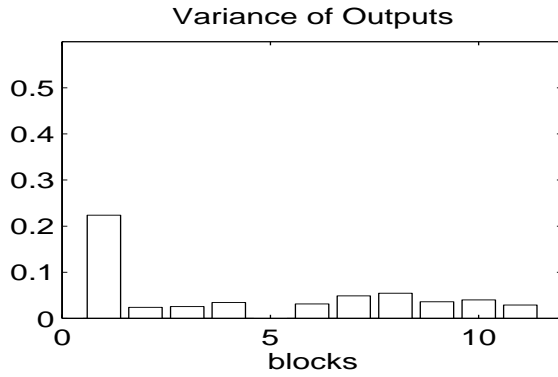


### Area under the Time to Goal Boundaries

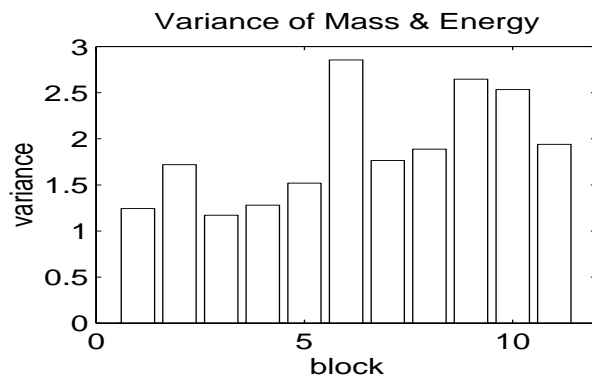




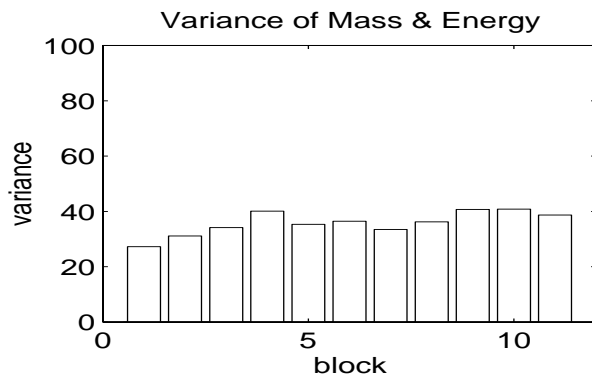
### Variance of Goal Variables (Outputs)



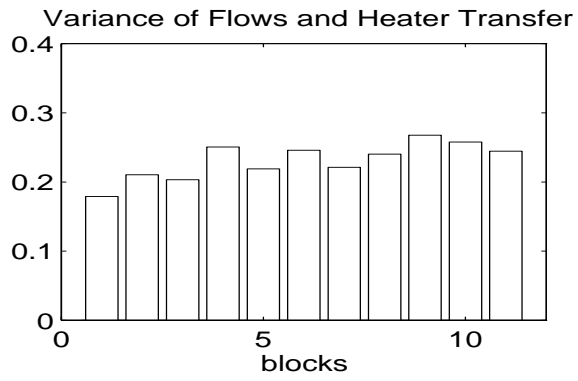
### Variance of Mass and Energy (Normalized by Goals and Scale)



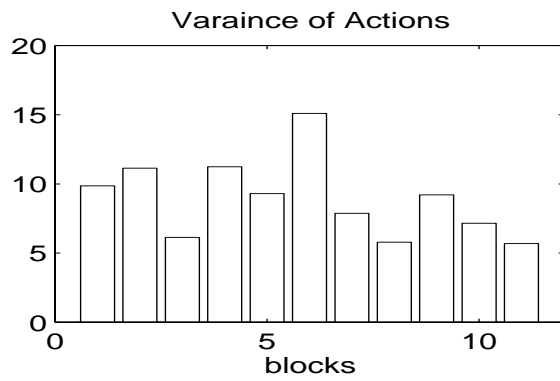
### Variance of Mass and Energy (Normalized by Scale Only)



