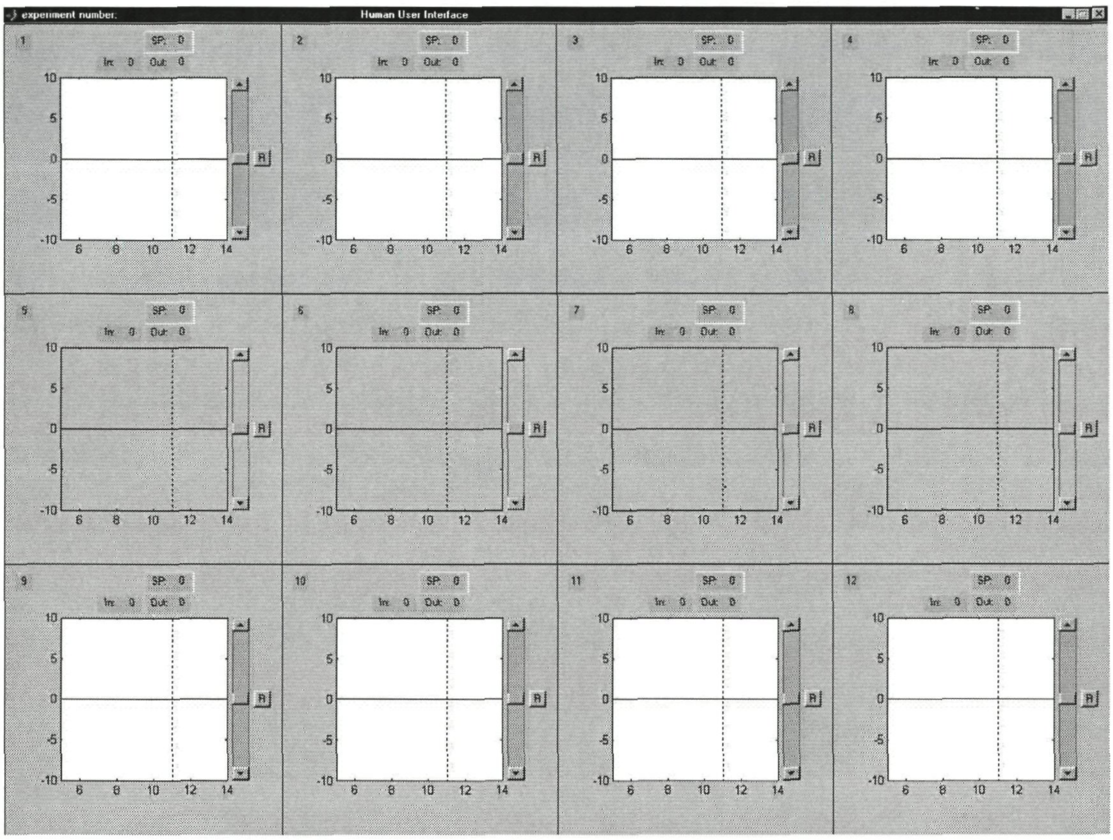
Interface-Complexity

And its influence on performance and workload



Graduation report by: Martijn Langelaar

Mentor University of Professional Education of Haarlem: Drs Ir C. van der Vlugt

Mentor Delft University of Technology: Prof. Dr. ir. P. A. Wieringa
Mentor Delft University of Technology: A. van der Horst

Devision: Man Machine Systems

Interface-Complexity

And its influence on performance and workload

Abstract

The objective of this research project on the field of human machine systems, especially human control of complex systems, is finding a relation between interface complexity and workload and a relation between interface complexity and performance.
Complexity is split into three parts: system complexity, task complexity and interface complexity. The border between task complexity and interface complexity is very narrow and only exists on higher abstraction levels.

A lot of information is available on interface design problems. It is concentrated on visual form or on content. No definition or measure of complexity has been found, so I didn't succeed in measuring interface complexity.

An experiment has been done. The experimental system consists of twelve coupled first order systems.
The subjects had to control the system using several different interfaces. I assumed different interfaces have different interface complexities. This is stated without being able to quantify this complexity. I looked for differences in human behavior, human performance and workload between the different interfaces, to see what the influence of changing the interface and interface complexity is. Unfortunately, no significant difference has been found, so no conclusions can be drawn about interface complexity.
As result of this experimental project, recommendations for future research on interface complexity are stated in this report.

Graduation Report

Company:	Delft University of Technology
Devision:	Man Machine Systems

Student name:	Martijn Langelaar
Student number:	00.2.00504
Graduation Period:	January 2002 – June 2002

Mentor University of Professional Education of Haarlem: Drs Ir C. van der Vlugt

Menor Delft University of Technology: Prof. Dr. ir. P. A. Wieringa
Menor Delft University of Technology: A. van der Horst

Issued : June 17th, 2002

Preface

During my graduation project at the Man Machine Systems Section I had my own small research project, quite a lot of freedom to make several choices. Sometimes it was hard to find information or to make a founded choice.
On several occasions discussions with several people were necessary to get insight in the material and to be able to make founded choices.

Several people were very supportive. In explanation of the MatLab experimental system, discussions on research-setup, helping me search for relevant articles, ordering those articles form several libraries, sending some emails to experts on the human machine interface field, helping me solve several problems with the computer (or should I say Microsoft). There were also people that studied the text I produced very critically and asked: “What do you actually mean with this word”, or suggested several solutions for some of my problems I confronted them with. And last but not least there were people who motivated me when I saw so much more problems than solutions.

The final conclusion of this research isn’t what I had hoped it would be. Nevertheless, I learnt a lot about doing research, working with people and about the field of Man Machine Systems. And I made some recommendations for similar research projects.

I’d like to thank al people who supported me with my research project in any way.

Delft, June 2002

Summary

This research project on the field of human machine systems, especially human control of complex systems has as objective to find a relation between interface complexity and workload and a relation between interface complexity and performance.
Complexity is split into three parts: system complexity, task complexity and interface complexity. The border between task complexity and interface complexity is very narrow and only exist on higher abstraction levels.

First a short literature review had to be made. Secondly a measure for interface complexity had to be developed for the available experimental system. Thirdly an experiment design had to be made. Finally the experiment is done, and the data had to be processed and interpreted, to come to some conclusions.

There is a lot of literature on interface design in general, and there is information about human behavior. The information about interface design can be split up into two categories, information on visual form and information on content. But no detailed information about interface complexity was found. An aesthetics measure was found, with the suggestion that it is very related to complexity.

Some information on complexity on general was found, this shows that there are no good definitions and measures for complexity, although there have been several attempts. Without a good definition of complexity or any ideas of possibilities for the measure of complexity, I didn't succeed in developing a satisfying measure of interface complexity for the available experimental system. However, some idea about complexity was formed.

An experiment has been done. The experimental system consists of twelve coupled first order systems. The output of each system is multiplied by 0,5 and added to the input of all the next systems. The operator has to control the first and last four cells, while the middle four cells are controlled automatically.

The subjects had to control the system using several different interfaces. I assumed different interfaces have different interface complexities. This is stated without being able to quantify this complexity. I looked for differences in human behavior, human performance and workload between the different interfaces, to see what the influence of changing the interface and interface complexity was. Unfortunately, no significant difference has been found, so no conclusions can be drawn about interface complexity.
I did make some recommendations for future research.

Table of Contents

Appendices

A. Complete text of Assignment; student goals and poster for poster session (all in Dutch)
B. Summary several articles
C. Ngo-like Complexity measure (SEE ALSO: REF 4 TO 8)
D. Calculations of Ngo-like complexity values
E. Question List used during experiments
F. Subject instructions
G. RSME form
H. Testers manual
I. MatLab files customized for this experiment
J. (MatLab/SimuLink) pictures of experimental system
K. Data sheets
L. General behavior plots
M. Script to create general behavior plots
N. MatLab scripts for data processing and plotting data
O. Figures for mean error, square error and RSME
P. Figures for performance vs workload, performance vs behavior and workload vs behavior
Q. Student t test [REF 17]

List of Symbols and abbreviations

symbol	description	[units]
α	unreliability (section 6.4)	[%]
α_1	angle between first mouse vector and a horizontal line (section 6.2.4)	[rad], [deg]
α_2	angle between second mouse vector and a horizontal line (section 6.2.4)	[rad], [deg]
β	angle between the two mouse vectors; mouse direction change (section 6.2.4)	[rad], [deg]
$x_i[k]$	output of cell i for the computing period k	[-]
i	cell number	[-]
M	total number of sampling periods	[-]
ME_i	mean error of cell i	[-]
N	total number of cells	[-]
SE	system error	[-]
SPF	system performance factor	[-]
t	testing quantity (in Dutch: “toetsingsgrootheid”) (Appendix Q)	[-]
t_{kr}	critical value of testing quantity (Appendix Q)	[-]
v	degrees of freedom (Appendix Q)	[-]
DURESS	Dual Reserervoir System	
EID	Ecological Interface Design	
KBB	Knowledge Based Behavior	
RBB	Rule Based Behavior	
RSME	Rating Scale for Mental Effort	
SBB	Skill Based Behavior	

1 Introduction

In this chapter, the background of the assignment of my graduation project is described shortly, after that, the assignment itself is explained, then something is told about the experiment that is done and finally there is an overview of what can be expected in this report.

1.1 Background

This report is the result of a graduation assignment (in order to finish the education) at the University of Professional Education in Haarlem, on the field of aerospace technology.

The assignment has been carried out at and for the Man Machine System group at the faculty of Design and Engineering, subfaculty Mechanical Engineering and Maritime technology.

One of the fields of interest at the Man Machine Systems groups is human supervisory control of complex systems. Modern technology creates more and more complex systems, and partly automates them. There are hardly any processes where no human operator is involved at all. In most cases there is a human involved in control and command, for example the steering of large ships, supervising chemical plants or nuclear power plants.
In these cases, there is an interaction between the human operator and the technology. There always is an interface between human and machine. This interface is the interest of this work.
The picture illustrates the position of the operator and the relations between operator and system for a manual controlled process and for an automated process with an operator present for supervisory control.

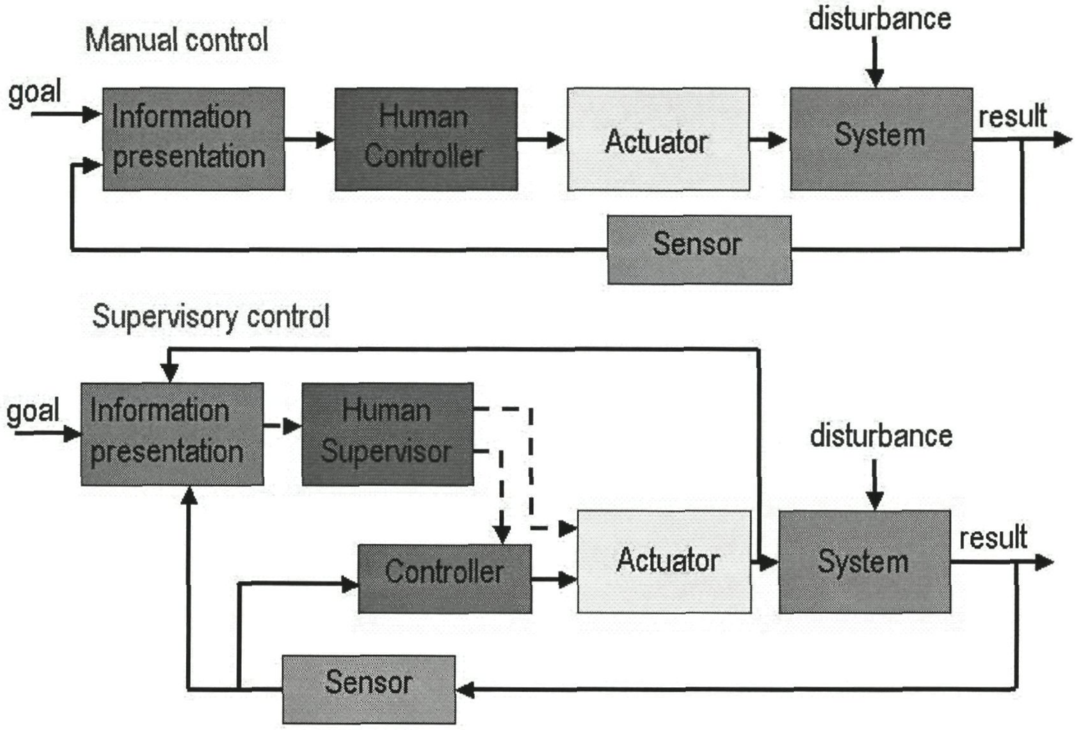


Figure 1: Manually and Supervisory controlled systems

For this research project, complexity is split into three parts: system-complexity, task-complexity and interface-complexity. Not everybody thinks it is possible or useful to do this, so this needs some consideration that is done in section 5.1.1.

1.2 Assignment

The topic of the assignment is "interface complexity" in relation to human performance and workload. The assignment is divided into three parts.
First an overview of (recent) relevant articles had to be made, to see what has been done in this field.
Second, a measurement/scale for interface complexity had to be developed for the interface of the available experimental system.
The third and last part is an experiment that has to be carried out to study the influence of interface complexity on human performance and workload.
The complete text of the assignment in Dutch is in Appendix A.
For this project, I have chosen to concentrate on the display, and not to investigate other aspects of the user interface like the hardware used for input (mouse, trackball, keyboard, special keyboard or control panel, etc).

The MMS group already has an existing experimental system for research in this field, based on the PhD thesis of Zhi-Gang Wei on degree of automation. [REF 16]. This system is a MatLab/Simulink (mathematical software) model with twelve cells. The relationships between these cells can easily be altered. It is also possible to control some cells automatically.

The operator's task is to keep the output of the cells at a requested setpoint value and sometimes alter this value if there is a setpoint request. The setpoint requests can be generated (semi) random. The operator has a slider bar for each cell, so he/she can influence the input of that cell.

In this experiment I hope to find a clear relation between the complexity of the interface and the performance, workload and possibly learning of the operator. Depending on the definition of interface-complexity, for a more complex process the importance of a simple (non-complex) interface is greater, but it might not always be possible to design a very simple interface for a very complex system.

1.3 Experiment

After some time, consideration and deliberation with several persons, the concrete choices for the experiment setup have been made. The experiment was carried out with the available experimental system. Cells 5-8 will be controlled automatically. The human behavior will be observed by counting the number of changes in the inputs of the manually controlled cells and by the calculation of the distance covered by the mouse-cursor. Some alterations to the experimental software are made to make this test possible.

1.4 Contents of this report

In this chapter, the assignment is described, and some very general and short background information is given.
Chapters two and three provide some more background information. Chapter two is about complexity. Unfortunately it isn't possible to give an exact scientific definition of complexity, and is also impossible to find a (universal) measure for complexity. The third chapter is about interface design and operator behavior.
In chapter four I discuss how complexity is handled during this experiment, which is necessary for defining an experiment set-up which is described in chapter five.
The results of the experiment are discussed in chapter six, some qualitative remarks are given, and the processing of the numerical test data is explained. This leads to the conclusions in chapter seven and the recommendations in chapter eight.

HOGESCHOOL HAARLEM
OPLEIDING LUCHTVAARTTECHNOLOGIE

BEOORDELING AFSTUDEEROPDRACHT

Naam: Martijn Langelaar | Datum:

Hogeschool/bedrijf/afdeling: TU Delft - fac. Werktuigb.k. | Afstudeerperiode: jan 2002 - jun 2002

Bedrijfsbegeleider(s): Prof dr ir PA Wieringa ; A. vd Horst | Examinator:

Omschrijving van de opdracht:
Interface Complexity and its influence on performance and workload.

(+++ zeer goed; ++ goed; + voldoende; o matig; - onvoldoende)

Beoordeling van functioneringsaspecten:

1. werkhouding : +++
2. zelfstandigheid : ++
3. initiatief : +
4. organisatorische vaardigheden : +
5. theoretisch inzicht : +
6. contactuele eigenschappen : ++
7. samenwerking met collega's : ++
8. samenwerking met begeleider : ++
9. kwaliteit tussenrapportage : +
10. kwaliteit van het werk : +

Toelichting bedrijfsbegeleider: Martijn was ondanks de lastige materie en de tegenvallende resultaten steeds weer te bewegen om door te gaan en verder te spotten.

Beoordeling van het afstudeerverslag:

11. indeling en opbouw : +
12. samenvatting, conclusies : +
13. niveau technische inhoud : n.v.t.
14. tabellen,figuren,grafieken : +
15. bijlagen,literatuurverwijzing : +
16. technische tekeningen : N.V.T.
17. softwaredocumentatie : +
18. stijl/leesbaarheid : +

Beoordeling presentaties:

19. presentatie op bedrijf :
20. postersessie/voordracht :
21. eindbespreking : +

Toelichting/aanvulling:

Het eindoordeel is samengevat in het toegekende cijfer (1-10): 8½

Opmerkingen van de student:

Bedrijfsbegeleider(s): | Externe deskundige: | Examinator: | Tweede docent:

Formulier AS-7

2 Complexity

In this chapter complexity is described, which is a very difficult task. This chapter is meant to give some background information and a better idea of complexity, and is NOT an attempt to give a definition. Several people have tried to give a definition and measure of complexity, but no one really succeeded. First a description of complexity is given, after that some descriptions of other ideas are given, which is necessary for the discussion in the later sections, in which the description is expanded and an historical overview of the field is given. Finally it is concluded that there is no single exact definition and no measure.

2.1 Description of complexity

Complexity is a very difficult (complex) idea. Although everybody knows the meaning of the word "complex", it is hard to explain what it means. The ATOMICA online dictionary says:

com·plex (kəm-plĕks', kŏm'plĕks') *adj*

1.
   a. Consisting of interconnected or interwoven parts; composite.
   b. Composed of two or more units: *a complex carbohydrate*

2.
   Involved or intricate, as in structure; complicated.

This is a nice try, but not a very good scientific definition, which is hard to give. A definition that makes it possible to quantify the complexity of an object, thought or system is even harder to give.

A lot of people have tried to do so, but nobody really succeeded.
One could say that complex is that what isn't of all days, but maybe it is better to state that we don't always realize how complex everyday life is and that we only notice the complexity of anything out of the ordinary. Everybody has an intuitive idea about complexity, but is still hard to describe.

2.1.1 Descriptions of some other ideas

The words or ideas below are used in the discussion about complexity. It is necessary that their meaning is clear to the reader, so below a description is given to explain what these words mean. Again, the ATOMICA online dictionary was used.

chaos	condition of great disorder or confusion; unformed matter; lack of order or regular arrangement; apparently random behavior
data	raw facts and figures; values derived form scientific experiments; factual information
dull	intellectually weak; stupid; not sharp or keen
entropy	(quantative measure of) disorder or randomness in a closed system (bearing energy or information); the tendency for all

	matter (and energy) in the universe to evolve toward a state of inert uniformity
evolution	a theory first proposed in the nineteenth century by Darwin, according to which the earth's species have changed and diversified through time under the influence of natural selection; a gradual process in which something (e.g. the genetic composition of a population) changes into a different and usually more complex or better form; gradual development
information	knowledge (of specific events or situations), derived from study, experience, instruction, perception, discovery (or taken for granted); a collection of facts or data; that which is known; that which makes you understand the world around you better
information contents	the rate to which information answers the question asked; rate to which the information relates directly to the considered subject; quality of the processing and/or selection of data; value of information, considered the question asked
amount of information	measure of the size of (unselected and unprocessed) information or data, measured e.g. by the space storage takes, or the time necessary for transfer
interesting	absorbing; arousing; holding the attention
nonsense	something that doesn't have or make sense; words or signs having no intelligible meaning; matter of little or no importance of usefulness; foolish or absurd;
order	(logical, comprehensible, methodical or prescribed) arrangement; sequence of successive things; a way in which things follow each other
pattern	plan, diagram or model; (component in a) composition, arrangement or design
random	having no specific pattern, purpose, organization, structure or objective; of or relating to an event that is described by a probability distribution (in which all outcomes are equally likely)
rhythm	regular or harmonious pattern; similar but less formal sequence; sense of temporal development
rules	standard method, conditions and/or procedure; usual, customary, or generalized statement, course of action or behavior
thermodynamics	branch of physical science that deals with the relationships and conversions between heat and other forms of energy (mechanical, electric and chemical energy)

2.2 Between order and chaos

Complexity is the area between complete chaos and complete order. Both extremities aren't very interesting. Something we see as a complete ordered system is very easy to describe, has a very short description and we think it is very dull. Something we experience as a complete chaos, is harder to describe but isn't always interesting because there is no regularity/pattern in the information, so we don't know what to do with it.
The interesting part is the area in between. It challenges us to find the pattern, rhythm or rules behind the (partial) chaos. Figure 2 on page 14 illustrates this.

Behind a lot of processes that seem very complex at the surface, there are sometimes some simple rules. It might be necessary to do a lot of calculations or iterations to get from these simple rules to the complex reality. It isn't possible to predict the results by reviewing the simple rules/ formulas, you have to carry out the calculations. The computer made these calculations possible.
To discover complexity, it's necessary to navigate between simple rules and formulas and the complex real world.