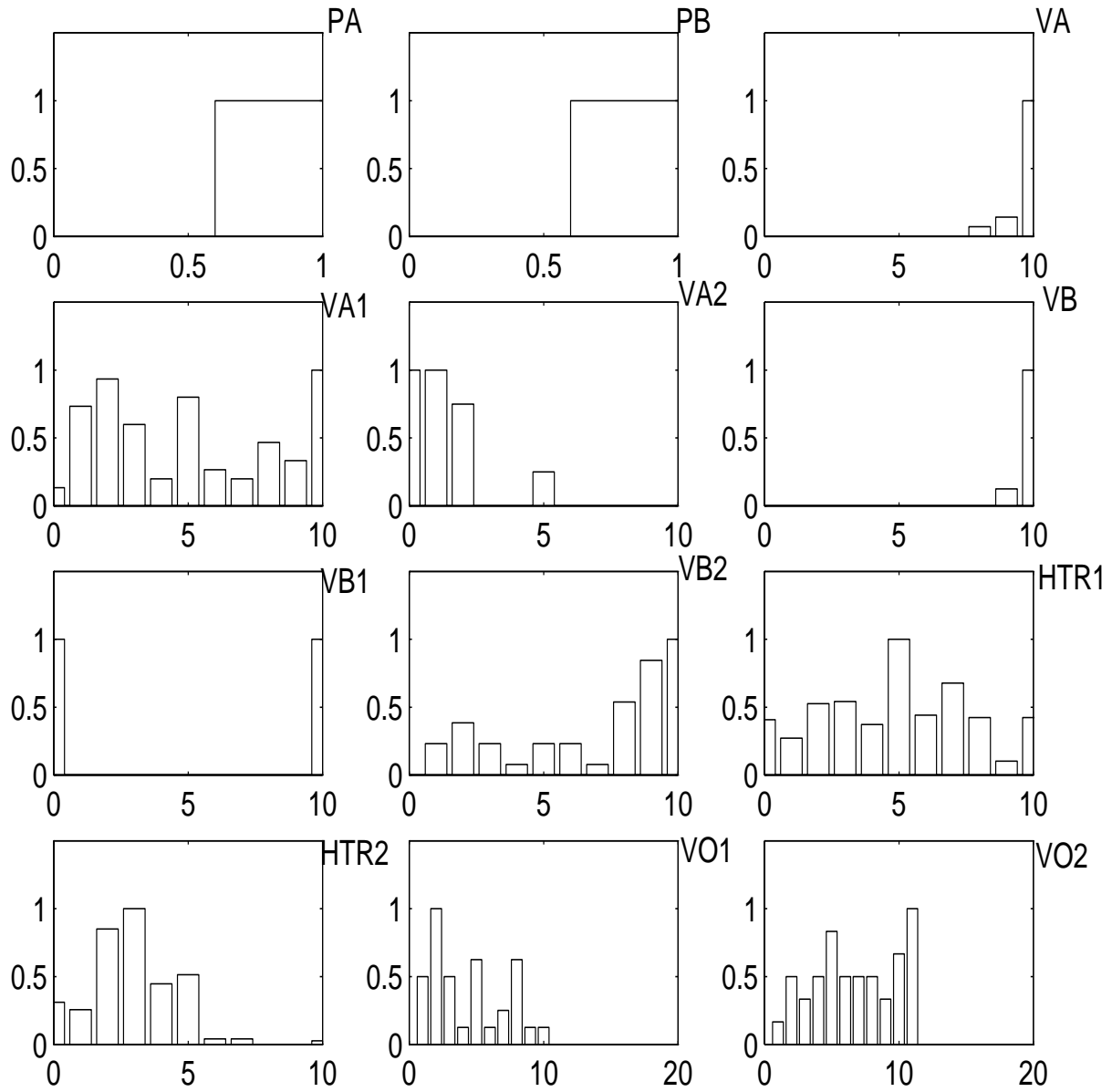
### Variance of Flows and Heater Transfer



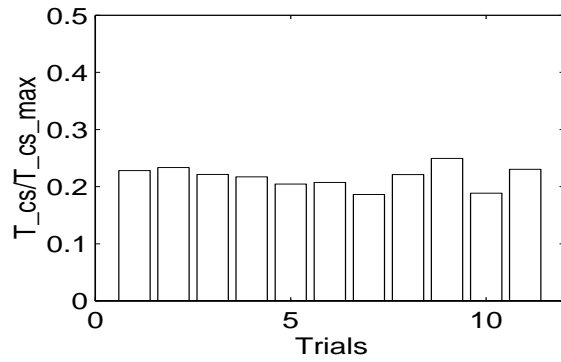
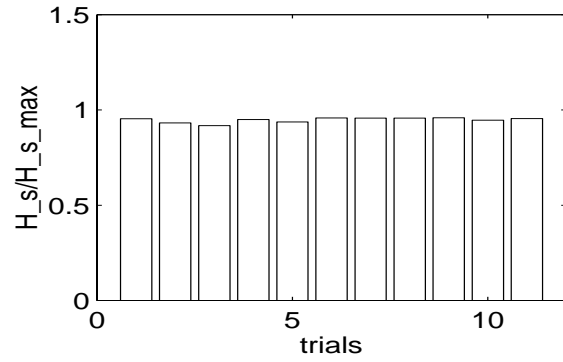
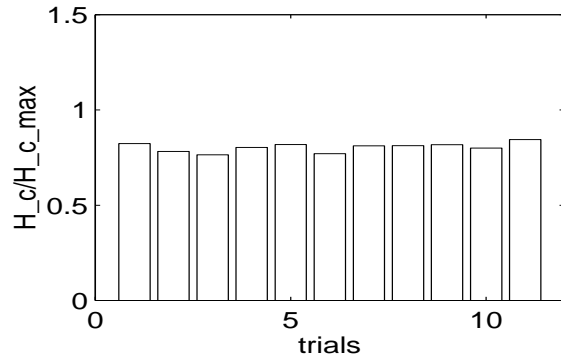
### Variance of Component Settings (Actions)



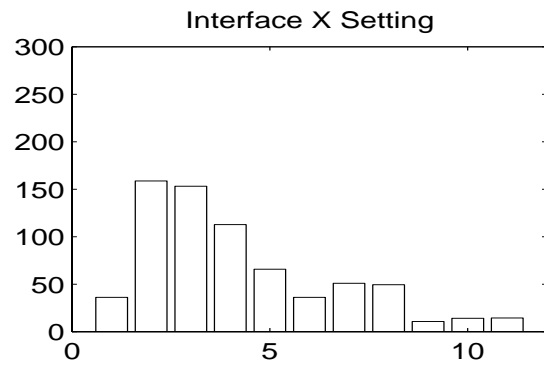
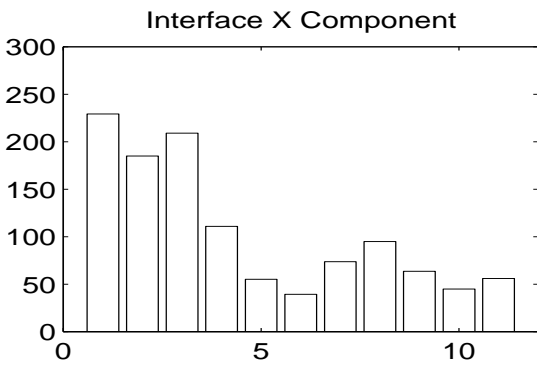
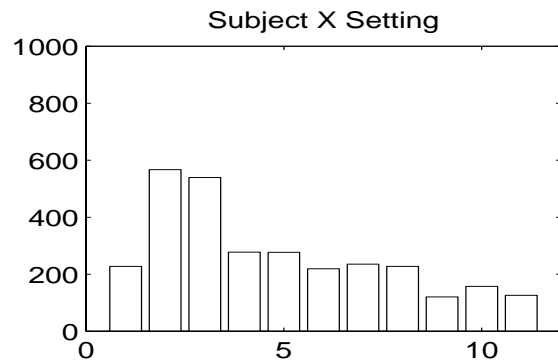
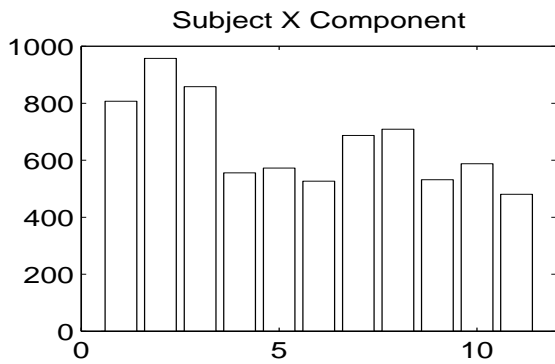
Actions Frequency Distribution