2.3 Example Definitions

A lot of researchers have tried to define and measure chaos. Some examples are derived from [REF 9]:

2.3.1 Kolmogorov-Complexity

The Kolmogorov-Complexity has been the only accepted measure for complexity for quite some time. In the sixties Kolmogorov suggested that complexity could be measured by examining the shortest description of a system. For expample, the series {1,1,1,1,1} can be describes as: five times one; {a,b,c,d,e,f,g} are the first letters of the alphabet; but {2,7,4,2,9,5,8} is a random series of numbers which can not be described any shorter than the series itself. (Of course some properties of the series, such as mean value and standard deviation can be calculated.) According to this, complexity is equal to arbitrariness. This isn't true; for example a musical composition is much more complex than some random notes, because a composition is thought to be esthetical by a large group of individuals in our socio-cultural society. There is no group that thinks the randomly played notes by a monkey are a musical masterpiece.

2.3.2 Between order and chaos

The idea mentioned before in section 2.2, that complexity is between order and chaos was coverd by from Huberman and Hogg (1985). Later they discovered the work of Simon, who did work in the field of artificial intelligence, which said the same.

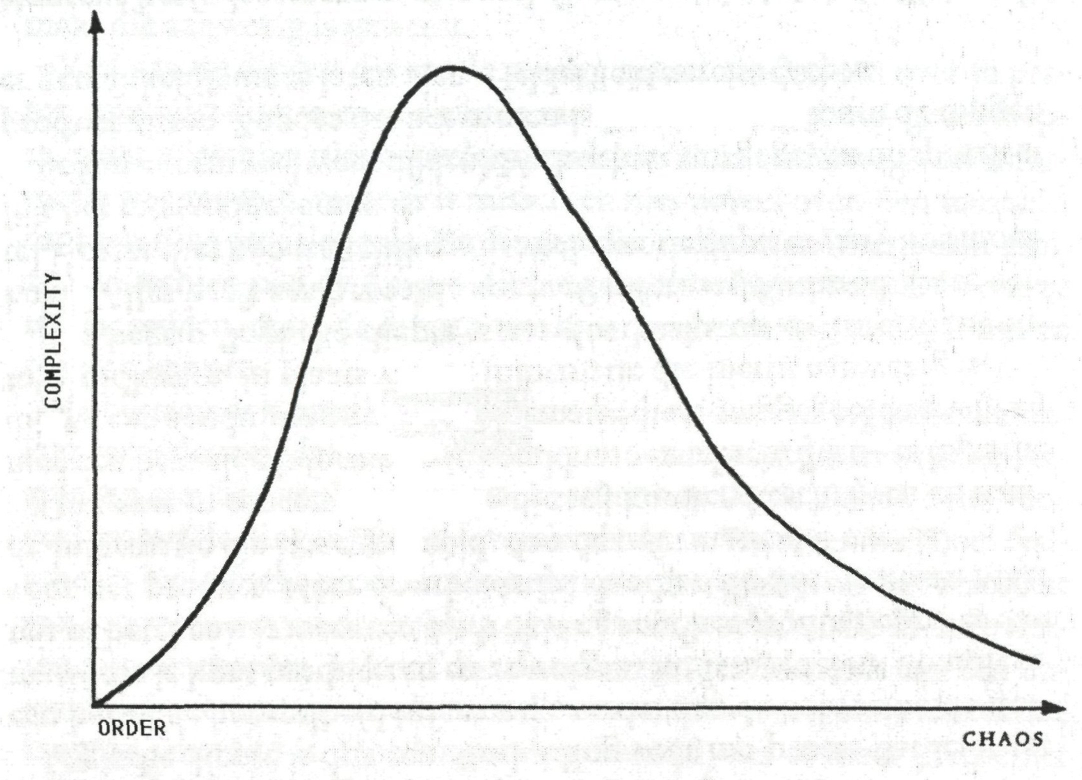


Figure 2: Complexity is between order and chaos [REF 9]

2.3.3 Logical depth

A great breakthrough came in 1986 when Grassberger connected the idea of "logical depth" to the complexity-theory. Logical Depth was described by C.H. Bennett. Bennett stated that complexity is similar to information contents, the effort that is needed to produce information, the calculation time both in the brain or in a computer. The length or amount of the information isn't relevant.
Logical depth can be defined informally as the number of steps necessary to draw a conclusion. Or as the number of links in a chain of causes that connects an object to its roots.
A more exact definition: Logical depth is the designation of the process which leads to a certain amount of information that can be produced and communicated. Complexity or Meaning is more related to the production process than to the product. Or maybe more related to the amount of information that is thrown away during the production process then to the amount of information that is left over.
This definition is more fitting than the Kolmogorov-Complexity. Nonsense is easy to create, and has very little depth, so a musical composition has a lot more logical depth than random notes. Because the creative process of composing has many more steps than randomly selecting notes.

It is very hard to calculate logical depth, because not everything can be calculated or simulated (yet). And if things can be calculated in several ways, the logical depth should probably be calculated using the shortest way. However, it is hard to make sure that the way you have chosen is the shortest way possible.

2.3.4 Thermodynamic depth

In 1998 Lord and Pagles developed a theory similar to the logical depth theory. They called their theory "the theory of thermodynamic depth." Their idea was to define complexity as the amount of information that is disposed during the creation or evolution of a physical object. The problems are the same as with logical depth: we don't always now the (evolutionary) history.
The name 'thermodynamic depth' stems from parallels to the thermodynamics, especially entropy. Chemical or physical equilibriums evolve in a complex process with simple rules. A system is complex as it is in equilibrium with its surroundings. There is no nett interaction between the system and the surroundings. The system is stable. The amount of time, and the number of steps that are necessary to achieve this equilibrium are a measure for complexity.

2.4 Conclusion

No satisfactionary way is found to define the idea complexity. Therefore it is very hard to vary the interface complexity in a controlled manner. The chosen solution is to alter the interface, and state that different interfaces have different complexities. The differences between the reactions to the several interfaces are interpreted as due to the change of interface and interface-complexity.

3 Interface-Design and Operator Behavior

This chapter provides some background information on interface design. Interfaces are designed for operators, so there also is some information on operator behavior. First a definition of interface will be given. After that, the importance of the availability of a well-designed interface is explained. Third, something is explained about operator behavior. Finally there is some information on other studies of interfaces. This information is divided into two groups: first studies that concentrate on the visual form, and secondly studies that concentrate on the content. Both issues need to be considered during the design process of an interface.

3.1 Definition of Interface

People will always be in some interaction with their surroundings. Modern technology makes it necessary and possible to interact with the surroundings in a complicated way. People are no longer necessarily in direct contact with the objects they are interacting with. Thes objects can be very small or large, very dangerous or otherwise hard to handle directly, for instance the control of a nuclear power plant or a chemical process. In many cases, some kind of interface is used between the actual desired process (which can be defined in several abstraction levels) and the person using it. The definition of the borders of the interface system is arbitrary. Many options are possible. In this study, the interface consists of the display and input devices, as well as a processing unit. For the experimental setup it isn't important whether the sensors and actuators are seen as part of the interface or part of the system.

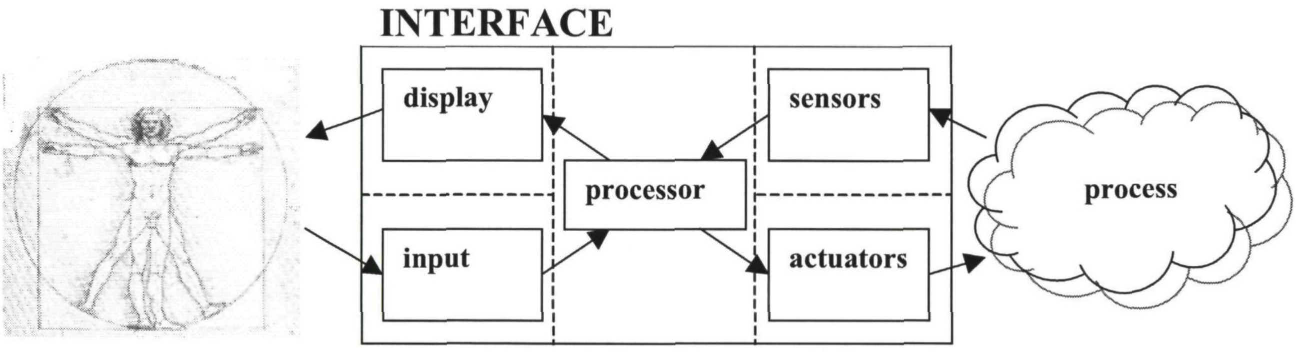


Figure 3: Definition of Interface

3.2 Importance

The interface is an important part of interaction between human and the surroundings. The interface will have to support the human operator in performing his tasks (the task can be for example the production of electricity, mayonnaise or matches). The product that is delivered will have to be of a certain quality; also a certain amount may have to be produced. These factors are the operator's performance.
The performance can also be measured by the (mean) error of the process parameters or by the search time necessary to find certain information.

A well-designed interface improves operator performance and reduces the workload. Also accidents can be prevented by a well-designed interface.
The operator has to accept the interface, and have confidence it presents relevant information. As this has been realized, the operator will be satisfied, and will be more motivated to learn how to use the interface.
The interface design should concentrate on the interaction between the operator and the process rather than the interaction between operator and interface. The main questions in interface design and their relations to the work domain and the operator are illustrated in the figure 4.

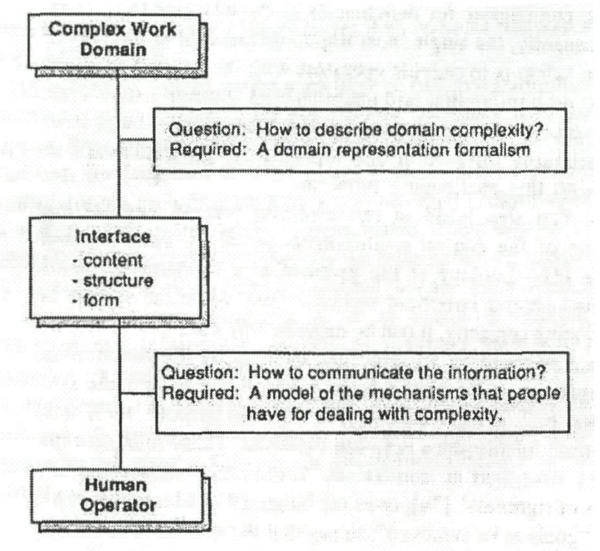


Figure 4: The structure of the interface design problem [REF 15]

3.3 Operator behavior

In order to design an interface for an operator, it is very useful to know something about the behavior of the operator. In general, in human (operator) behavior, three categories can be distinguished.
First there is **skill-based behavior** (SBB). This occurs at common tasks. The operator has carried out this task very often, and has enough experience to perform this task. He hardly thinks about it. Riding a bike is a good example of skill-based behavior. Someone who has experience in riding a bike does this unconsciously and without falling over.
Secondly, there is **rule-based behavior** (RBB). The operator has no skill or experience to handle some situations, but the events are anticipated. These *anticipated events* (and/or their image on the interface/display) act as cues that trigger certain rules. The operator has to perform certain cognitive activity here. This kind of behavior can be compared to traffic regulations. Everybody knows you have to wait for a red light, and no one has to think very hard about it.

(Cognitive means: related to mental processes such as knowing, perception, awareness and reasoning)

Third there is **knowledge-based behavior** (KBB). This is the trickiest one, as it is necessary to cope with *unanticipated events*. Most system designers try to avoid the occurrence of unanticipated events, but this can never be guaranteed. The operator has to decide what to do using his general system knowledge, without being able to rely on skills or rules. This requires great amount of cognitive activity. To complete the examples, knowledge-based behavior is like finding your way. You do have map (on paper, electronically or in your mind) but have to think to find the fastest, nicest or easiest route from A to B.

There are five general tasks or roles an operator has to perform.

1) planning
2) teaching/programming
3) monitoring
4) intervening
5) learning

Most processes require all of these roles on different abstraction levels. These tasks should be kept in mind while designing an interface.

Processes can be categorized in three categories

1) processes only some operators consider difficult
2) processes that are difficult by general opinion
3) processes that only few operators can perform

Whether an operator can control a process has to do with the operator's capabilities, but also with the design of the system and its user interface. The effort an operator is willing to make can also be an important factor.

3.4 Studying interfaces

There are generally two ways to study interfaces: concentrating on the visual form or concentrating on the content. These two things are interwoven and both points of view are necessary to create a good interface. The connecting factor is the structure, or the relationships between different parts. How are the parts related, and in how can we present these relationships clearly?

3.4.1 Visual form

The visual form can be described in terms of various factors. We can fit most of these criteria in five categories:

1) physical techniques
2) composition techniques
3) association techniques
4) ordering techniques
5) photographic techniques

In the table, all the factors are mentioned and placed in their respective categories.

The numbers between brackets refer to the references.

	physical	Composition	association	order	photographic
[5]	equilibrium	Density (local, global [10]) homogenity			
[5],[13]	balance symmetry proportion	Economy Simplicity	unity	sequence	
[13]	regularity alignment horizontality	Understatement Neutrality Singularity Positivity Transparency	repartition grouping [10],[13] (chunks [3]) sparing	consistency predictability continuity	sharpness roundness stability leveling activeness subtlety representation realism fatness
			organization [4]		

Table 1:Factors in interface design, aesthetics and complexity

Some factors don't fit in the categories:

- language [4]
- cohesion [5]
- rhytm [5]
- order and complexity [5]
- size [10]
- order [3]

3.4.2 Content

The work of Rasmussen and Vicente [REF 14 AND 15] concentrates mainly on the content. To decide scientifically what information has to be presented, the system is analysed in the **means-end** hierarchy; this is a system with abstraction levels, which are

1) goal oriented
2) functional purpose
3) abstract function
4) generalized function
5) physical function
6) physical form

The goals in one level are reached with the means of the lower level.

These levels can be illustrated with the Dual Reservoir System Simulation used by Rasmussen and Vicente. This is a two-reservoir water supply system. From both reservoirs there is a certain demand of water, and there is a desired temperature. The reservoirs are filled by a cross-feed system.

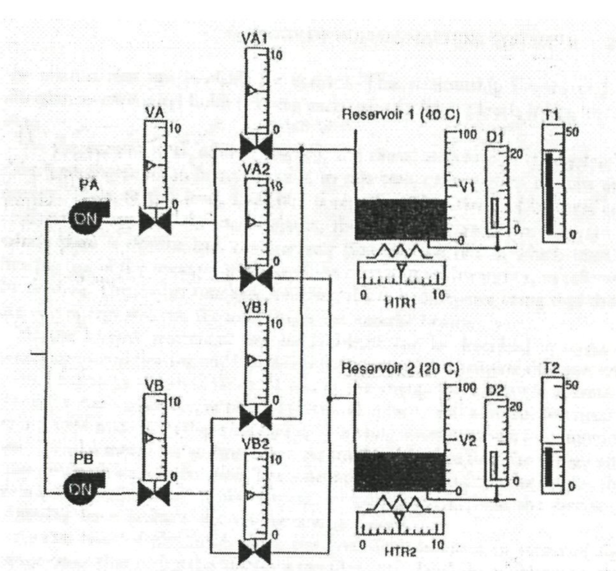


Figure 5: The DURESS system [REF 14]

The goal of this system, level one of the means-end hierarchy, is water supply. On the second level, the functional purpose is twofold, first the water demand has to be satisfied, and secondly the water has to be kept at the desired temperature at the same time.
At the abstract-function level, a mass-balance and an energy-balance can be defined to examine the behavior of both purposes. At the generalized function level, a flow-diagram is scheduled to be able to combine the mass or flow demand and the temperature demand. The physical function level represents the actual reservoirs and heaters, which are defined in a more detailed way in the physical form level.

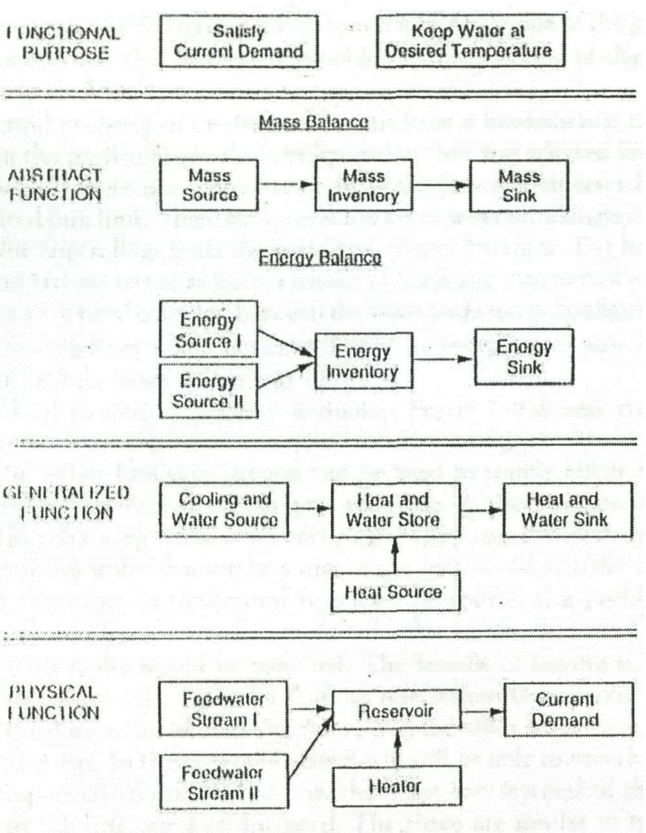


Figure 6: Means-End hierarchy representation of DURESS [REF 14]

Rasmussen and Vicente developed the theory of Ecological Interface Design. In this theory they state that if the internal representation of the system that most operators have is presented externally on the display, the cognitive work is reduced, mainly on the side of knowledge-based behavior. It is important to visualize the relations between the levels in the abstraction-hierarchy and between the several parts of the system. If the interface shows a complete overview of the system, and also shows these relations, it supports the operator in a great way.

4 Measure for interface complexity

In this chapter the problem of measuring interface complexity is discussed. First the problem is stated: for this research project, a well quantified measure for interface complexity is desirable. Unfortunately there is no such measure available. What is available is a (not yet fully quantified) measure for aesthetics. This is altered to an improvised, measure for visual-form-complexity. For the content a definition for interface complexity is presented, but there is no measure for this definition. Finally, the solution is to use different interfaces, and state they are NOT equally complex. The differences in human behavior, performance and workload can be interpreted as due to differences in interface-complexity.

4.1 Problem: measure complexity

The research question for this paper is to relate interface complexity to operator performance and to workload.
In the second chapter, several definitions for complexity are proposed and rejected. Most of them have to do with a process, and therefore might be usable for knowledge based behavior, to describe the cognitive process, but for skill based behavior and rule based behavior, where is no process involved, they are far less applicable.
The definition "complexity is between order and chaos" seems the best one, but it isn't quantified and therefore useless. However, the complexity of a system or process is something that has to be dealt with by the designer of a machine, and/or by its operator. In this paper, I want to look at the complexity of the interface. The complexity theories in chapter two are not applicable, so something else has to be done.

4.2 Visual form

The work of Ngo contains a measure for "aesthetics" that is, according to the work, also highly related to complexity [REF 4-8]. Unfortunately, there are several problems. First, not all of the suggested calculations can be made for the used experimental system. Second, the measure suggests that all the factors have a linear correlation with aesthetics and therefore complexity. Third all factors are considered equally important, which may not be the case.
The measure for aesthetics of Ngo has a scale from 0 (not aesthetic) to 1 (very aesthetic) and is usually calculated in 4 decimals. This measure is transferred to a complexity measure that looks like a percentage. The scale now has a range from 0 (not complex) to 100 (very complex). This is calculated in 2 significant numbers. The smaller amount of significant numbers is more realistic. But still the measure isn't very accurate, and doesn't make any difference if not the visual form, but the contents changes.
However, it is nice to see if there is a difference between the standard interface of the system and the interface created for this experiment. The transferred measure is in Appendix C

4.3 Content

If we look at the work on the field of Ecological Interface Design [REF 14 AND 15], the interface complexity is inversely proportional to the extent to which the interface support the operator in performing his task. This is something we can't quantify unambiguously.

Another option is looking how satisfactory the answers to the main questions on interface design as mentioned in section 0 are. This is hard to quantify as well.
We can look at the operator performance and workload for several interfaces at the same system configuration, but there is no exact measure.

$$\text{Interface complexity} = \frac{1}{\text{the rate in which the interface supports the operator in performing his task}}$$

Figure 7: definition of interface complexity by IR MAX MULDER

4.4 Solution

The research question for this paper is to relate interface complexity to operator performance and to workload. Untill now, no satisfactory measure for interface complexity has been found.
Therefore it is very hard to vary the interface complexity in a controlled manner. The chosen solution is to change the interface, and assume that different interfaces have different complexities. The differences between the reactions to the several interfaces are interpreted as due to the change of interface and interface-complexity.

The proposal made here is to observe the operator behavior. The experimental system logs a lot of data. Among this data is the location of the mouse cursor. With this data we can calculate the distance the mouse has covered, and calculate the changes of direction the mouse has made. It is also possible to look at the number of control-actions by counting the number of changes of the input variables.
The idea is that with simple interfaces, a steady, calm and straightforward process control is possible with little mouse movement and control actions. If the mouse is moved more chaotically and more control actions are taken by the operator, the interface is apparently more complex. Figure 8 shows that I am not able to measure complexity, but hope to be able to conclude something about it by studying quantities that we suspect are related to complexity.