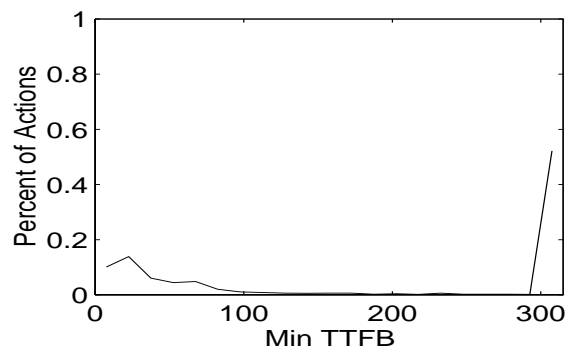
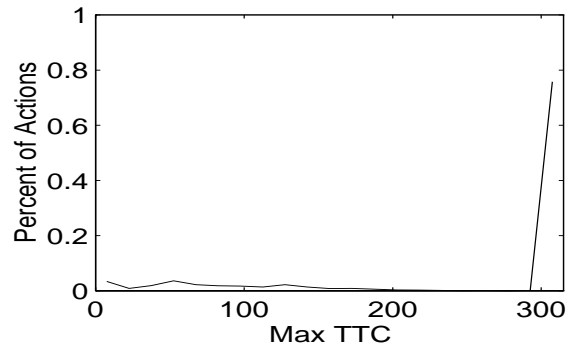
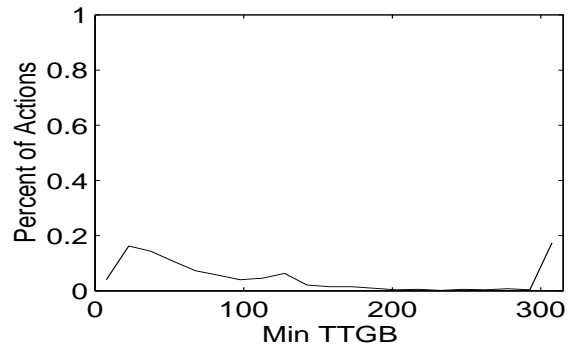
## Entropy



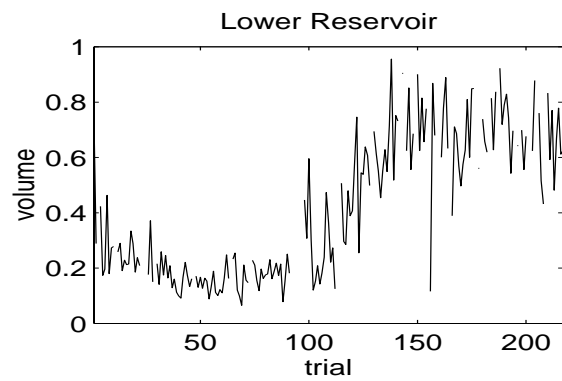
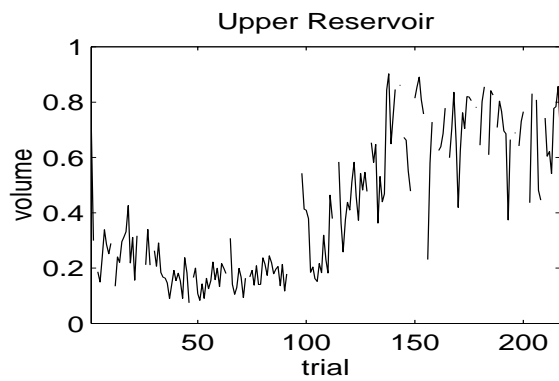
## Contingency Table



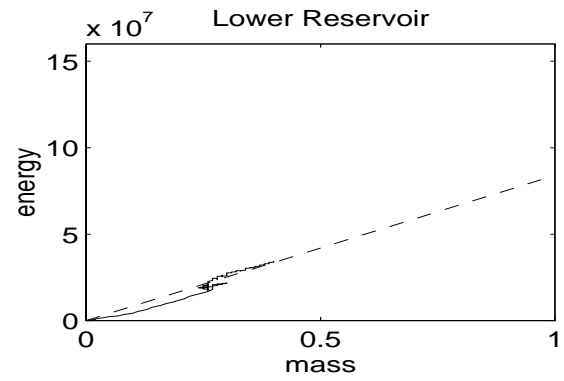
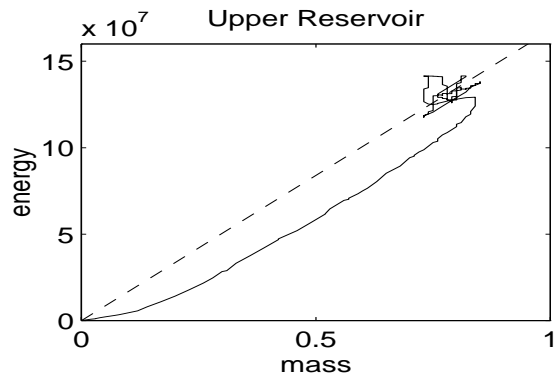
### Actions vs. TTGB, TTC, and TTFB