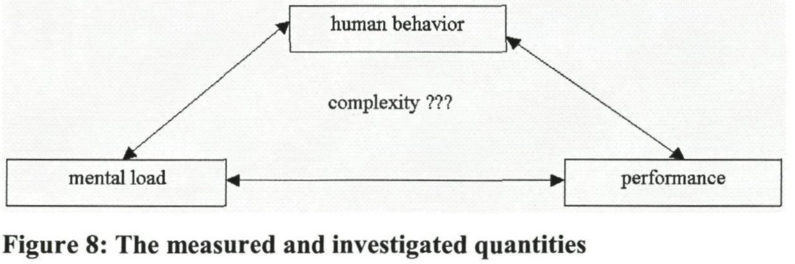


Figure 8: The measured and investigated quantities

5 Experiment set-up

In this chapter, the set-up of the experiment is explained. First the objective of this experiment is stated, and the setup is very generally discussed. Explained are the variation of the interface-complexity, the variation in interface, and the measure for mental load and for performance.
After that, the experiment is described in more detail. There is a description of the experimental system and the choice of (semi-random) setpoint requests. Furthermore, the used interfaces are described in detail. After that the sequence of experiments is explained, a description of a test run is given and the choice of test subjects is mentioned.

5.1 Purpose and general experiment

The purpose of the experiment is to investigate the influence of the interface- or display complexity on the workload and performance of the operator. To investigate only the influence of interface complexity, the system-complexity and the task-complexity have to remain constant.

5.1.1 Only vary interface-complexity

The system-complexity can be kept constant very easily by always using the same system. This can be done without calculating the complexity, because a constant system has a constant complexity.

However, keeping the task- complexity constant is somewhat more difficult, because the task is strongly related to the interface. If the interface changes, the operator-task will often change too, because an operator will have to perform other activities or more or less cognitive activities.
Here is a potential problem. Several experts think it isn't possible to think of task-complexity and interface complexity as two separate, independent concepts [IR. MAX MULDER, DELFT UNIVERSITY OF TECHNOLOGY, FACULTY OF AEROSPACE ENGINEERING, CONTROL AND SIMULATION GROUP ; DR. C.J. OVERBEEKE, DELFT UNIVERSITY OF TECHNOLOGY, INDUSTRIAL DESIGN ENGINEERING, ASSOCIATE PROFESSOR FORM THEORY, JOHN FLACH, OHIO STATE UNIVERSITY, PROFESSOR OF PSYCHOLOGY, ADJUNCT RESEARCH SCIENTIST, HUMAN-MACHINE SYSTEMS.] The one person who does think it possible makes the remark that it's possible, but doesn't make sense (IR HARRIE BOHNEN, DUCH AEROSPACE LABORATORY, MAN MACHINE INTEGRATION (VE)).
At this point, I do basically the same as Rasmussen and Vicente: look at the system at several abstraction levels. At a very high level, I define the task as: " The operator has to make sure the outputs of all cells meet the requested (setpoint) value". This definition is independent of the interface. So if we change the interface, the task doesn't change, at least not at this level of abstraction. If the task doesn't change, the task complexity is also constant. The effects that are measured are only caused by changes in the interface.

5.1.2 Variation in interface

The task the operator has to perform is defined on a very high abstraction level as: " The operator has to make sure the outputs of all cells meet the requested (setpoint) value". With this, we can vary the interface and state that all differences that occur are due to the difference of the interface.
I try to express the differences between the several interfaces on a complexity scale. However, there is no good measure for complexity. The only quantative measure that comes close is the Ngo-like measure for complexity in appendix C.
Several options were tried. The position of the mouse cursor is logged, the clicks on the control elements are logged and the input values of the system are logged.
Someone suggested to try several, very different interfaces; (DR. C.J. OVERBEEKE, DELFT UNIVERSITY OF TECHNOLOGY, INDUSTRIAL DESIGN ENGINEERING, ASSOCIATE PROFESSOR FORM THEORY). This could lead to very large differences in which it is hard to find a pattern or relation, so I decided to try to figure out the effect of trend information or a short-time prediction. (A short time prediction is calculated by and presented on the interface.) I will investigate of this makes the display less complex (easier to control the process). I expect this increases the score or system performance, decreases the mean error and decreases the mental load.

5.1.3 Mental load

The mental load will me measured using a RSME (Rating Scale for Mental Effort) question form. This form is printed in Appendix G. At the MMS group there have been good results using the RSME form, and as it is used in similar experiments, so it will be possible to compare data.

5.1.4 Performace

There are several measures for performance. The mean error for each cell in the twelve-cell experiment experimental system can be calculated. These values can be used to calculate an overall mean system error, which is a measure for performance. A is a performance factor of the system is defined as well. These are calculated as shown below [REF 16 PAGE 63,64]:

$$ME_i = \frac{\sum_{k=0}^{M} |x_i[k] - SP_i[k]|}{M+1}$$

$$SE = \frac{\sum_{i=1}^{N} ME_i}{N}$$

$$SPF = 1 - SE$$

Where ME_i is the cell mean error of cell i, $x_i[k]$ the output of cel i for the time-period k, M is the total of sample periods, i is the cell number, SE is the system error, N is the total number of cells and SPF is the system performance factor.

5.2 System

For this experiment, I use an existing experimental system. It is written in MatLab/Simulink 6.1 and uses its real-time toolbox. With several adaptions, this system can be used in this experiment.

The adaptations are:

1. create another interface with trendinformation
2. logging the mouse data
3. data evaluation scripts

The experimental system that the test subjects are requested to regulate will be a system of twelve cells containing a first order process that is described by the following equations:

$$Y(n) = Cx(n) + Du(n)$$
$$x(n+1) = Ax(n) + Bu(n)$$

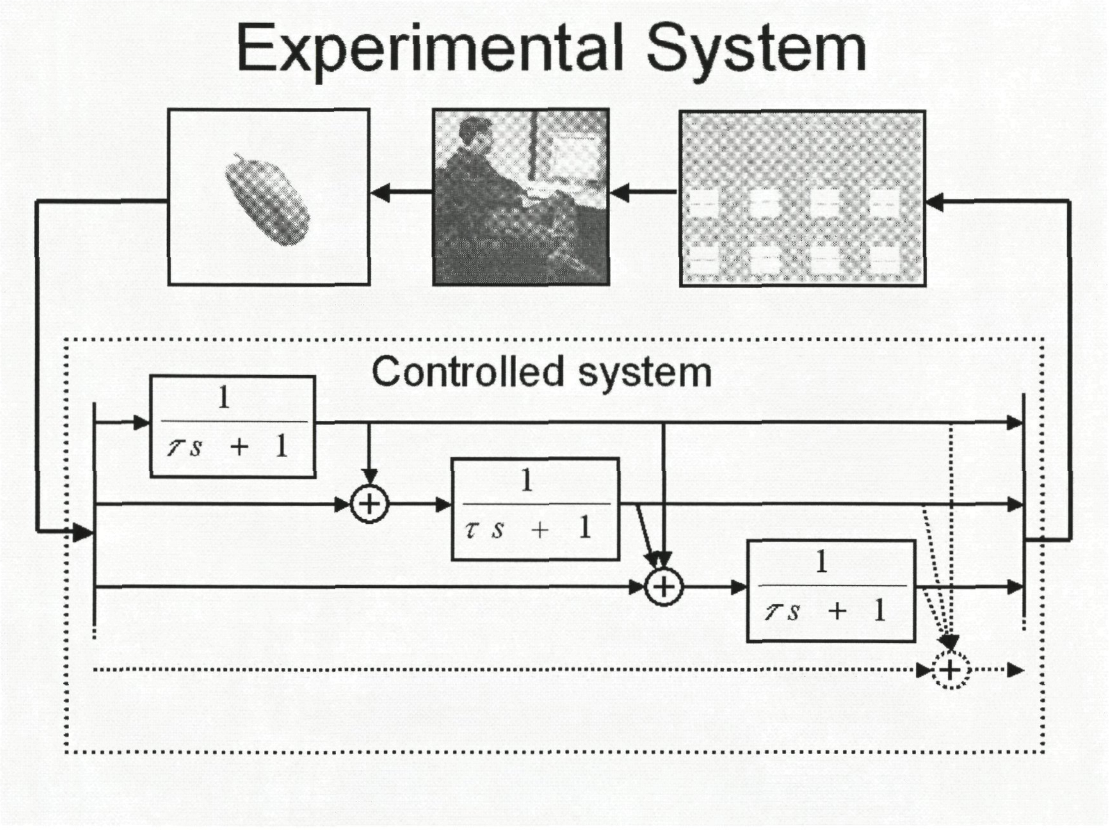


Figure 9: The experimental system **(figure by ARJEN VAN DER HORST, photograph by MARTIJN LANGELAAR)**

(τ=10 sec)

The cells and the input values can be set to be dependent of each other. Feedback and feed forward values can be altered, coulomb frictions a time delays can be added.

For this experiment, all the feed forward factors are set to 0,5 and all other options aren't used. So the output of a cell will be multiplied by 0,5 and added to the input of all the next cells. They don't react instantly because of the sample time and system dynamics. The sample time causes a little delay. For example, if you set the input of cell 1 to 1, it takes some time untill you see cell number 12 increase. This effect is due to the dynamics of the system, and definitely no time delay.

The system has the possibility to control cells automatically, to reduce the workload of the operator. In this experiment set-up, cells 5 to 9 are automatically controlled.

The responses of the experimental system are calculated in the simulink-part of the software. An overview of the structure of the simulink system is in Appendix J.
The simulation runs with a sample time of 0,1 second. In the beginning of each second, 10 iterations are done, and if there is some time left, the system waits. If the system can't do 10 iterations in one second, then there is a problem. This problem is logged, but it doesn't occur very often and doesn't seem to influence the test results.

The rest of the test software is written in the MatLab script language. Several functions are carried out:

1. changing system variables to define system behavior
2. creating (semi) random setpoint requests
3. determining the experiment-sequence (latin-square)
4. displaying the user-interface
5. logging data
6. examining data

In Appedix H the testers manual is printed. Several MatLab script files that were customized for this experiment are printed in Appendix I.

5.2.1 Semi-random setpoint requests

There is a setpoint request with a value between 0 and 2 every 30 to 40 seconds. These requests are generated semi-randomly. This is done because totally randomly chosen setpoint-requests can result in a very easy or difficult experiment.
To avoid this, the list of setpoints is tested for several requirements:

1. the maximum difference in number of setpoints between the rows is 1
2. the maximum difference in number of setpoints between the columns is 3
3. the maximum difference in number of setpoints between the left half and the right half of the screen is 3
4. the maximum difference in number of setpoints between each cell is 3

The setpoint requests are not predetermined for each experiment because in that case there would be a risk that not the properties of the system and the interface are measured, but rather the properties of the chosen set of setpoint reqests.

5.3 Interface

The interface is the only element that the test-subjects see of the system. They are requested to regulate the system using one of the interfaces below. The interfaces are similar, but not quite equal.

Each cell is regulated by a box. The box contains a graph presenting historical data. The system data is a blue line, and the setpoints are presented as a red dotted line. The width of the graph is a time period of 9 seconds. The display refresh rate is once per second.
Right to the graph are a slider bar that changes the input value and a reset button to set the input value back to zero. For each cell there are several alerts. When there is a setpoint request, the box is accentuated by a blue border and the background of the setpoint indicator turns to yellow. Each cell has an alarm : when the difference between the actual and requested value is greater than 0,3 a yellow alarm-box appears at the top of the cell. If the difference is greater than 0,4 the box turns red. The text in the alarm box indicates if the actual output of the cell is too high or too low.

The experimental system came with a standard interface shown below. Directly above the graph there are boxes with the set point-request, input value and output value.On the top of each square there is the cell number and an alarm. The slider steps are 0.5.

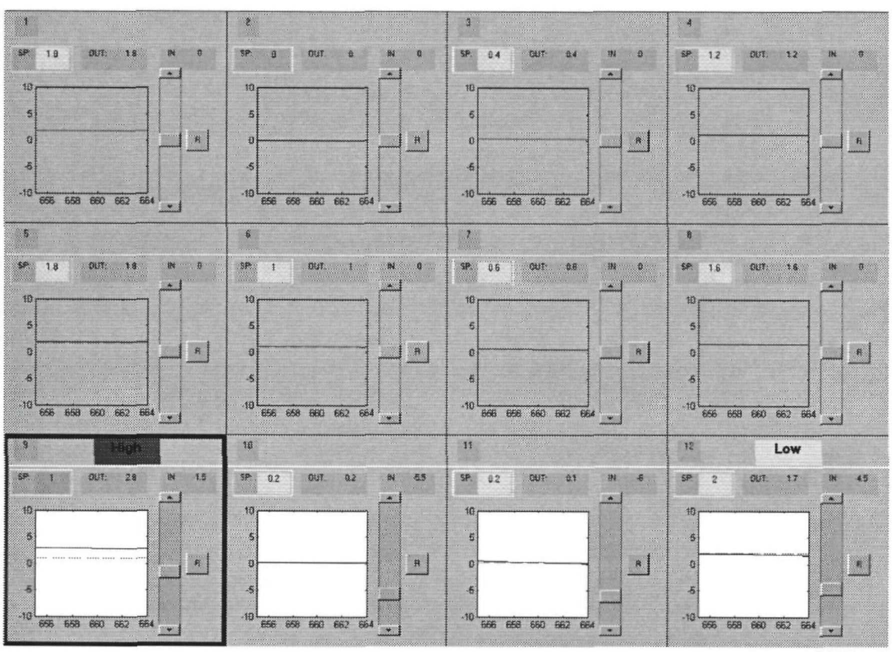


Figure 10: Standard interface

For this experiment, some modifications were introduced in the interface. The positions of the input value, output value, setpoint value and the alarm were changed. Also the text values have been downsized to create more room for the graph.

Their order is now more logical, with the input at the left of the output. The setpoint value above the output is also a little more logical.

The input value is no longer displayed directly above the slider that changes it. The slider step is now set to 0,1.

For the versions with trend information (future prediction), a vertical line in the graph represents the actual value. The information left of the vertical line is the history and the information right of the line is a prediction. The prediction calculates the slope of the last two (1s) steps. The weighted average of these slopes, the most recent slope having twice as much weight, is used to calculate the prediction. The prediction time in the new interface will be 0, 2 or four seconds.

$$slope = \frac{(y_{t-1} - y_{t-2}) + 2*(y_t - y_{t-1})}{3}$$

$$prediction = slope * predictiontime$$

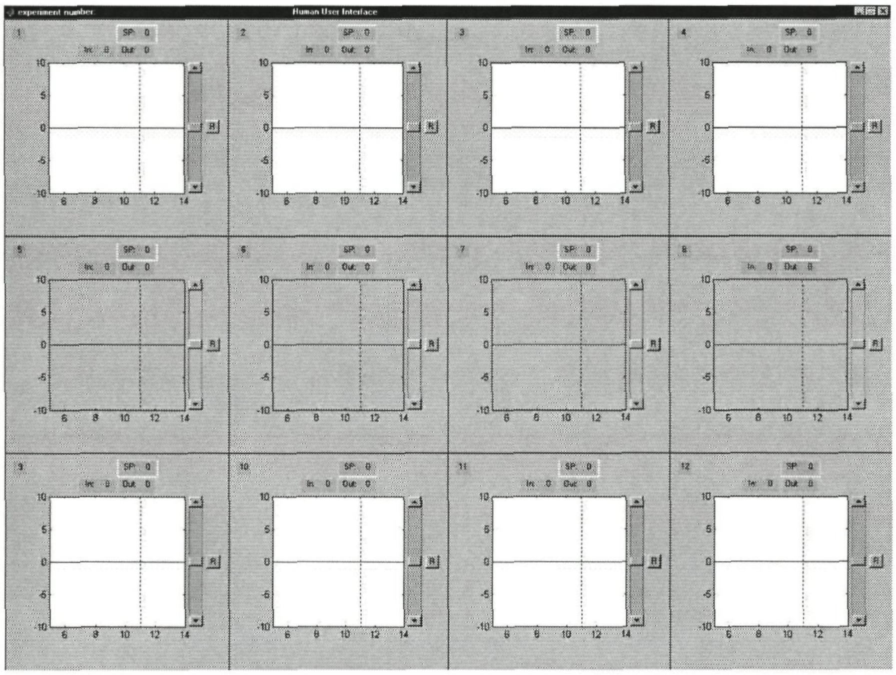


Figure 11: New interface

The score for Ngo-like interface complexity of both interfaces is calculated in Appendix D.

The scores of the standard interface are:

Balance	Equilibrium	Symmetry	Sequence	Simplicity	Density	Regularity	Economy
47	0	46	0	84	9	45	0
Homogenity	Unity	Cohesion	Proportion	Rhytm			complexity
33	50	20	11	47			33

Table 2: Ngo-like complexity for standard interface

The scores of the new interface are:

Balance	Equilibrium	Symmetry	Sequence	Simplicity	Density	Regularity	Economy
86	1	55	0	82	12	28	0
Homogenity	Unity	Cohesion	Proportion	Rhytm			complexity
33	50	22	14	56			37

Table 3: Ngo -like complexity for new interface

5.4 Tests

5.4.1 Sequence (latin square)

Each test-subject has to do several tests. The adjusted interface is configured with trend information of 0 sec, 2 sec and 4 sec ahead. These interfaces all are used twice. To be able to compare the results with other experiments, the old interface is also used, but only once.

In the beginning of a test session of 7 tests, a subject learns how to operate the system, and at the end he might lose concentration or he might be tired. So the performance of the subjects also depends on the timing. The first and/or last few tests might show a worse performance. To eliminate this effect, the **latin square** methodology is used. This means the test sequence is kept the same, but each subject starts in a different place in the sequence. When the mean value of the performance and workload is taken, the learning effect is compensated. The table shows the configurations of the tests.

Experiment number	Interface
E1	new, 0sec trend
E2	new, 2sec trend
E3	new, 4 sec trend
E4	old (no trend)
E5	new, 0sec trend
E6	new, 2sec trend
E7	new, 4 sec trend

Table 4: Experiments

5.4.2 Test run

Appendix E shows used to describe one test-run. This form was printed for each subject, and was also used to write down any comment that the subjects made during the test sessions. After welcoming a test subject, they received the subject instructions for reading. These instructions can be found in Appendix F . Thereafter, the task was explained, and they could ask questions. The subject is also told that he/she can determine himself when he wants break. Then the subject can do a little training/experimenting with the system, starting with the old/original interface. After about 200 seconds, this is stopped. Then, the subjects are asked to do a complete 15 minute exercise run, using the new interface with 2sec of trend information. Now the subject is assumed to be trained. At the end of each session, the computer displays a graph that presents the errors in the separate cells, and displays a score that is similar to the system performance. When the exercise/training, is finished, they are requested to fill in an RSME-form. RSME, the Rating Scale for Mental Effort is a way to measure the operator's workload through a subjective analysis. The subject has to categorize the perceived workload on a predefined scale. The form used is included in Appendix G. When this is done, the real test sequence begins. The computer determines the sequence of the experiments by a latinsquare routine. The subject has to do 7 experiments, each taking 15 minutes (900 seconds).
When all the experiments are done, there is a small interview of the subject to ask if he or she noticed the several differences between the interfaces, and what he liked or disliked, or that he meant is easy or difficult.
Finally the subject is paid (€5 an hour) and thanked for his or her time and effort.

5.4.3 Choice of test subjects

It is desired that test subjects have some knowledge of (and experience in) the field of control systems engineering. This knowledge is preferably practical rather than theoretical. To get a reasonably homogenous group of subjects, only students of the Delft University of Technology are asked. People of the following (sub)faculties make good test persons: Aerospace Engineering, Chemical Technology (ChemTech), Applied Physics, Mechanical Engineering, Marine Technology, Technical Informatics.

6 Results of experiment

In this chapter, the results of the experiments are discussed. First some remarks of the subjects are stated, in general, and specially the remarks about the interface. After that, the data processing is described. The calculated measures are defined and explained. With the results of the calculations, some specific plots were be made and printed in Appendices L, O and P.
Finally, the data in the plots can be analyzed to see if there are some relationships between the measured quantities, especially between the quantities that are in the objectives of this research.

6.1 General remarks of test subjects

Some subjects talked a lot; to themselves, to the computer and to the tester. Most of the remarks were written down. A lot of people commented on their score and performance. They told at the end of the test they were getting tired; sometimes they were demotivated by a bad score in the previous test, or a bad start of a test session. Some of them thought the 15 minutes were too long for a continuous test session, and some commented that 7 tests were a lot, and they didn't like it anymore.

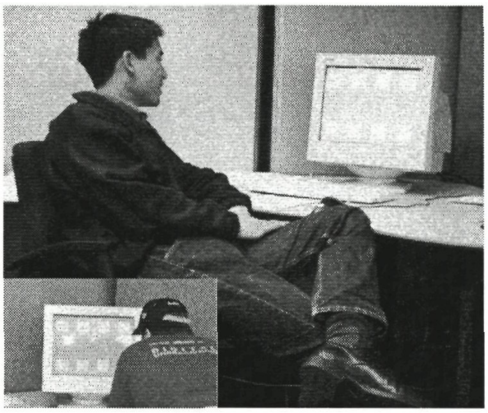


Figure 12: Test in progress

6.1.1 Remarks about the interface

The subjects were directly asked about the complexity of the interface. They couldn't answer that question. They didn't seem to see a difference, or couldn't handle the construct of complexity.
Most people told the tester that the new interface was better. However, not that they worked with the new interface a total of 7 times (6 test sessions and once during training) and with the old one only twice (once during a session, and about 200 secs during training).
The only remark they all made was that they greatly preferred the slider change step of 0.1 in the new interface. Other comments were rather personal. Some subjects liked the big font and big alarms of the old interface, other people mentioned the big alarms made them nervous. The locations of the in-, out-, and setpoint values were experienced as more logical, however, one of the subjects noted that she liked that the input value of the cell was above the slider that changed it. All in all, the personal preferences varied greatly between the test subjects.

6.2 Data processing

6.2.1 Relations

This objective of this research project is to investigate a relations between interface-complexity and workload, and between interface-complexity and performance. As I haven't been able to measure interface-complexity, some human behavior properties are logged in order to see if some connection can be revealed. Figure 13 shows that I hope to be able to conclude something about complexity by analyzing some other quantities.

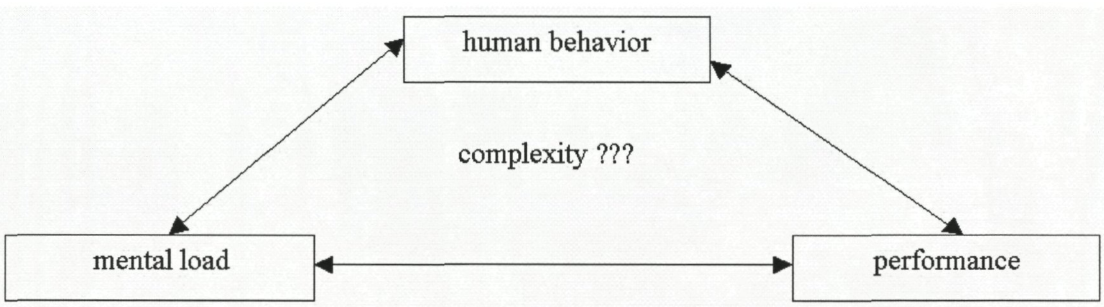


Figure 13: The measured and investigated quantities

The measures in the boxes are examined separately and the relations indicated by the arrows are eamined as well.

6.2.2 Data sheet

First, for each subject a small data sheet was generated, with some basic results of the tests and some basic remarks. These data sheets are printed in Appendix K. The best session for each experiment is printed bold.

6.2.3 General overview plot

After that, a general overview plot was made for each session. An example is in figure 14, the rest of the plots are printed in Appendix M. This gives a general idea of the behavior. The total mean error, the mean error of the upper four cells, and the lower four cells was plotted in time. When there was a setpoint request was also plotted. Some subjects show a total different general image. Some su bjects seem to be in control of the process, and others responded more chaotically. The plots are printed in Appendix L and the MatLab script used to produce them in Appendix M.

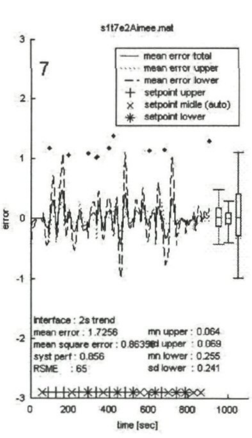


Figure 14: Example of general overview plot

By examining these plots, it might be necessary to reject some of the test subjects that responded rather chaotically. This has to be statistically tested.

6.2.4 Human behavior measures

The research is about the relation between complexity and performance and workload. As measure for performance, the system's mean error of all cells is used. For the measure of workload the overall RSME value is used. I try to discover if human behavior is related to human performance, workload or maybe even to complexity. Therefore, several human behavior measures are calculated.
The first property is the number of **control actions**. The input value of each cell was logged each second. The changes of these input values were counted to find the number of control actions carried out by the subject.
Secondly, the number of **mouse clicks** on control elements was logged.
Third, the **mouse distance** was calculated. The mouse position was logged and from these coordinates, the total distance covered by the mouse, in screen-coordinates (1 is from the left to the right or from top to bottom of the screen), was calculated.
Finally, the number of **mouse direction changes** was studied. The datalog of the mouse coordinates was used for this. Three points on the mouse path define two vectors. The second vector, between point 2 and 3 is also plotted pointing from the same origin as the first vector. The angles α_1 and α_2 between both vectors and the horizontal line are calculated. The difference between these angles is β, the angle over which the mouse direction changed.

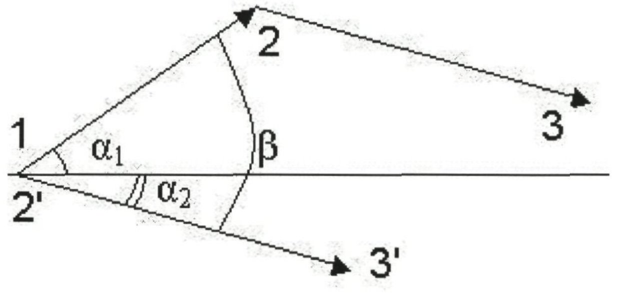


Figure 15: calculation of mouse direction changes

The total change of angle is calculated, and also the number of angle changes in several categories (0-45 deg, 45-90 deg, 90-135 deg, 135-180 deg, 0-90 deg and 90-180 deg) are counted. The groups used in the plots are 0-90 degrees and 90-180 degrees.

6.2.5 Efficiency of human behavior

From the variables described above, we derive several measures of efficiency.
The Efficiency can be examined by examining the mouse distance in relation to the number of the control actions. The expectation is that a chaotic, nervous subject covers a lot of distance and makes a lot of control actions.
Something similar is done with mouse distance in relation to mouse clicks. Again the expectation is they both have large or small numbers.
The last combination, the relation between control actions and mouse clicks is plotted as well.
There might be a relation between the efficiency and the workload or between the efficiency and the performance. Therefore, three measures of efficiency are defined. First the quotient of the number of control actions and the number of mouse clicks, and secondly two different quotients of the mouse distance and the number of direction changes. The 0-90 and the 90-180 degree groups are examined. Finally the procduct of the mouse distance and the number of direction changes is examined, again for 0-90 and 90-180 degrees groups.

6.2.6 Performance

For performance, the generally used measure is the **mean error**, as defined in section 5.1.4. But also the **system performance factor** and the **mean square error** are calculated.

6.2.7 Workload

For workload, there only is one measure, the **RSME** value.

6.3 Data plots

First, the general behavior plots were mentioned in section 6.2.3 were made.
After that, for all the mentioned measures for performance, human behavior and workload, several figures were created with the same y-axes (e.g. mean error, RSME, mouse distance). The experiments were grouped and sorted in several ways. These figures are in Appendix O.

1. The results first sorted by experiment and plotted in four separate subplots. On the x-axes of the subplots the subject numbers are located. At the right side of each square is a figure that indicates the mean value, standard deviations for points above and below the mean value, and the minimum and maximum value.
   This figure shows if the several subjects behaved similar in each experiment.
2. Experiments sorted first on subject number, and secondly on experiment number. For each subject the figure showing the mean values and standard deviation is shown.
   This figure shows the behavior of the subjects during the different experiments.
3. Results first sorted by experiment, secondly by subject number. This plot shows the same as the first, only in this figure the experiments can be compared too.
4. The last figure shows seven subplots, one for each subject. The experiments are sorted by test number.
   This figure might give some information on the learning effect.

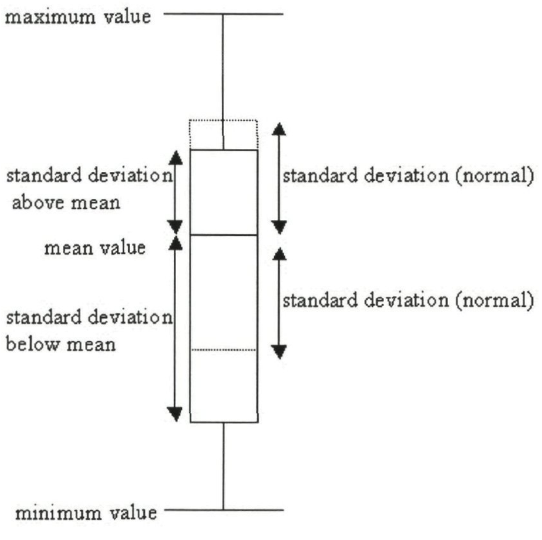


Figure 16: presentation of mean value and standard deviation

These figures for mean error, square error and RSME are printed in Appendix O.

The mouse behavior was studied separately. Several of the mouse behavior factors and efficiency factors were plotted to see if there was a connection between the mutual mouse behavior factors. The only connection found is the linear relation between number of mouse clicks and number of control actions.

Finally, some figures were created to look for relations between several measured quantities. These figures are in appendix P.

A. Performance and workload

A graph was plotted to look for a connection between performance (mean error and workload (RSME)
These figure is printed in Appendix P.

B. Performance and human behavior

Several plots were made, sum mean error on the x-axes was plotted with on the y-axes :

- control actions
- mouse clicks
- mouse distance
- mouse direction changes (0-90deg and 90-180 deg)

These figures are printed in Appendix P.

C. Workload and human behavior

To investigate this connection, I plotted the RSME value on the x-axes against several measures on the y-axes:

- control actions
- mouse clicks
- mouse distance
- mouse direction changes (0-90deg and 90-180 deg)

These figures are printed in Appendix P.

6.4 Data analysis

Most of the plots show no relations at all.
The plot where the mean absolute error of all the experiments was plotted, created the suspicion that some points belonged to a different group. There might be a group of subjects that are in control of the system, and a group of subjects that is not in control of the system. This was analyzed with a student-t test for a random spot check for the mean value of two groups with unknown, but assumed equal standard deviation, with an unreliability α of 5% [LIT 17, PAGE 34 TEST 2.10]. A copy of this test is in appendix Q.
The hypothesis tested was that the selected points formed a group with the same mean value as the rest of the points. If this hypothesis was rejected, the points could be rejected, otherwise the points have to be accepted as part of the main dataset.
The points that are marked by a circle were rejected, they indeed belong to a different group. The points marked by a box weren't rejected, they do belong to the main dataset.

experiment	v	t	t_{kr}	conclusion
0 sec	5	-4.095	2.57	points rejected
2 sec	1	-2.664	12.7	points accepted
4 sec	14	-14.258	2.15	points rejected
old	2	-1.972	4.30	points accepted

Table 5: Student t test

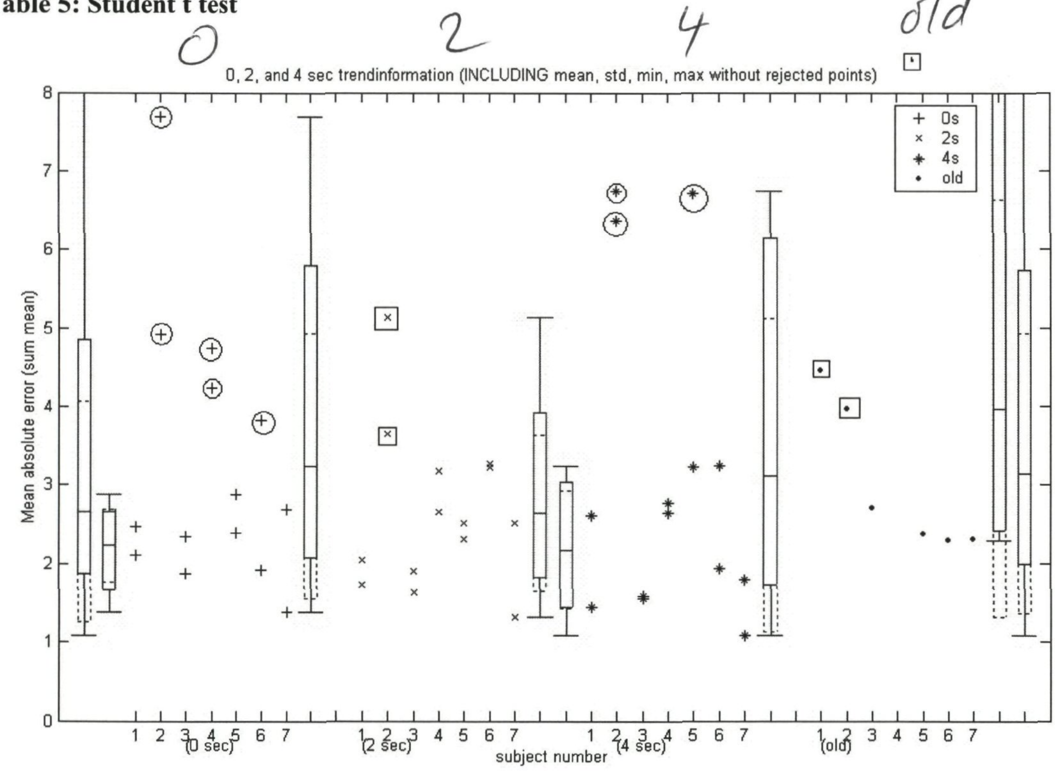


Figure 17: mean error plot used for the student-t test

The figure shows the plotted mean error values for the several experiments and subjects. Directly after each experiment (after the 14 plotted points) the figure indicating the mean, standard deviations, minimal and maximal values is plotted for those 14 points. For the experiments with rejected (circled) points, a similar figure is printed for the values without the rejected points, this figure is before the 14 points.

The figure totally right is the figure for all plotted points, and the figure totally left is a similar figure without the rejected points.

The picture clearly shows that the mean values fluctuate a little, but always within the bandwidth of the standard deviation, so the fluctuation isn't a significant difference between the several experiments. The standard deviations themselves fluctuate too.

This corresponds with the remarks of the subjects, who generally told me they didn't see very much difference between the interfaces.

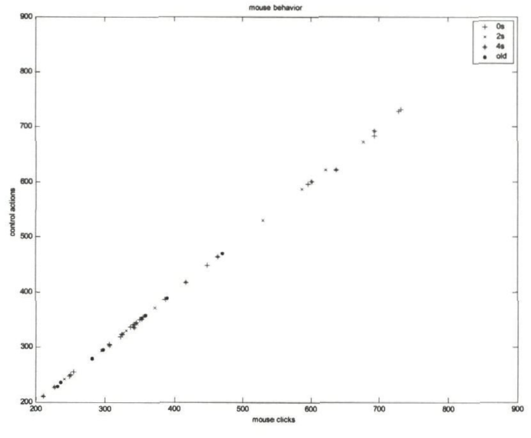
The data can be compared to Wei's data [LIT 16, PAGE 71]

	ex rejected points	Wei	incl rejected points
performance factor	0.715	0.724	0.738
RSME2	45.95	47.00	49.69
RSME standard deviation	21.21	19.66	22.97

Table 6: Comparision with Wei's data

This shows that the subjects performed similarly to the subjects of Wei.

The only plot that showed a very clear coherence was the plot of control actions versus mouse-clicks. There is a very clear linear relation between these two.



7 Conclusions

The subjects responded similar to Wei's subjects [REF 16], so their performance and motivation was not insufficient.

The only relation found was between control actions and mouse clicks, this is a very trivial relation.

The fact that the subjects didn't see a lot of difference between the interfaces with the time prediction or trend information and the fact that their performance, workload or behavior didn't change significantly, leads to the conclusion that the addition of this extra information didn't help. Probably the system reacted in a reasonably predictable way, the subjects could predict the system's reactions very well. There was no need for the interface or computer to help them with the prediction, or the prediction isn't necessary to be able to control the system. People can do more than we think they can.

Not all the subjects liked the same interface, and they had several reasons. Apparently some issues about interface design are rather personal. It isn't necessarily true that interfaces that are experienced better, also show better performance.

Because no relation was found, unfortunately, no conclusions can be drawn about complexity. Complexity is hard to measure.

8 Recommendations

Although no interesting relation was found, some possible improvements for further experimental research are suggested.

If you are varying the interface by displaying trend information, there are three main recommendations:

1. use a less predictable system, so the subjects can't do the prediction themselves
2. use a system that is more difficult to control, so the prediction will be necessary
3. display the trend information clearly separately from the actual value and historical information

The experimental system software can be improved, maybe by using Pascal or C and not MatLab/Simulink, because MatLab isn't very suitable for extended interface design research. It might be possible to consider LabView.

With the set of data, maybe some extended analyses can be performed, I don't expect any information on the relations I was looking for, but maybe there is some information in the RSME data. There might be a connection between the separate cell RSME values and the overall value that can be discovered.

Further research can be done to see if there is difference between the reactions of a test person to a real system and a fictive system. And it is interesting to investigate if the interfaces that lead to low workload and high performance are the ones the subjects experience as the better interfaces.

References

Author; title and type of document; source; location; year

1. Guimaraes, Tor; Martensson, Nils; Stahre, Johan; Igbaria, Magid; *Empirically testing the impact of manufacturing system complexity on performance*; INTERNATIONAL-JOURNAL-OF-OPERATINONS-AND-PRODUCTION-MANAGEMENT. 1999; 19 (12) pp 1254-1269; 1999
2. Ham, Donh-Ham; Yoon, Wan Chul; *The effects of presenting functionally abstracted information in fault diagnostics tasks*; RELIABILITY ENGINEERING AND SYSTEM SAFETY 73 (2001) pp 103-119; 2001
3. Keyes, Elizabeth and Krull, Robert; *User Information Processing Strategies and Online Visual Structure*; SIGDOC '92 The 10th Annual Internationtal Conference pp 121-8; New York, USA; 1992
4. Ngo, David Chek Ling; Samsudin, Azman and Abdullah, Rosni; *Aesthetic measures for Assessing Graphic Screens*; JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 16, pp 97-116; 2000
5. Ngo, D.C.L., Teo, L.S., Byrne, J.G.; *Formalizing Guidelines for the design of screen layouts*; DISPLAYS 21, Issue 1, March, Elsevier Science Journals; 1999
6. Ngo, David Chek Ling; Byrne, John G; *Aesthetic Measures for Screen Design*; : Proceedings 1998 Australasian Computer Human Interaction Conference; Los Alamitos, CA, USA; 1998
7. Ngo, David Check Ling; Ch'ng, David; *Screen Design: compsing with dynamic symmertry*; DISPLAYS 22 (2001) pp 115-124; Elsevier Science Journals; 2001
8. Ngo, David Chek Ling and Byrne, John G.; *Another Look at a model for evaluating interface aesthetics*; INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS AND COMPUTER SCIENCE, vol 11, no2, 2001, pp 515-35; 2001
9. Nørretranders, T; *Het bewustzijn als bedrieger ; een mythe ontrafeld* (chapter 4: De diepte van complexiteit); translation from Danish to Dutch by Kor de Vries; Arbeiderspers, Amsterdam, 2000
10. Parush, Avraham; Nadir, Ronen and Shtub, Avraham; *Evaluating the Layout of Graphical User Interface Screens: Validation of a Nummerical Computerized Model*; INTERNATIONAL JOURNAL OF HUMAN-COMPUTER INTERACTION, 10(4) pp 343-360; 1998
11. Tractinsky, N; Katz, A.S. and Ikar, D; *What is beautiful is usable*; INTERACTING WITH COMPUTERS 10 (2000) pp 127-145; 2000
12. Tullis, Thomas S.; *Screen Design*, Chapter 18 of HANDBOOK OF HUMAN-COMPUTER INTERACTION, Elsevier Science Publishers; 1988

13. Vanderdonckt, Jeanl; *Visual techniques for Traditional and Multimedia Layouts*; PROCEEDINGS OF THE WORKSHOP ON ADVANCED VISUAL INTERFACES; 1994
14. Vicente, K.J. and Rasmussen, J; *The Ecology of Human-Machine Systems II: Mediating "Direct Perception" in Complex Work Domains;* ECOLOGICAL PSYCHOLOGY 2(3), 207-249; Laence Erlbaum Associates; 1990
15. Vicente, K.J. and Rasmussen, J; *Ecological Interface Design: Theoretical Foundations*; IEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS, Vol22 no.4, 1992
16. Wei, Zhi-Gang; *Mental Load and Performance at Different Automation Levels*, Thesis for Delft University of Technology; Delft; 1997
17. Zijp, drs W.L.; *handleiding voor statistische toetsen*; h.d.tjeenk willink; Groningen; 1974

Appendix A: Complete text of Assignment; student goals and poster for poster session (all in duch)

TUDelft
Delft University of Technology

FACULTY OF DESIGN & ENGINEERING
MECHANICAL ENGINEERING AND MARINE TECHNOLOGY
Man Machine System

TUDelft - OCP – MMS, Mekelweg 2, 2628 CD Delft, The Netherlands

Tel.: +31-15 278 6400 FAX: +31-15 278 4717
mms@ocp.tudelft.nl
http://www.ocp.tudelft.nl/mms

Afstudeeropdracht: Martijn Langelaar

t.b.v. Hogeschool Haarlem

17 januari, 2002

Binnen de sectie Mens-Machine Systemen van de opleiding Werktuigbouwkunde van de TUDelft wordt o.a. gekeken naar de rol van de mens bij het superviseren van complexe systemen. Bij complexe systemen is er een onderscheid te maken tussen de Systeem-complexiteit, de Taak-complexiteit, en de Interface-complexiteit. Voor de eerste twee zijn in het laboratorium proeven gedaan waarbij operator prestatie maten en werkbelasting werden gemeten afhankelijk van de complexiteit (waarvoor een maat is ontwikkeld). Voor de Interface-complexiteit zijn nog geen experimenten gedaan.