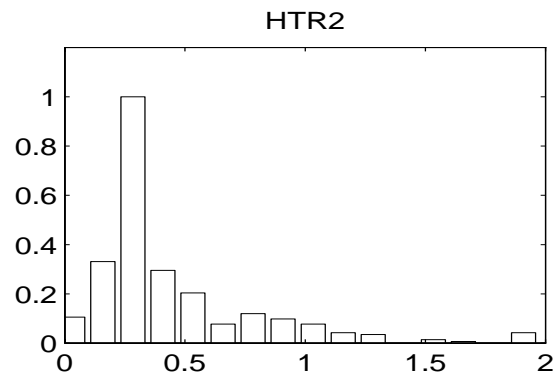
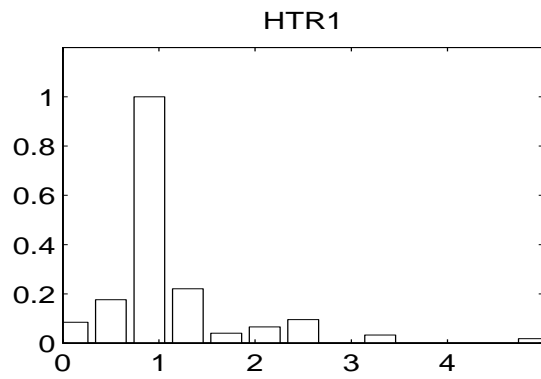
### Reservoir Volumes at Steady State



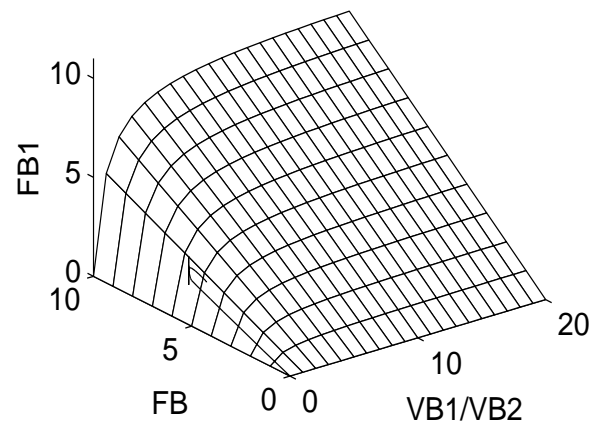
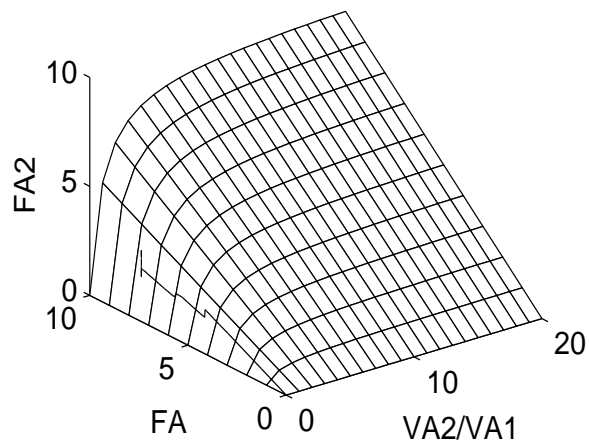
## Mass vs. Energy



## Actions on Heaters



## Constraint Surfaces



## Appendix 2: Variables Names and Descriptions

Variable Names	Descriptions
PA	PA setting
PB	PB setting
VA	VA setting
FA	VA flow rate
VA <sub>1</sub>	VA1 setting
FA <sub>1</sub>	VA1 flow rate
VA <sub>2</sub>	VA2 setting
FA <sub>2</sub>	VA2 flow rate
VB	VB setting
FB	VB flow rate
VB <sub>1</sub>	VB1 setting
FB <sub>1</sub>	VB1 flow rate
VB <sub>2</sub>	VB2 setting
FB <sub>2</sub>	VB2 flow rate
HTR <sub>1</sub>	HTR1 setting
FHTR <sub>1</sub>	HTR1 heat transfer rate
HTR <sub>2</sub>	HTR2 setting
FHTR <sub>2</sub>	HTR2 heat transfer rate
M1	Reservoir 1 mass level
T1	Reservoir 1 actual water temperature
MI1	Reservoir 1 mass inflow rate
MO1	Reservoir 1 mass outflow rate
D1	Reservoir 1 target mass outflow demand
EI1	Reservoir 1 energy inflow rate from water and heater
EO1	Reservoir 1 energy outflow rate
E1	Reservoir 1 energy level
T <sub>1goal</sub>	Reservoir 1 target water temperature
M2	Reservoir 2 mass level
T2	Reservoir 2 actual water temperature
MI2	Reservoir 2 mass inflow rate
MO2	Reservoir 2 mass outflow rate
D2	Reservoir 2 target mass outflow demand
EI2	Reservoir 2 energy inflow rate from water and heater
EO2	Reservoir 2 energy outflow rate
E2	Reservoir 2 energy level
T <sub>2goal</sub>	Reservoir 2 target water temperature