Van u wordt gevraagd

1) een korte 'state of the art' te geven van de wetenschappelijke studies op dit onderwerp voorzover gepubliceerd,
2) een maat te ontwikkelen voor de interface complexiteit voor de simulatie van het artificiele systeem dat in ons laboratorium beschikbaar is,
3) een experiment te ontwerpen en uit te voeren om de invloed van de Interface-complexiteit op de prestatie en de werkbelasting te bepalen.

Bij de uitvoering van uw studie kunt u gebruik maken van het laboratorium en van de ondersteuning door de heer A. van der Horst.

Wij gaan er van uit dat u tijdens uw studie deel uitmaakt van de onderzoeksgroep en dus ook deelneemt aan de werkbesprekingen. Wij zullen bij gebleke geschiktheid u stimuleren de resultaten van uw onderzoek te publiceren.

Ik wens u veel succes bij de uitvoering van uw studie.

Met vriendelijke groet,

Prof.dr.ir. P.A. Wieringa

Eindtermen

Door de HTS geformuleerde kern-eindtermen:

a) De student kan zelfstandig en interdisciplinair werken.
b) De student kan relevante informatie in het kader van de opdracht opsporen, selecteren en verwerken.
c) De student kan voor technische problemen, die samenhangen met de afstudeeropdracht, oplossingsalternatieven genereren. Door systematisch denken analyseert hij het probleem, bedenk daarvoor oplossingen en maakt een beredeneerde keuze voor de best passende oplossing.
d) De student beschikt over goede communicatieve vaardigheden in woord en geschrift.

Door de student geformuleerde aanvullende-eindtermen:

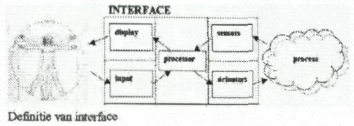
e) De student is in staat een wetenschappelijk experiment te ontwerpen, uit te voeren, de data te analyseren en interpreteren. En vervolgens is de student in staat om conclusies te trekken.
f) De student krijgt een beeld van de ontwikkelingen op het gebied van de Man Machine Systems.
g) De student krijgt inzicht in het gedrag van een menselijke bestuurder (in het engels: Human operator) bij verschillende interfaces.

Appendix A: Complete text of Assignment; student goals and poster for poster session (all in duch)

Interface-Complexity

And its influence on performance and workload

Bedrijf, afdeling: TU Delft; Mens Machine Systemen
Naam student: Martijn Langelaar (studentnummer: 00.2.00504)
Afstudeerperiode: Januari 2002 – Juni 2002
Begeleider HTS: Drs.Ir C. van der Vlugt
Begeleiders TU: Prof. Dr. ir. P. A. Wieringa; Arjen van der Horst

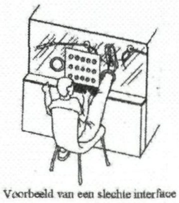


Definitie van interface

Overal waar sprake is van Mens-Machine interactie is er een interface die deze interactie mogelijk maakt. De interface is de brug tussen mens en techniek.

De kwaliteit van de interface bepaalt in belangrijke mate of en hoe de techniek gebruikt wordt.

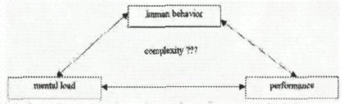
De opdracht is het bestuderen van de complexiteit van de interface.


Voorbeeld van een slechte interface
[Rasmussen & Vicente (1990)]

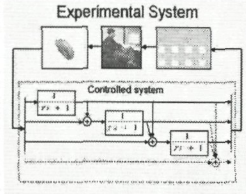
Helaas is complexiteit niet goed te definiëren en ook niet te meten.

Daarom wordt naar andere grootheden gekeken om te zien of daarmee iets gezegd kan worden over complexiteit.

Er is gekeken naar prestatie, werkbelasting en gedrag van de operator, in de hoop uit deze grootheden iets af te kunnen leiden over de complexiteit.



Onderzochte grootheden in het onderzoek naar complexiteit

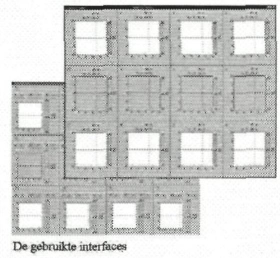


Opzet van het experimentele systeem (τ =10s)
[Foto: M.Langelaar, plaatje: A. van der Horst]

Met het op de vakgroep beschikbare experimentele systeem (van 12 gekoppelde eerste orde processen) is een experiment uitgevoerd om te zien hoe mensen omgaan met verschillende interfaces. (Waarvan wordt aangenomen dat de complexiteit ook verschilt.)

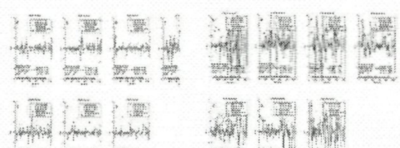
Proefpersonen hebben het systeem geregeld met verschillende interfaces. De layout van de interfaces verschilt, en sommige interfaces hebben een trendindicatie of voorspelling.

Hierbij is de performance gemeten, zijn de muisbewegingen gelogd en is gevraagd naar de mentale belasting met behulp van een RSME (Rating Scale for Mental Effort) formulier.

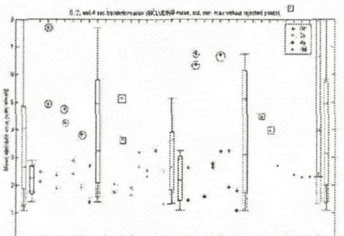

De gebruikte interfaces

Resultaten:

- De prestaties van de gebruikte proefpersonen zijn vergelijkbaar met de prestaties van proefpersonen in vergelijkbare onderzoeken.
- Uit de analyse van de testdata blijkt dat er geen significant verschil is tussen de verschillende interfaces. De proefpersonen zeggen ook dat ze in het algemeen erg weinig verschil hebben gemerkt.
- Waarschijnlijk zijn mensen in staat het systeem te regelen zonder informatie over wat er in de toekomst gaat gebeuren, of het kost erg weinig moeite deze voorspelling zelf te doen.Deze voorspelling heeft kennelijk nauwelijks meerwaarde. Mensen kunnen meer dan we denken.
- Er kan geen conclusie getrokken worden over de complexiteit, wel kunnen er aanbevelingen gedaan worden voor verder onderzoek.


Algemeen beeld van het gedrag van een operator


Een proefpersoon aan het werk


De gemiddelde absolute fout van de verschillende proefpersonen

Author(s) and year	Title and Summary	
Ngo, David Check Ling; Byrne, John G.; 2001 [8]	Another look at a model for evaluating interface aesthetics Design guidelines for HUI's are generally considered useful, but there are significant problems in their application. The design rules are often oversimplified. The aesthetic measures are the same as in 'Formalising guidelines for the design of screen layouts'. The range of the measures is between 0 (worst) end 1 (best) Future work includes investigation of two assumptions: 1) interaction between characteristics is linear 2) all characteristics are equally important	HUI: Human User Interface
Ham, Dong-Han; Yoon, Wan Chul; 2001 [2]	The effects of functionally abstracted information in fault diagnosis Inadequate display design would be the cause of severe accidents. The key questions the designers should solve are what information to provide through interface and how to visualize the information. A user-centered approach is necessary. Ecological approach emphasizes a proper interaction between the human operator and the work, rather than the machine. The taskload of the operator changes with the abstraction level of the display. Best performance is achieved when information on Functional Purpose, Physical function and Generalized function is displayed.	EID: Ecological Interface Design VID: Visual Interface Design EOD: Ecological Interface Design GF: Generalized Function AH: Abstraction Hierarchy AF: Abstract Function FP: functional Purpose AF: Abstract Function PF: Physical Function
Ngo, David Chek Ling; Ch'ng, Eugene; 2001 [7]	Screen Design: composing with dynamic symmetry Very high correlations have been found between aesthetics of a screen and perceived use of a screen. A poorly designed screen can hinder communication. Aesthetically pleasing layouts motivate students to learn. Symmetry and certain rectangles (1:1; 1:1.414; 1:1.618; 1:732; 1:2) are considered aesthetic. Static vs dynamic symmetry. Hambidge's technique for realising dynamic symmetry.	

Ngo, David Chek Ling; Sansudin, Azman; Abdullah Rosni; 2000 [4]	Aesthetic Measures for Assessing Graphic Screens One way to enhance the usability of a computer system is to improve appearance of the user interface. Improved interface aesthetics makes computers easier to use because the visual communication is better. Important factors are: amount of information on the screen, screen organization, language, distinctiveness of components and feedback. One way to make products easier to use, and users more productive, is to improve the visual communication that takes place in all the elements of the UI: metaphors, mental models, navigation scheme, look and feel. Aesthetics may not be the only solution. Certainly, it is related to our appreciation of computer systems. Layouts have effect on a student motivation to learn. In particular: A=attention, R=Relevance, C=Confidence, S=Satisfaction. Measure of balance, equilibrium, symmetry and sequence is described. Also a formula to calculate measure of order and complexity is given.	UI: User Interface. A=Attention, R=Relevance, C=Confidence, S=Satisfaction
Tractinsky, N; Katz, A.S.l Ikar, D.; 2000 [11]	What is beautiful is Usable The tension between form and function has long been at the crossroad of artifact design. Whereas emphasis on function stresses the importance of the artifact's usability and usefulness, accentuating the artifact's form serves more the aesthetic, and perhaps social, needs of designers and customers. Perhaps in a backslash to recent tendencies by the computer industry to oversell glitz and fashion in these products or because of its origins in disciplines that emphasize efficiency, the field of Human-computer interaction appears to stress the prominence of usability over aesthetics. In a sense the concepts of aesthetics and usability represent two orthogonal dimensions of HCI. Whereas aesthetics usually refers often to non-quantifiable, subjective and affect-based experience of computer use, usability is measured by relatively objective means and sets efficiency as its foremost criterion. The near neglect of the aesthetic aspect is for several reasons; first there is a gap between the practice of industry and the research. Second it ignores important needs of computer users who are likely to value aesthetics	HCI: Human Computer Interface

	in addition to usability, and third previous research suggests that aesthetic perceptions of an interface are highly correlated with perceptions of the interface's ease of use, therefore it appears that users do not perceive these two design items independent. The mechanism that links affective and cognitive evolutions of user interfaces in not clear. Three different processes may induce positive relationships between interface aesthetics and perceived usability. (a) popular stereotyping which associates successful design on one design dimension with successful design of other, less implicit design dimensions. (b) a halo effect may cause carry over of an aesthetic (of not aesthetic) design to perceptions of other design featured. (c)_ an affective response to the design's aesthetics may improve users' mood.	
Guimares, Tor; Martensson, Nis; Stahre, Johan; Igbaria, Magid; 1999 [1]	Empirically testing the impact of manufacturing system complexity on performance New developments in information technology will forever change advanced manufacturing towards a higher level of abstraction and complexity. Higher level of computerization may result in less flexible plants unable to compete in a business environment. Activities during system design, particularly planning and scheduling, mainly rely on human competence. A main set of factors relates to manufacturing processes being supported by a system: the complexity of the supervisory tasks performed, the complexity of the operators' tasks and the complexity of the required operator behaviours. Another major determinant is the complexity of the system. Supervisory (operator) task complexity The supervisor(operator) must be able to interact with the system, he acts in five roles. 1. Planning 2. Teaching/Programming 3. Monitoring	

	4. Intervening 5. Learning Task complexity can be classified in three categories: 1. Difficult by general opinion 2. Only some operators consider difficult 3. Few operators can perform There are three levels of operator behaviour complexity: 1. Skill-based behaviour 2. Rule-based behaviour 3. Knowledge-based behaviour Task complexity and operator behaviour complexity do not contribute significantly to system performance. Supervisory task complexity is inversely related to system performance System complexity is inversely related to system performance Operator training effectiveness will moderate the relationships between each independent variable and system performance The quality of the MMI will moderate the relationships between each independent variable and system performance. Operator task complexity is inversely related to system performance Operator behaviour complexity is inversely related to system performance. While effective man/machine interfaces seem to help reduce the negative influence of complex supervisory roles, operator tasks, operator behaviour, as well as system complexity, training had a relatively minor impact on just the effects of supervisory role complexity.	
Ngo, D.C.L.; Teo, L.S.; Byrne, J.G.; 1999 [5]	Formalising guidelines for the design of screen layouts The role of aesthetics in human affairs has been widely documented. Careful application of aesthetic concepts can aid: • Acceptability • Learnablilty • Comprehensibility	

	• Productivity There are four metrics for alphanumeric displays: overall density, local density, grouping, and layout complexity. There are fourteen measures for graphic displays: 1. Balance Distribution of optical weight. 2. Equilibrium Stabilisation, midway centre of suspension 3. Symmetry Axial duplication 4. Sequence Arrangement of objects in a layout in a way that facilitates the movement of the eye.. from left to right, top to bottom Certain (large) objects attract the eye 5. Cohesion Similar aspect ratios promote cohesion 6. Unity Unity is Coherence, a totality of elements that is visually all one piece, elements seem to belong together 7. Proportion Cultures preferred proportional relationships. Some shapes are considered more aesthetically pleasing then others 8. Simplicity Directness and singleness of form. Achieved by optimising the number of elements and minimising the alignment points 9. Density The extent to which the screen is covered with objects 10. Regularity Uniformity of elements based on some principle or plan 11. Economy Careful and discreet use of display elements 12. Homogenity Even distribution of objects over the quadrants 13. Rhythm Regular patterns of changes in the elements 14. Order and complexity Aggregate of all measures mentioned above	

Parush, Avraham; Nadir, Ronen, Shtub, Avraham; 1998 [10]	Evaluating the Layout of Graphical User Interface Screens: Validation of a Numerical Computerized Model The model consists of design guidelines concerning screen factors – element size, local density, alignment, and grouping – and produces a complexity score for a given screen. Participants' search time is measured. Very well designed screens resulted in shorter search times. The "quality" of visual layout can affect both the time taken to learn how to use a software program as well the time taken to accurately perform tasks requiring this initial visual orientation The basic factors in visual layout are: global density, local density, grouping and complexity. Complexity is the degree to which the organization of the element follows a predictable visual scheme.. Main factors presented in the model are: Size, Local Density, Global Density and Grouping.	
Ngo, David Chek Ling; Byrne, John G.; 1998 [6]	Aesthetic measures for Screen Design Paying attention to aesthetics is important in making interfaces easy to learn and use. Aesthetics affect people's perceptions of the usability of a computer system which may in turn, influence their attitudes towards the system. If the system presents the information in a confusing manner, the user may decide not to use the system at all. Very high correlations have been found between the users' perception of interface aesthetics and usability and acceptability. Some people doubt that aesthetic value can be measured. Five aesthetic measures for graphic displays are defined in this paper: balance, equilibrium, symmetry, sequence, order and complexity. The measure of screen order and complexity is derived based on the work of Brikhoff (Birkhoff, G.D., Aesthetic Measure, Harvard University Press, Cambridge, Mass., 1933)	

Wei, Zhi-Gang, 1998 [16]	Mental Load end Performance at Different Automation Levels The research presented in this tesis, concentrates on the influence of various levels of automation on the perfromance and mental load of human operators. The tasks that human operators perform have been changed from directly operating the process to monitoring and managing the process and controlling the product quality. The operator risks to lose his operational skill and his involvement in operating the process, so he might not be able to adequately solve problems. A quantative measure for Degree of Automation (DofA) is derived, it has been defined as a function of the number and the nature of tasks to be automated. The study is a preliminary step to develop a measurement for the degree of automation in the design of operator supervisory tasks or during system evaluation. The following relations/insights are obtained: • The relationship between performance and DofA can be expressed as a second order polynomal • The performance increases as the DofA increases, but levels off at large DofA values • The relationship between mental load and DofA can be expressed as a first order polynomal. The mental load decreases as the DofA inceases • When the operator controls a partly-automated system the overall mental load that he/she perceives may be predicted from the mental load measurement obtained for each task in a situation where all tasks are manual performed. • If automation fails and the DofA drops, the mental load first increases a lot, and later decreases and levels off to the level of the unautomated system • If automation fails and the DofA drops, the performance first degrades a lot, and later increases and levels off to the level of the unautomated system • A well-performing operator not only focuses on the manually performed tasks, but also supervises the automated tasks.	DofA : Degree of Automation

	• The operator takes more frequent visual samples, rather than longer fixation time, on the automatic control systems of highly ctrotocal tasks, compared to the automatic systems of tasks with a low task criticality • The monitoring of automated, highly critical tasks will contribute more to the overall mental load then monitoring tasks with a low criticality	
Vanderdonckt, Jean; Gillo, Xavier; 1994 [13]	Visual Techniques for Traditional and Multimedia Layouts When the designer sketches a user interface, he first selects the appropriate interaction and interactive objects. The second concern is determining the basic layout of the selected objects. Conventional layout grids consist of a set of parallel horizontal and vertical lines that devide the layout into units that have visual and conceptual integrity. Today, modern user interfaces, especially multimedia interfaces, no longer consist of horizontal and vertical lines, so the concept of layout grid should be extended to layout frame. Instead, the layout may be based on other lines (oblique, discontinuous), convex shapes, planes or volumes. To help the designer, several visual techniques are introduced. • physical techniques: balance, symmetry, regularity, alignment, proportion, horizontality • composition techniques: simplicity, economy, understatement, neutrality, singularity, positivity ans transparency • association and disassociation techniques: unity, repartition, grouping and sparing • ordering techniques: consistency, predictability, sequentially and continuity • photographic techniques: sharpness, roundness, stability, levelling, activeness, subtlety, representation, realism and flatness Physical: Balance is a search for equilibrium along a vertical or horizontal axis. Symmetry consists of duplicating the visual image of IO along a horizontal and/or vertical axis. Regularity establishes uniformity of IO placing	IO: interaction object layout grid / layout frame P balanced vs unbalanced symmetric vs asymmetric regular vs irregular alignment vs misalignment proportioned vs disproportioned horizontal vs vertical C

	Alignment is reducing the number of vertical and horizontal alignment points Proportion strives for an aesthetically appealing ratio between the dimensions of IO Horizontality shows that layouts with greater length then height are predominant. Composition: Simplicity is directness and singleness of layout, free from secondary complications or sophistications. Economy is the frugal and judicious use of IO (NL: sober/spaarzaam en verstandig/afgewogen) Understatement is equivalent to the couple economy-intricacy, but in the domain of intellectual, mental representation rather than physical, spatial representation. Neutrality cuts every resistance, repulsion, or even belligerency of the layout's viewer. Singularity is the focus of a layout on one separate and solitary IO, unsupported by any other IO. Negativity displays IO in dark colours in a light background Transparency means a visual layout where IO, superseded by another IO, can still stay visible behind of through them. Unity is the placement of individual IO into one totality Repartition proposes to share IO among the four quadrants of the layout as systematically as possible Grouping creates a circumstance of give and take of relative interaction Sparing looks for avoiding cluttered or overcrowded layouts. It suggests keeping the visual loading of a layout within reasonable boundaries. Ordering: Consistency expresses visual compatibility with the subject, for development of a layout whose IO are dominated by one sound, uniform, constant thematic Predictability indicates whether IO are placed according to some order or plant that is highly conventional and recognisable. Sequentiality is a plan of layout that is arranged in a logical, rhythmic, expected order. Continuity uninterrupts the visual connections	simple vs complex economic vs uneconomic (or intricate) Understated vs exaggerated Neutral vs accentuated Singular vs juxtaposed (naast elkaar plaatsen; (doen) contrasteren) Negative and positive transparent vs opaque unitary vs fragmented equally repartitioned vs sprinkled grouped vs splitted uncluttered vs cluttered O consistent vs varied predictable vs spontaneous sequential vs random continuous vs episodic P shapr vs diffused round vs angular stable vs unstable active vs passive

	existing between the IO Photographic: Sharpness can be interpreted as distinctiveness or as hard outlined, hard edges, distinct margins Roundness is the preference for round IO Stability is the expression of preference for IO that have clear base to rest on Levelling automatically establishes balance through artefacts. IO are laid out so that balance axis will stand out. Activeness reflects motion through explicit representation or implicit suggestion. Subtelty makes a fine distinction, shinning any obviousness and energy of purpose. Often synonym with ingeniosity since it requires delicate, highly refined IO. Realism is the natural technique of camera Flatness doesn't use any technique for providing perspective, so erasing the natural feeling of dimension and space. Apply physical visual techniques for traditional layouts an especially when the layout is text-dominant rather than graphic-dominant; balance is the most important visual techniques to achieve. Visual techniques cannot be controlled or parameterised conclusion: • multimedia IO could be qualified with attributes 1. visual techniques that have been exposed are purely visual, resting on aesthetic and psychological factors and disregarding human deriving from user requirements. 2. effective visual design should rely on task analysis 3. visual techniques only form one of the very first steps towards a better understanding of effective visual layouts 4. applying these techniques in general fashion doesn't preclude that users have to be avoided 5. other techniques shouldn't be left out	subtle vs bold real vs distorted flat vs depth

Keyes, Elizabeth; Krull, Robert; 1992 [3]	User information Processing Strategies and Online Visual Structure Previous work: Users' mental load can be reduced if screen chunks are placed in order expected by the user, based on simple rules. Users are able to handle higher level of complexity if information is layered or staged at multiple levels. We try to explain how people handle complexity. If users have expended too much of preliminary tasks of orientation, navigation, deciding what is important and where to focus their attention, they may have little energy left for the main tasks. Users focus their attention by concentrating on important information and ignore in they perceive to be irrelevant. Several categories of variables effect information load: complexity (the number of information items and the number of relationships among them), conflict, ambiguity and multiple meaning, and instability. Design rules about placement: • items placed together are related (users can handle about three to five chunks a panel, and that screen densities should be held about 30% of characters used • Items with the same shape are related and should be read together • Information that has the same vertical alignment on the screen will be seen as a group • Items should be chunked in a consistent, meaningful way in sequences of panels (two design: strategies ..individual panel, or all panels) • Items are to e scanned in the normal reading order, from top down and from left to right. • Proportional emphasis establishes a logical hierarchy Dual-level access: First (macro level), users scan the field of elements, look for identification and relationships, and decide where to focus their attention. Second (micro level), user focus on what's within a particular chunk	

	The Grid can be used for visual orientation by five an overall structure, by managing informational and visual complexity. A well designed grid: • is the first level of information structure, organization and simplification • creates macro-level chunks for specific types and levels of information through defined spatial zones • defines placements and spatial relationships of sub-levels of information within each macro-chunk • creates consistency in placement and resulting visual patterns • creates a framework for multiple cueing Information theory suggests that chunking and consistent placement of information helps users.	
Tullis, Thomas S; 1988 [12]	Chapter 18: Screen Design (of Handbook of Human-Computer Interaction) The content and format of the information presented will always have a substantial impact on the usefulness and efficiency of the interaction. If the system doesn't present the information the user needs, or presents it in an unusual or confusing manner, the user may decide not to use the system, or the user's performance may be bad. Te knowledge about screen design is derived from several sources: * basing psychological research, * human factors studies, * experiences of designers and users, * graphic design experience. The total amount of information should be minimized by presenting only what is necessary to the user. As long as the information necessary to perform the task is present, human performance deteriorates with increasing display density. Some concerns: • make appropriate use of abbreviations • avoid unnecessary detail • use concise wording Group inforation using: * color and * graphical boundary Highlighting techniques such as brightness or boldness, underlining and flashing can aid the user in locating important information. The screen layout should make it possible to easily	

	find any information, by considering the placement and sequence of information. This can be sorted by: • Sequence of Use • Conventional Usage • Importance • Frequency of Use • Generality / Specificity • Alphabetical • Chronological Series of related data elements should be presented vertically in a list rather than horizontally. Several other things could be considered: • Indentation • Label/data relationships • Process associations • Symmetry Graphics can be used representing real word Images, or Complex, Real world systems. It is important to determine standard meanings for symbols. Numerical data is often easier to interpret if presented graphically, unless the task requires precise computations. Icons can be classified using the following characteristics: A=Abstract symbols, CA= Concrete analogy associated with action, CO=Concrete object operated on. The screen design process is iterative, and had the following elements (not in any order:) • Requirements and Constraint Analysis • Task Analysis and Scenario Development • Development of Design Rules • Development of Implementation Philosophy • Early design, Prototyping and Evaluation • Full-scale Prototyping and Implementation	

Overall display properties		
a_{layout}	area laout	cm^2
b_{frame}	width of frame	cm
b_{layout}	width layout	cm
h_{frame}	height of frame	cm
h_{layout}	height layout	cm
n	number of objects (on the frame)	
n_{hap}	number of horizontal alignment points	
n_j	number of objects in quadrant j	
$n_{spacing}$	number of distinct distances between column and row starting points	
n_{vap}	number of vertical alignment points	
Object properties		
a_i	area	cm^2
b_{ij} or b_i	width of object ij	cm
c	color	
d	distance to central axis	cm
h_{ij} or h_i	height of object ij	cm
i	object number	
j	location ; j=L, R, T, B (left, right, top, bottom)	
j	location ; j=UL, UR, LL, LR (upper left, etc)	
s	shape factor	
x_c	x-coordinate of center of frame	cm
x_i	x-coordinate of object i	cm
y_c	y-coordinate of center of frame	cm
y_i	y-coordinate of object y	cm
Measures and submeasures		
a_{max}	maximum area	cm^2
B	overall balance	
B_{hor} B_{vert}	horizontal and vertical balance	
B_j	total width	cm
B_j'	normalized total width	
C	cohesion	
C_{fl}	relative measure of the ratios of the layout and screen	
C_{lo}	average relative ratio of object and layout	

CO	complexity	
D	density	
E	overall equilibrium	
EC	Economy	
E_x E_y	Equilibrium components along x and y axes	
f_i	relative ratio of the objects and layout	
H_j	total height	cm
H_j'	normalized total height	
M_{layout}	difference between the proportions of the layout and the closest proportional shape	
P	proportion	
p_i	ratio of a separate object	
p_j	aesthetically most pleasing ratios	
p_{layout}	ratio of the layout	
P_{obj}	mean difference between the object and the closest proportional shape	
R	overall regularity	
$R_{alignment}$	alignment regularity	
RH	thythm	
RH_{area}	area rhythm factor	
RH_x	x-axis rhythm factor	
RH_y	y-axis rhythm factor	
R_j	total radius	
R_j'	normalized total radius	
$R_{spacing}$	spacing regularity	
SM	Simplicity	
SYM	overall symmetry	
SYM_{vert} SYM_{hor} SYM_{rad}	vertical, horizontal and radial symmetry	
t_{fl}	ratio factor of layout and fram	
t_i	object/layout ratio	
U	Unity	
U_{form}	extent to which the objects are related in size	
U_{space}	relative measure of the space between groups	
w	weighing	
W	number of different ways a group of n objects can be arranged for the four quadrants	
w_j	total weighting of a sector	
W_{max}	maximum of W (when the n objects are evenly allocated to the various quadrants of the screen)	
X_j	total displacement in x direction	
X_j'	normalized total displacement in x direction	
Y_j	total displacement in y direction	
Y_j'	normalized total displacement in y direction	
Θ_j	total location tangens	
Θ_j'	normalized total location tangens	

Balance

Balance is the distribution of optical weight. Some interfaces seem to fall over to the left or right, or are very heavy at the top.

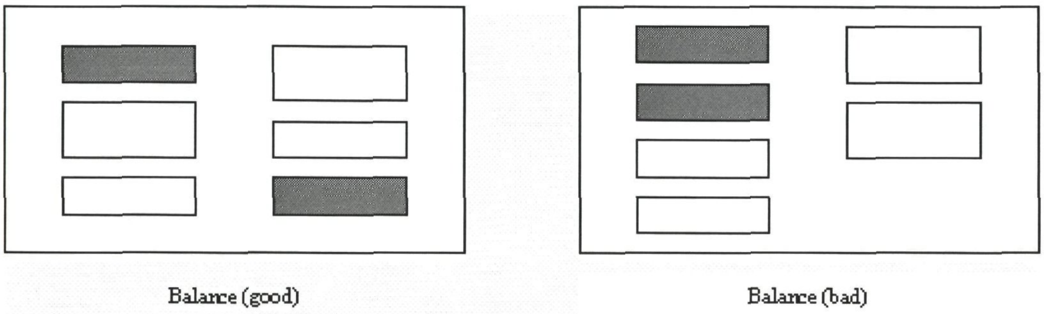
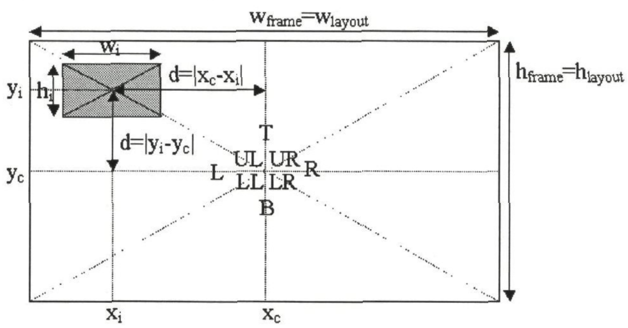


Figure 1: Example of good (a) and bad (b) in balance study



j = location of object i ; L,R,T,B for left, right, top, bottom
w is the weighting of an object

$$w = d \circ \left(\frac{a}{a_{\max}} \right) \qquad \text{(eventually } w = d \circ \left(\frac{a}{a_{\max}} + \left| c - c_{frame} \right| + s \right) \text{)}$$

for j=t and j=b, d is the horizontal distance, otherwise, d is the vertical distance.

$$w_j = \sum_{i=1}^{n_j} w$$

these are the total weightings of the objects on the left, right, top and bottom of the objects. The normalized differences between the total weightings are an indication for horizontal and vertical balance.

$$B_{hor} = \left| \frac{w_T - w_B}{\max\left(\left|w_T\right|, \left|w_B\right|\right)} \right| \in [0,1] \qquad B_{vert} = \left| \frac{w_L - w_R}{\max\left(\left|w_R\right|, \left|w_L\right|\right)} \right| \in [0,1]$$

The overall balance is the average of the horizontal and vertical balance. This value is scaled to the interval [0,100]

$$B = \frac{B_{hor} + B_{vert}}{2} \circ 100 \in [0,100]$$

Equilibrium

Equilibrium is stabilization or midway center of suspension, and determines if the center of the layout is the same as the center of the frame.

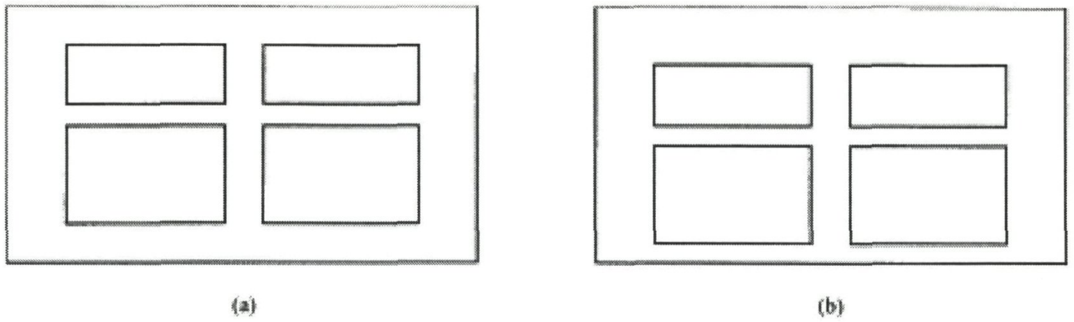
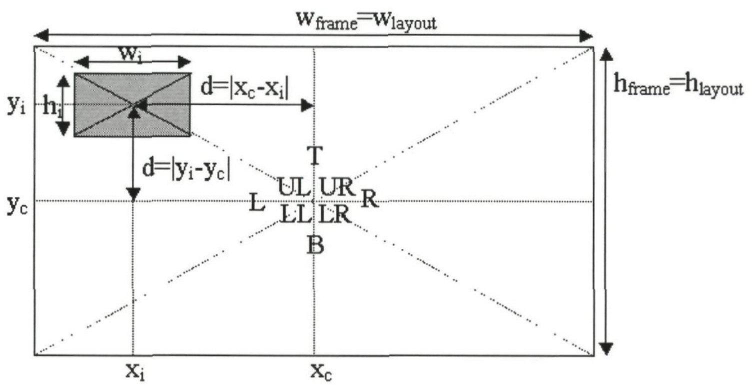


Figure: Example of good (a) and bad (b) in equilibrium study



The equilibrium components along the x and y axes are:

$$E_x = \frac{2\sum_{i=1}^{n} a_i (x_i - x_c)}{n \circ b_{frame} \circ \sum_{i=1}^{n} a_i} \in [0,1] \qquad E_y = \frac{2\sum_{i=1}^{n} a_i (y_i - y_c)}{n \circ h_{frame} \circ \sum_{i=1}^{n} a_i} \in [0,1]$$

These two can be combined to the overall equilibrium measure:

$$E = \frac{|E_x| + |E_y|}{2} \circ 100 \in [0,100]$$

Symmetry

The extent to which the screen is symmetrical in three dimensions: horizontal, vertical and diagonal.

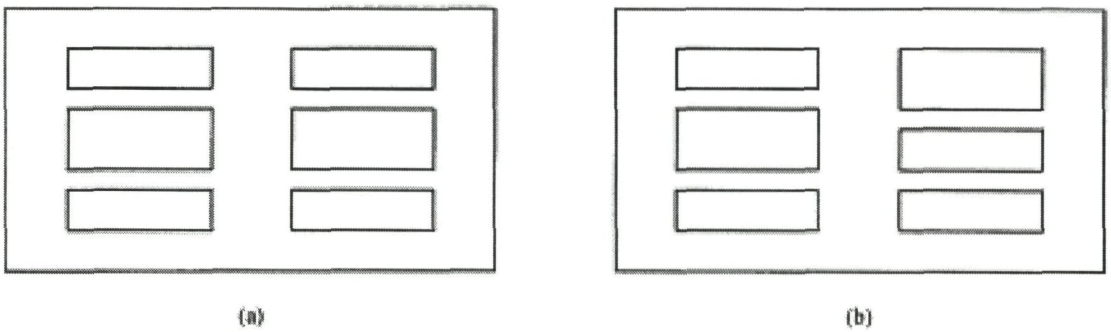
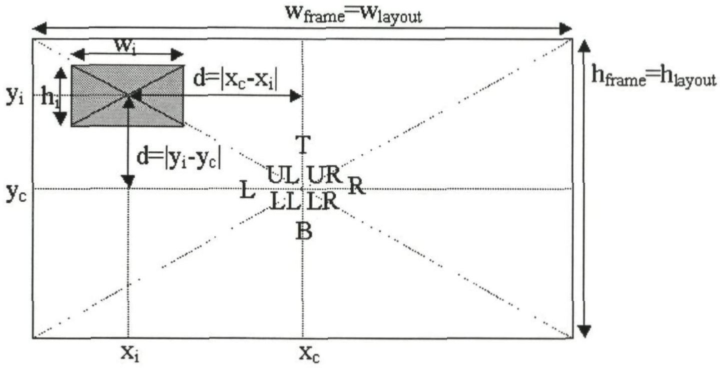


Figure: Example of good (a) and bad (b) in symmetry study



j = UL, UR, LL, LR

To be able to calculate the symmetry, the total x and y distance to the centerline (X, Y), the total height and width (H,B or h,w), the total tangens of the angle between horizontal and line connecting the center of the frame with object i (Θ) and the total radius (R) must be known.

$$X_j = \sum_{i=1}^{n_j} \left| x_{ij} - x_c \right| \qquad Y_j = \sum_{i=1}^{n_j} \left| y_{ij} - y_c \right|$$

$$H_j = \sum_{i=1}^{n_j} h_{ij} \qquad B_j = \sum_{i=1}^{n_j} b_{ij} = \sum_{i=1}^{n_j} w_{ij}$$

$$\Theta_j = \sum_{i=1}^{n_j} \left| \frac{y_{ij} - y_c}{x_{ij} - x_c} \right|$$

$$R_j = \sum_{i=1}^{n_j} \sqrt{(x_{ij} - x_c)^2 + (y_{ij} - y_c)^2}$$

X_j', Y_j', H_j', B_j', Θ_j' and R_j' are the normalized values of the things mentioned above for example:

$$X'_j = \frac{X_j}{\max(X_{UL}, X_{UR}, X_{LL}, X_{LR})} \in [0,1]$$

Now, horizontal, vertical and radial symmetry can be calculated:

$$SYM_{vert} = \frac{\begin{array}{l}|X'_{UL}-X'_{UR}|+|X'_{LL}-X'_{LR}|+|Y'_{UL}-Y'_{UR}|+|Y'_{LL}-Y'_{LR}|+ \\ |H'_{UL}-H'_{UR}|+|H'_{LL}-H'_{LR}|+|B'_{UL}-B'_{UR}|+|B'_{LL}-B'_{LR}|+ \\ |\Theta'_{UL}-\Theta'_{UR}|+|\Theta'_{LL}-\Theta'_{LR}|+|R'_{UL}-R'_{UR}|+|R'_{LL}-R'_{LR}|\end{array}}{12} \in [0,1]$$

$$SYM_{hor} = \frac{\begin{array}{l}|X'_{UL}-X'_{LL}|+|X'_{UR}-X'_{LR}|+|Y'_{UL}-Y'_{LL}|+|Y'_{UR}-Y'_{LR}|+ \\ |H'_{UL}-H'_{LL}|+|H'_{UR}-H'_{LR}|+|B'_{UL}-B'_{LL}|+|B'_{UR}-B'_{LR}|+ \\ |\Theta'_{UL}-\Theta'_{LL}|+|\Theta'_{UR}-\Theta'_{LR}|+|R'_{UL}-R'_{LL}|+|R'_{UR}-R'_{LR}|\end{array}}{12} \in [0,1]$$

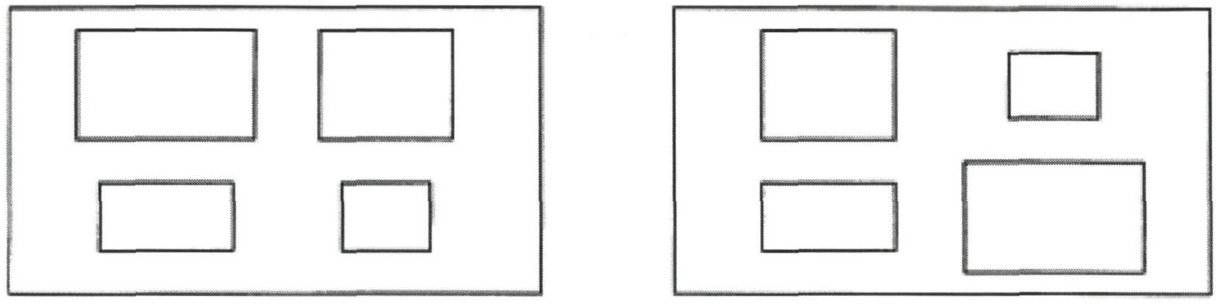
$$SYMM_{rad} = \frac{\begin{array}{l}|X'_{UL}-X'_{LR}|+|X'_{UR}-X'_{LL}|+|Y'_{UL}-Y'_{LR}|+|Y'_{UR}-Y'_{LL}|+ \\ |H'_{UL}-H'_{LR}|+|H'_{UR}-H'_{LL}|+|B'_{UL}-B'_{LR}|+|B'_{UR}-B'_{LL}|+ \\ |\Theta'_{UL}-\Theta'_{LR}|+|\Theta'_{UR}-\Theta'_{LL}|+|R'_{UL}-R'_{LR}|+|R'_{UR}-R'_{LL}|\end{array}}{12} \in [0,1]$$

The overall symmetry is the average of the three above, here scaled to the [0,100] interval.

$$SYM = \frac{|SYM_{vert}|+|SYM_{hor}|+|SYM_{rad}|}{3} \circ 100$$

Sequence

Arrangement of objects in a way that facilitaits the movement of the eye.



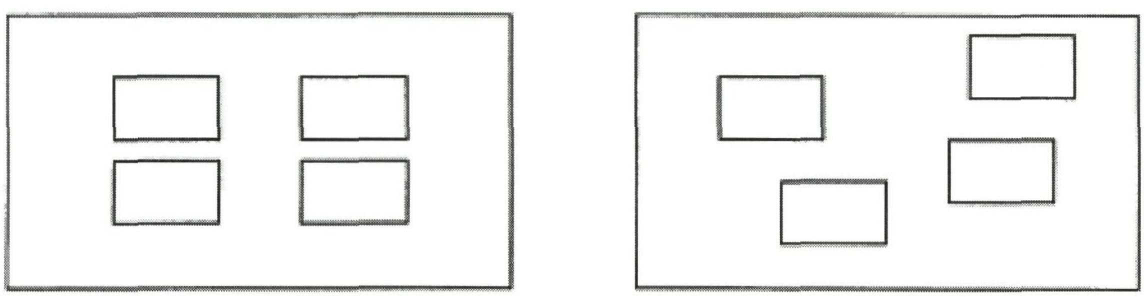
(a) (b)

Figure: Example of good (a) and bad in sequence study

The measure Ngo suggest for sequence isn't applicable in this situation, so the sequence measure is omitted here.

Simplicity

Directness and singleness of form.



(a) (b)

Figure: Example of good (a) and bad (b) in simplicity study

The simplicity depends on the number of objects and the number of allocation points.

$$SM = \left(1 - \frac{3}{n_{vap} + n_{hap} + n}\right) \circ 100 \qquad \in [0,100]$$

n_{vap} is the number of vertical alignment points, n_{hap} the number of horizontal alignment points and n is the total number of objects.

Density

The extent to which the screen is covered with items.

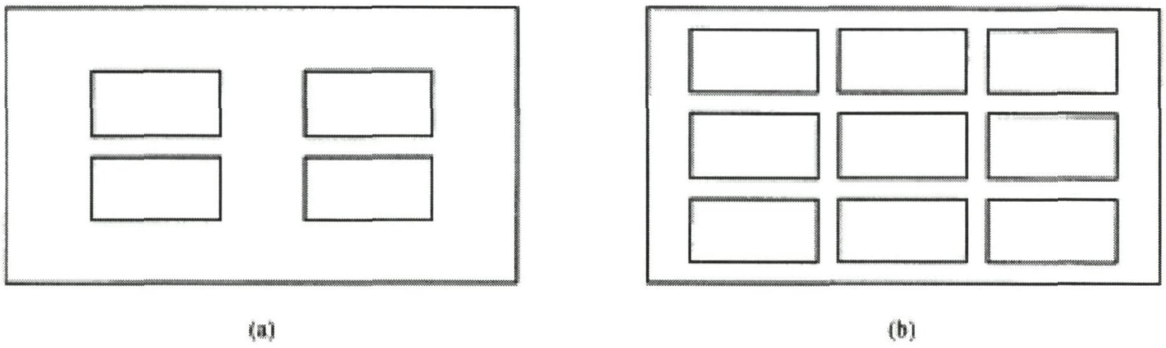


Figure: Example of good (a) and bad (b) in density study

The density is dependent on several (total) areas

$$D = 200 \circ \left| 0.5 - \frac{\sum_{i=1}^{n} a_i}{a_{frame}} \right| \in [0,100]$$

Assumed that the optimum screen density level for grapic screens is 50% (=0.5)
a_i is the area of object i and a_{frame} is the area of the frame

Regularity

Uniformity of elements based on some priniciple or plan.

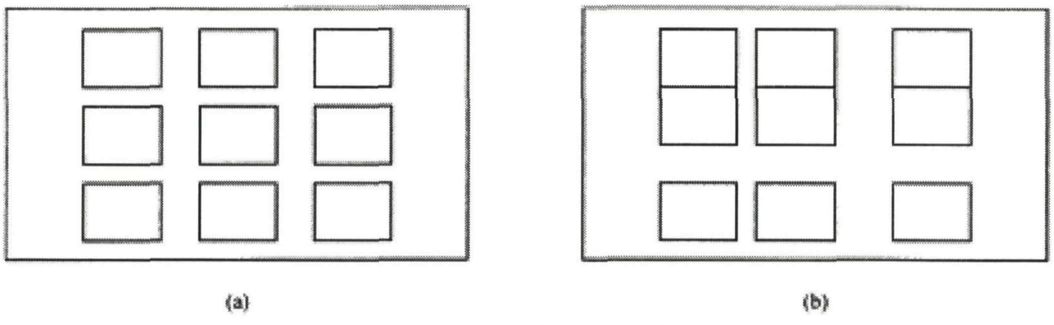


Figure: Example of good (a) and bad (b) in regularity study

The alignment can be calculated using the number of objects, the number of spacing distances and the numbers of alignment points. The alignment regularity is defined as:

$$R_{alignment} = \begin{cases} 0 & \text{if n=1} \\ \dfrac{n_{vap} + n_{hap}}{2n} \circ 100 & \text{otherwise} \end{cases} \in [0,100]$$

n_{vap} is the number of vertical alignment points, n_{hap} the number of horizontal alignment points and n is the total number of objects.

The spacing regularity is defined as:

$$R_{spacing} \quad \begin{cases} 0 & \text{if n=1} \\ \dfrac{n_{spacing} - 1}{2(n-1)} \circ 100 & \text{otherwise} \end{cases} \in [0,100]$$

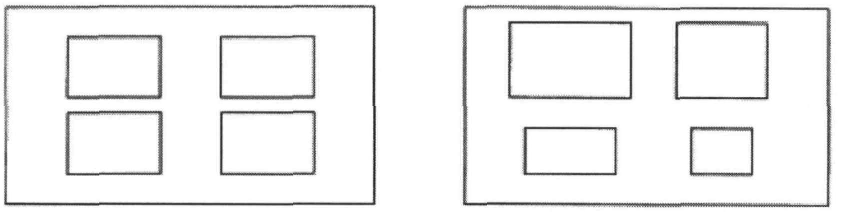
$n_{spacing}$ is the number of distinct distances between column and row starting points

The overall regularity can now be calculated:

$$R = \frac{|R_{alignment}| + |R_{spacing}|}{2} \in [0,100]$$

Economy

Economy is the careful and discrete use of screen elements to get the message across as simple as possible.



(a) (b)

Figure 2: Example of good (a) and bad (b) in economy study

$$EC = \left(1 - \frac{1}{n_{size}}\right) \circ 100$$

n_{size} is the number of different sizes used in the display.

Homogenity

Even distribution of objects over the quadrants.

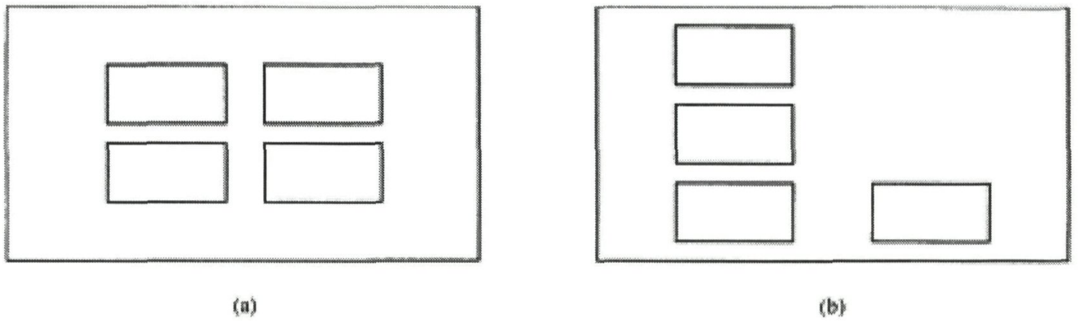


Figure: Example of good (a) and bad (b) in homogenity study

W is the number of different ways a group of n objects can be arranged for the four quadrants when n_j is the total number of objects in quadrant j.

$$W = \frac{n!}{\prod_{j=UL,UR,LL,LR} n_j} = \frac{n!}{n_{UL}!n_{UR}!n_{LL}!n_{LR}!}$$

W_{max} is maximum when the n objects are evenly allocated to the various quadrants of the screen, as compared to more or less uneven allocations among the quadrants

$$W_{max} = \frac{n!}{\frac{n}{4}!\frac{n}{4}!\frac{n}{4}!\frac{n}{4}!} = \frac{n!}{\left(\frac{n}{4}!\right)^4}$$

Now, the homogeneity measure can be calculated:

$$HM = \left(1 - \frac{W}{W_{max}}\right) \circ 100 \qquad \in [1,100]$$

Unity

Coherence, totality of elements that is visually one piece, elements seem to belong together.

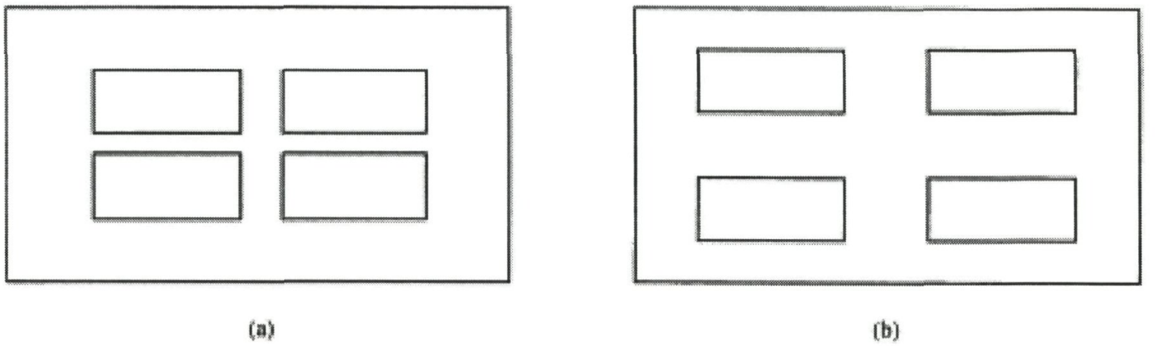


Figure: Example of good (a) and bad (b) in unity study

a_i, a_{layout} and a_{frame} are the areas of object i, the layout and the frame. n_{size} is the number of the sizes used, and n is the number of objects on the frame.
The extent to which the objects are related in form can now be calculated.

$$U_{form} = \frac{n_{size} - 1}{n} \circ 100 \qquad \in [1,100]$$

The relative measure of space between the groups is defined as:

$$U_{space} = \frac{a_{layout} - \sum_{i=1}^{n} a_i}{a_{frame} - \sum_{i=1}^{n} a_i} \circ 100 \qquad \in [1,100]$$

Together they result in the unity measure :

$$U = \frac{\left|U_{form}\right| + \left|U_{space}\right|}{2} \qquad \in [1,100]$$

Cohesion

Similarity of aspect ratios.

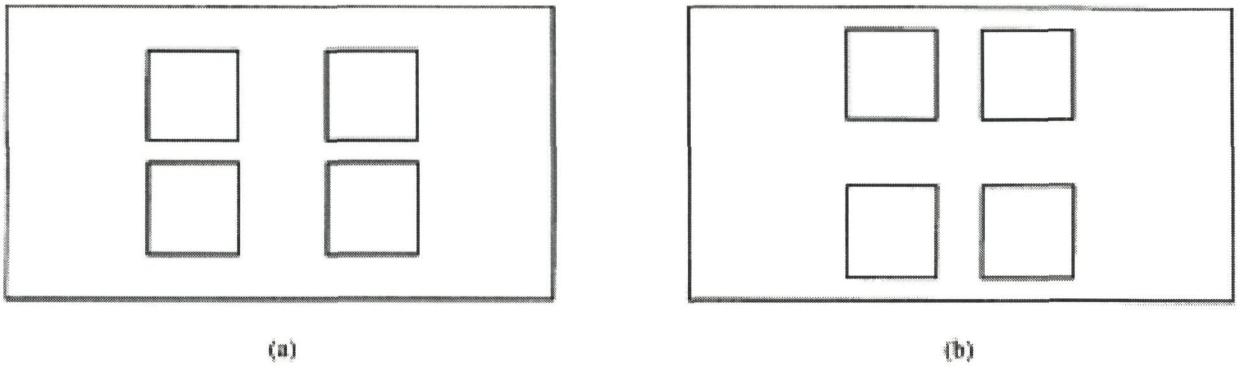


Figure 3: Example of good (a) and bad (b) in cohesion study

The ratio factor of layout and frame is:

$$t_{fl} = \frac{h_{layout}/b_{layout}}{h_{frame}/b_{frame}}$$

h_{layout} and b_{layout} are the height and width of the layout, and h_{frame} and b_{frame} are the height and width of the frame.
The relative measure of the ratios of the layout and screen is defined as:

$$C_{fl} = \begin{cases} t_{fl} & \text{if } t_{fl} \le 1 \\ \dfrac{1}{t_{fl}} & \text{otherwise} \end{cases}$$

The object/layout ratio is:

$$t_i = \frac{h_i/b_i}{h_{layout}/b_{layout}}$$

h_i and b_i are the height and width of object i

The relative ratio of the objects and layout is defined as:

$$f_i = \begin{cases} t_i & \text{if } t_i \le 1 \\ \dfrac{1}{t_i} & \text{otherwise} \end{cases}$$

This can be calculated for all objects and results in the average relative ratio of object and layout:

$$C_{lo} = \frac{\sum_{i=1}^{n} f_i}{n}$$

The cohesion measure now is :

$$C = \frac{(1 - C_{fl}) + (1 - C_{lo})}{2} \circ 100 \qquad \in [0,100]$$

Proportion

Some rectangle aspect ratios are aesthetically more pleasing.

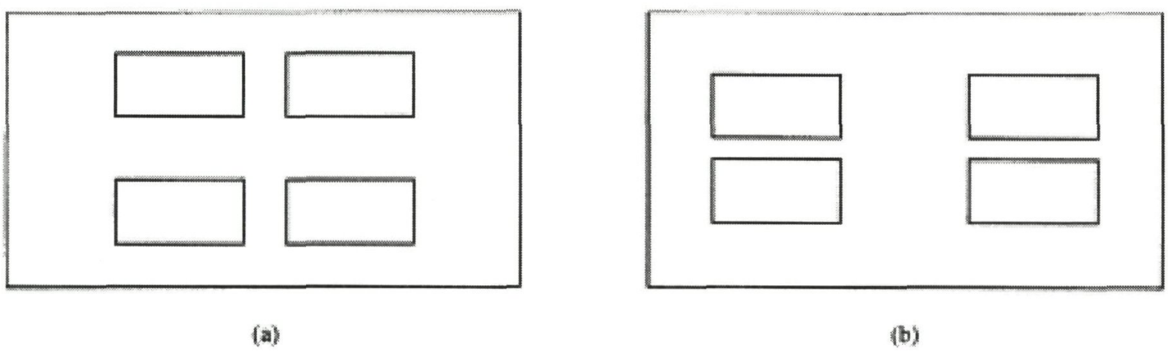


Figure: Example of good (a) and bad (b) in proportion study

The aesthetically most pleasing ratios are:

$$p_j = \left\{ \frac{1}{1}, \frac{1}{1.414}, \frac{1}{1.618}, \frac{1}{1.732}, \frac{1}{2} \right\} \qquad (j=sq,r2,gr,r3,ds)$$

The ratio of the layout is :

$$p_{layout} = \begin{cases} \dfrac{h_{layout}}{b_{layout}} & \text{if} \quad b_{layout} > h_{layout} \\ \dfrac{b_{layout}}{h_{layout}} & \text{if} \quad h_{layout} > b_{layout} \end{cases} \qquad \in [0,1]$$

h_{layout} and b_{layout} are the height and width of the layout

The difference between the proportions of the layout and the closest proportional shape is:

$$M_{layout} = \left(\frac{\min(|p_j - p_{layout}|)}{0.5} \right) \circ 100$$

The ratio of a separate object is:

$$p_i = \begin{cases} \dfrac{h_i}{b_i} & \text{if} \quad b_i > h_i \\ \dfrac{b_i}{h_i} & \text{if} \quad h_i > b_i \end{cases} \qquad \in [0,1]$$

h_i and b_i are the height and width of object i

The mean difference between the object and the closest proportional shape is:

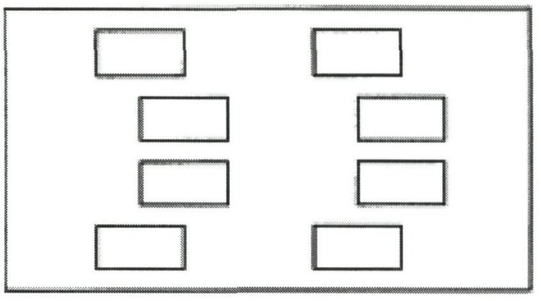
$$P_{obj} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\min(|p_j - p_i|)}{0.5} \right) \circ 100 \qquad \in [0,100]$$

The measure for proportion now is:

$$P = \frac{|P_{obj} + P_{layout}|}{2} \qquad \in [0,100]$$

Rhythm

Regular patterns of changes in the elements.


(a)

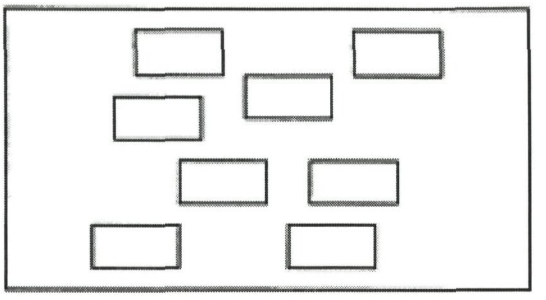

(b)

Figure: Example of good (a) and bad (b) in rhythm study

j = UL, UR, LL, LR (for upper left, upper right, lower left and lower right)
To calculate the rhythm, the displacement and area measures are needed:

$$X_j = \sum_{i=1}^{n_j} |x_{ij} - x_c| \qquad Y_j = \sum_{i=1}^{n_j} |y_{ij} - y_c|$$

$$A_j = \sum_{i=1}^{n_j} a_{ij}$$

X_j', Y_j', H_j', B_j', Θ_j' and R_j' are the normalized values of the things mentioned above for example:

$$X'_j = \frac{X_j}{\max(X_{UL}, X_{UR}, X_{LL}, X_{LR})} \in [0,1]$$

The rhythm components of x-axits, y-axis and area are :

$$RH_x = \frac{|X'_{UL} - X'_{UR}| + |X'_{UL} - X'_{LR}| + |X'_{UL} - X'_{LL}| + |X'_{UR} - X'_{LR}| + |X'_{UR} - X'_{LL}| + |X'_{LR} - X'_{LL}|}{6}$$

$$RH_Y = \frac{|Y'_{UL} - Y'_{UR}| + |Y'_{UL} - Y'_{LR}| + |Y'_{UL} - Y'_{LL}| + |Y'_{UR} - Y'_{LR}| + |Y'_{UR} - Y'_{LL}| + |Y'_{LR} - Y'_{LL}|}{6}$$

$$RH_{area} = \frac{|A'_{UL} - A'_{UR}| + |A'_{UL} - A'_{LR}| + |A'_{UL} - A'_{LL}| + |A'_{UR} - A'_{LR}| + |A'_{UR} - A'_{LL}| + |A'_{LR} - A'_{LL}|}{6}$$

The rhythm measure is:

$$RH = \frac{|RH_x| + |RH_y| + |RH_{area}|}{3} \circ 100$$

Complexity

The measure for complexity is the mean value of all the factors mentioned above.

$$CO = \frac{B + E + SYM + SM + D + R + EC + HM + U + C + P + RH}{12}$$

Appendix D: Calculation of Ngo-like complexity measure

Balance	Equilibrium	Symmetry	Sequence	Simplicity	Density	Regularity	Economy
47	0	46	0	84	9	45	0
Homogenity	**Unity**	**Cohesion**	**Proportion**	**Rhytm**			**complexity**
33	50	20	11	47			33

number of objects	n	8
number of vertical alignment points	nvap	3
number of horizontal alignment points	nhap	8
number of distinct distances between column and row starting points	nspacing	4
area layout	alayout	197.38
width layout	blayout	14.2
height layout	hlayout	13.9
width of frame	bframe	14.2
height of frame	hframe	13.9
	aframe	197.38
	nsize	1

Sally, old

complexity: 32.7

Balance	Equilibrium	Symmetry	Sequence	Simplicity	Density	Regularity
47	0	46	0	84	9	45
Homogenity	**Unity**	**Cohesion**	**Proportion**	**Rhytm**		
33	50	20	11	47		

		1	2	3	4	5	6	7	8
object number	i	1	2	3	4	5	6	7	8
quadrant		UL	UR	UL	UL	UR	LL	LR	LR
width	bi, bij	1.4	4.4	3.5	3.5	3.5	7.1	1.4	1.4
height	hi, hij	1.4	2	1.4	1.4	1.4	7	9.2	1.4
area	ai	1.96	8.8	4.9	4.9	4.9	49.7	12.88	1.96
x-coordinate	xi	1.4	7.8	2.4	7	11.7	5.6	10.5	12.7
y-coordinate	yi	12.9	12.9	10.8	10.8	10.8	5.4	5.4	5.4
x-coordinate fr	xc	7.1	7.1	7.1	7.1	7.1	7.1	7.1	7.1
y-coordinate fr	yc	6.95	6.95	6.95	6.95	6.95	6.95	6.95	6.95

Balance	47.2								
amax	49.7								
wt	3.686156942	0.224788732	1.381086519	0.236619718	0.690140845	1.153521127	0	0	0
wb	8.821971831	0	0	0	0	0	5.6	2.721126761	0.50084507
wl	7.764225352	0.234647887	0	1.064788732	1.064788732	0	5.4	0	0
wr	4.961287726	0	2.284104628	0	0	1.064788732	0	1.39943662	0.212957746
Bhor	0.582161787								
Bvert	0.361006733								
Equilibrium	0.5								
totaal ai	90								
Ex	-0.005041862	-11.172	6.16	-23.03	-0.49	22.54	-74.55	43.792	10.976
Ey	0.00411271	11.662	52.36	18.865	18.865	18.865	-77.035	-19.964	-3.038
Symmetry	45.6								
Xul	10.5	5.7	0	4.7	0.1	0	0	0	0
Xur	5.3	0	0.7	0	0	4.6	0	0	0
Xll	1.5	0	0	0	0	0	1.5	0	0
Xlr	9	0	0	0	0	0	0	3.4	5.6
Yul	13.65	5.95	0	3.85	3.85	0	0	0	0
Yur	9.8	0	5.95	0	0	3.85	0	0	0
Yll	1.55	0	0	0	0	0	1.55	0	0
Ylr	3.1	0	0	0	0	0	0	1.55	1.55
Hul	4.2	1.4	0	1.4	1.4	0	0	0	0
Hur	3.4	0	2	0	0	1.4	0	0	0
Hll	7	0	0	0	0	0	7	0	0
Hlr	10.6	0	0	0	0	0	0	9.2	1.4
Bul	8.4	1.4	0	3.5	3.5	0	0	0	0
Bur	7.9	0	4.4	0	0	3.5	0	0	0
Bll	7.1	0	0	0	0	0	7.1	0	0
Blr	2.8	0	0	0	0	0	0	1.4	1.4
fi ul	40.36300859	1.043859649	0	0.819148936	38.5	0	0	0	0
fi ur	9.336956522	0	8.5	0	0	0.836956522	0	0	0
fi ll	1.033333333	0	0	0	0	0	1.033333333	0	0
fi lr	0.732668067	0	0	0	0	0	0	0.455882353	0.276785714
Rul	18.16655483	8.239690528	0	6.075565817	3.851298482	0	0	0	0
Rur	11.98957646	0	5.991034969	0	0	5.998541489	0	0	0
Rll	2.156965461	0	0	0	0	0	2.156965461	0	0
Rlr	9.547193627	0	0	0	0	0	0	3.736642878	5.810550748

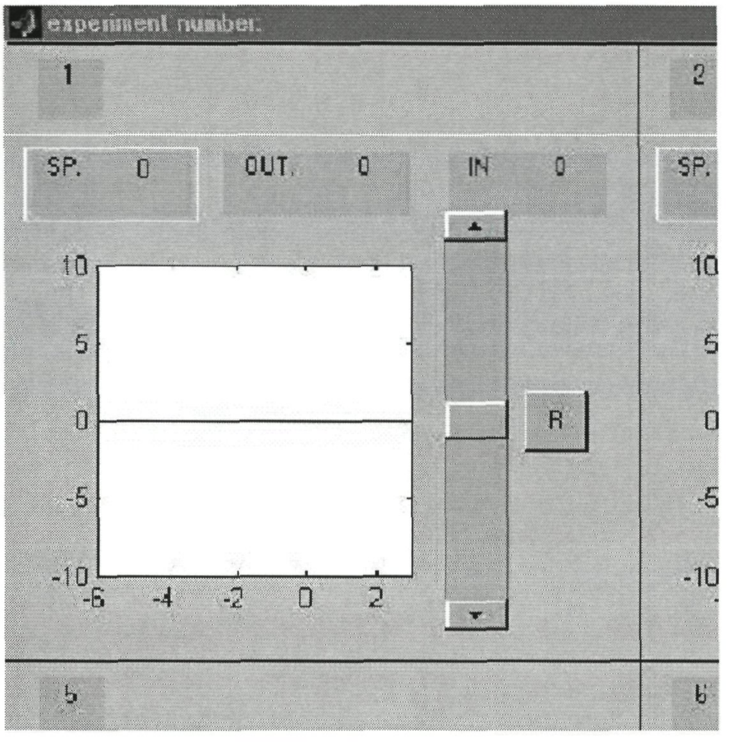
Appendix D: Calculation of Ngo-like complexity measure

normalised:

Xul'	1	Hul'	0.396226415	fi ul'	1
Xur'	0.504761905	Hur'	0.320754717	fi ur'	0.231324593
Xll'	0.142857143	Hll'	0.660377358	fi ll'	0.025600999
Xlr'	0.857142857	Hlr'	1	fi lr'	0.018151969
Yul'	1	Bul'	1	Rul'	1
Yur'	0.717948718	Bur'	0.94047619	Rur'	0.659980749
Yll'	0.113553114	Bll'	0.845238095	Rll'	0.118732775
Ylr'	0.227106227	Blr'	0.333333333	Rlr'	0.525536829
SYMvert	0.342883238	SYMhor	0.541283079	SYMrad	0.48255292

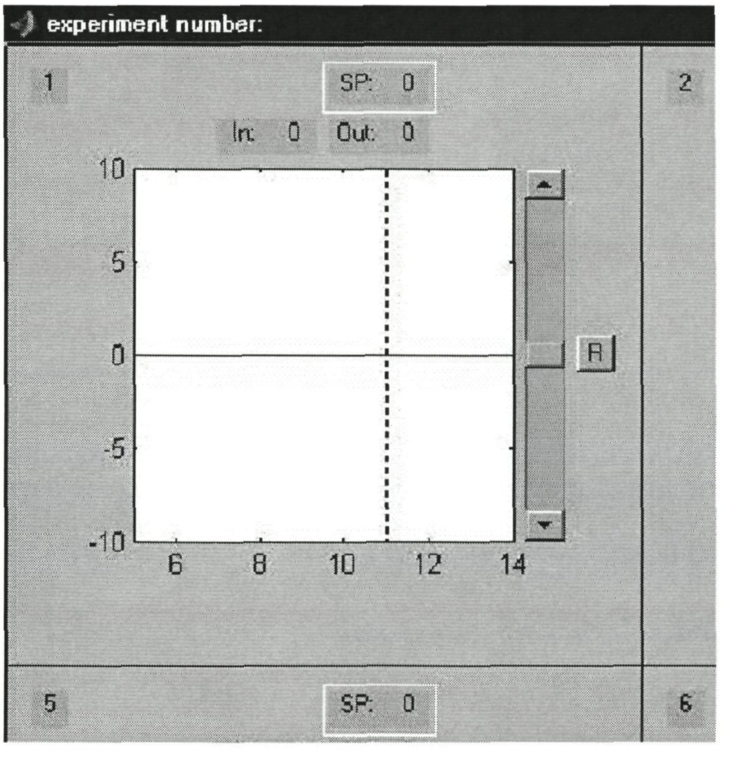
Sequence									
Simplicity	84.2								
Density	8.8	0.45597325							
Regularity	45.1								
Ral	68.75								
Rsp	21.42857143								
Economy	0.0								
Homogenity	33.3								
W	1680								
Wmax	2520								
nul	3	1	0	1	1	0	0	0	0
nur	2	0	1	0	0	1	0	0	0
nll	1	0	0	0	0	0	1	0	0
nlr	2	0	0	0	0	0	0	1	1
Unity	50.0								
Uform	0								
Uspace	100								
Cohesion	20.1								
tfl	1								
Cfl	1								
ti	11.45387275	1.021582734	0.464355788	0.408633094	0.408633094	0.408633094	1.007194245	6.713257965	1.021582734
fi	4.789817662	0.978873239	0.464355788	0.408633094	0.408633094	0.408633094	0.992857143	0.148958971	0.978873239
Clo	0.598727208								
Proportion	11.0	pj	1	0.707213579	0.618046972	0.577367206	0.5		
playout	0.978873239		0.021126761	0.271659661	0.360826268	0.401506034	0.478873239		
Mlayout	4.225352113								
pi		1	0.454545455	0.4	0.4	0.4	0.985915493	0.152173913	1
		0	0.545454545	0.6	0.6	0.6	0.014084507	0.847826087	0
		0.292786421	0.252668124	0.307213579	0.307213579	0.307213579	0.278701914	0.555039665	0.292786421
		0.381953028	0.163501517	0.218046972	0.218046972	0.218046972	0.367868521	0.465873059	0.381953028
		0.422632794	0.122821751	0.177367206	0.177367206	0.177367206	0.408548287	0.425193292	0.422632794
		0.5	0.045454545	0.1	0.1	0.1	0.485915493	0.347826087	0.5
min(abs(pj-pi))	1.414730279	0	0.090909091	0.2	0.2	0.2	0.028169014	0.695652174	0
Pobj	17.68412849								
Rhytm	46.6								
aul	11.76	1.96	0	4.9	4.9	0	0	0	0
aur	13.7	0	8.8	0	0	4.9	0	0	0
all	49.7	0	0	0	0	0	49.7	0	0
alr	14.84	0	0	0	0	0	0	12.88	1.96
aul'	0.236619718								
aur'	0.275653924								
all'	1								
alr'	0.298591549								

Detail of old interface:



Detail of new interface:

τ = 0
voorsp.
2
4 sec.



Appendix D: Calculation of Ngo-like complexity measure

Balance	Equilibrium	Symmetry	Sequence	Simplicity	Density	Regularity	Economy
86	1	55	0	82	12	28	0
Homogenity	**Unity**	**Cohesion**	**Proportion**	**Rhytm**			**complexity**
33	50	22	14	56			37

number of objects	n	8
number of vertical alignment points	nvap	3
number of horizontal alignment points	nhap	6
number of distinct distances between column and row starting points	nspacing	1
area layout	alayout	190.43
width layout	blayout	13.9
height layout	hlayout	13.7
width of frame	bframe	13.9
height of frame	hframe	13.7
	aframe	190.43
	nsize	1

Martijn , NEW

complexity: 36.5

Balance	Equilibrium	Symmetry	Sequence	Simplicity	Density	Regularity
86	1	55	0	82	12	28
Homogenity	**Unity**	**Cohesion**	**Proportion**	**Rhytm**		
33	50	22	14	56		

		1	2	3	4	5	6	7	8
object number	i	1	2	3	4	5	6	7	8
quadrant		UL	UL	UL	UR	UR	LL	LR	LR
width	bi, bij	0.8	2.2	2.2	2.2	2.2	8.4	0.8	0.8
height	hi, hij	0.8	0.8	0.8	0.8	0.8	8.2	8.4	0.8
area	ai	0.64	1.76	1.76	1.76	1.76	68.88	6.72	0.64
x-coordinate	xi	1	5.6	5.6	8.2	8.2	6.8	11.9	13
y-coordinate	yi	12.7	12.7	11.6	12.7	11.6	6.8	6.8	6.8
x-coordinate fr	xc	6.95	6.95	6.95	6.95	6.95	6.95	6.95	6.95
y-coordinate fr	yc	6.85	6.85	6.85	6.85	6.85	6.85	6.85	6.85

Balance	86.3								
amax	68.88								
wt	0.760511034	0.055284553	0.143089431	0.143089431	0.20952381	0.20952381	0	0	0
wb	8.081765389	0	0	0	0	0	6.8	1.16097561	0.120789779
wl	7.475261324	0.054355401	0.324506388	0.296399535	0	0	6.8	0	0
wr	1.347502904	0	0	0	0.324506388	0.296399535	0	0.663414634	0.063182346
Bhor	0.905897908								
Bvert	0.819738355								

Equilibrium	0.6								
totaal ai	83.92								
Ex	0.004853029	-3.808	-2.376	-2.376	2.2	2.2	-10.332	33.264	3.872
Ey	0.008098606	3.744	10.296	8.36	10.296	8.36	-3.444	-0.336	-0.032

Symmetry	54.7								
Xul	8.65	5.95	1.35	1.35	0	0	0	0	0
Xur	2.5	0	0	0	1.25	1.25	0	0	0
Xll	0.15	0	0	0	0	0	0.15	0	0
Xlr	11	0	0	0	0	0	0	4.95	6.05
Yul	16.45	5.85	5.85	4.75	0	0	0	0	0
Yur	10.6	0	0	0	5.85	4.75	0	0	0
Yll	0.05	0	0	0	0	0	0.05	0	0
Ylr	0.1	0	0	0	0	0	0	0.05	0.05
Hul	2.4	0.8	0.8	0.8	0	0	0	0	0
Hur	1.6	0	0	0	0.8	0.8	0	0	0
Hll	8.2	0	0	0	0	0	8.2	0	0
Hlr	9.2	0	0	0	0	0	0	8.4	0.8
Bul	5.2	0.8	2.2	2.2	0	0	0	0	0
Bur	4.4	0	0	0	2.2	2.2	0	0	0
Bll	8.4	0	0	0	0	0	8.4	0	0
Blr	1.6	0	0	0	0	0	0	0.8	0.8
fi ul	8.835045129	0.983193277	4.333333333	3.518518519	0	0	0	0	0
fi ur	8.48	0	0	0	4.68	3.8	0	0	0
fi ll	0.333333333	0	0	0	0	0	0.333333333	0	0
fi lr	0.018365473	0	0	0	0	0	0	0.01010101	0.008264463
Rul	19.28602551	8.344159634	6.003748829	4.93811705	0	0	0	0	0
Rur	10.89377718	0	0	0	5.982056503	4.911720676	0	0	0
Rll	0.158113883	0	0	0	0	0	0.158113883	0	0
Rlr	11.00045913	0	0	0	0	0	0	4.950252519	6.050206608

Appendix D: Calculation of Ngo-like complexity measure

normalised:									
Xul'	0.786363636	Hul'	0.260869565	fi ul'	1				
Xur'	0.227272727	Hur'	0.173913043	fi ur'	0.959813999				
Xll'	0.013636364	Hll'	0.891304348	fi ll'	0.037728538				
Xlr'	1	Hlr'	1	fi lr'	0.002078707				
Yul'	1	Bul'	0.619047619	Rul'	1				
Yur'	0.6443769	Bur'	0.523809524	Rur'	0.564853405				
Yll'	0.003039514	Bll'	1	Rll'	0.008198365				
Ylr'	0.006079027	Blr'	0.19047619	Rlr'	0.570384972				
SYMvert	0.339808356	SYMhor	0.689071693	SYMrad	0.610840963				
Sequence									
Simplicity	82.4								
Density	11.9	0.440686867							
Regularity	28.1								
Ral	56.25								
Rsp	0								
Economy	0.0								
Homogenity	33.3								
W	1680								
Wmax	2520								
nul	3	1	1	1	0	0	0	0	0
nur	2	0	0	0	1	1	0	0	0
nll	1	0	0	0	0	0	1	0	0
nlr	2	0	0	0	0	0	0	1	1
Unity	50.0								
Uform	0								
Uspace	100								
Cohesion	21.7								
tfl	1								
Cfl	1								
ti	15.14870288	1.01459854	0.368944924	0.368944924	0.368944924	0.368944924	0.990441432	10.65328467	1.01459854
fi	4.531311911	0.985611511	0.368944924	0.368944924	0.368944924	0.368944924	0.990441432	0.093867763	0.985611511
Clo	0.566413989								
Proportion	13.6	pj	1	0.707213579	0.618046972	0.577367206	0.5		
playout	0.985611511		0.014388489	0.278397932	0.367564539	0.408244305	0.485611511		
Mlayout	2.877697842								
pi		1	0.363636364	0.363636364	0.363636364	0.363636364	0.976190476	0.095238095	1
		0	0.636363636	0.636363636	0.636363636	0.636363636	0.023809524	0.904761905	0
		0.292786421	0.343577215	0.343577215	0.343577215	0.343577215	0.268976898	0.611975483	0.292786421
		0.381953028	0.254410608	0.254410608	0.254410608	0.254410608	0.358143505	0.522808876	0.381953028
		0.422632794	0.213730842	0.213730842	0.213730842	0.213730842	0.398823271	0.48212911	0.422632794
		0.5	0.136363636	0.136363636	0.136363636	0.136363636	0.476190476	0.404761905	0.5
min(abs(pj-pi))	1.948051948	0	0.272727273	0.272727273	0.272727273	0.272727273	0.047619048	0.80952381	0
Pobj	24.35064935								
Rhytm	55.8								
aul	4.16	0.64	1.76	1.76	0	0	0	0	0
aur	3.52	0	0	0	1.76	1.76	0	0	0
all	68.88	0	0	0	0	0	68.88	0	0
alr	7.36	0	0	0	0	0	0	6.72	0.64
aul'	0.06039489								
aur'	0.051103368								
all'	1								
alr'	0.106852497								

Appendix E: Question List used during experiments

Proefpersoon:

Proefpersoon nummer: datum:

1. Klaarleggen: 9 taakbelastingsformulieren, uitleg, declaratieformulier
2. Uitleg geven / laten lezen
3. Eventueel toelichten (*na ieder experiment RSME invullen* ; *er is gelegenheid tot pauze*)
4. Oefenen
5. Experiment uitvoeren + invullen taakbelastingsformulieren
6. Vragenlijstje, nabespreken
7. Bedanken/Betalen

Vragenlijstje:

Hoe vind/vond je de verschillende Interfaces ?

lastig	handig
complex	eenvoudig
mooi	lelijk
symmetrisch	asymmetrisch
druk	rustig
overzichtelijk	chaotisch

Instructions

You have 12 cells

AIM:

1. Control the cells with a **white graph** (the others are automatic) *Note that it is possible to change the input of an automatically controlled cell. The slider input is added to the controller input. The controller will always change its output to minimize the error (SP-output).*
2. You must make your output (blue line) as close as possible to the expected output, 'SP' (orange line)

HINTS:

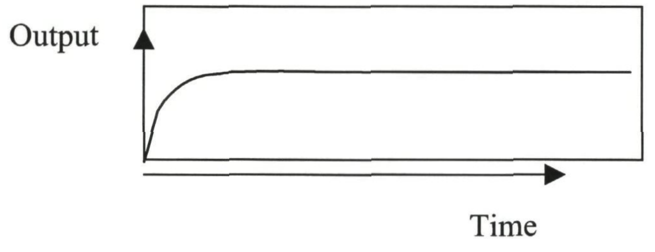
- The SP will be 0 initially, and at a random time change to any value between 0 and 2.

Altering the Input and Output:

- You can alter the input value using the slider
- -The input goes from the slider into a simulink function
  -The simulink function calculates the output
  -The output is graphed as a blue line

Therefore the input from the slider effects the output **but not directly**

- A constant slider input would give you an output like this:



Notice that the output increases rapidly at first and then remains constant

- The output from a cell is fed into all following cells, with a gain of 0.5*

e.g. 1) the input to cell 4 = the input from you **+** 1/2 the output from cells 1,2 and 3.
e.g. 2) What you do in cell 7 will effect cells 8 to 12

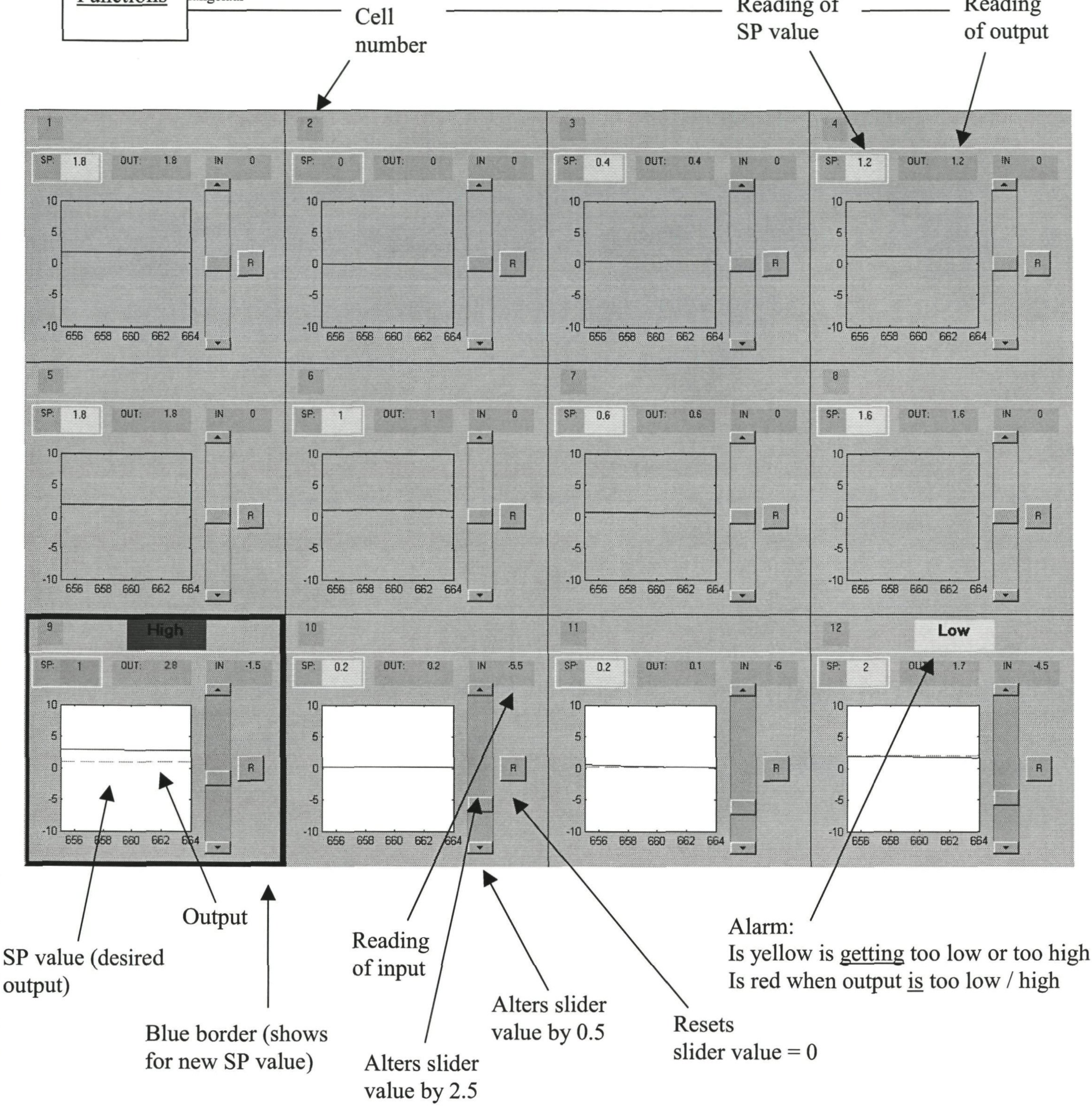
* this is alterable by the tester

SCORE:

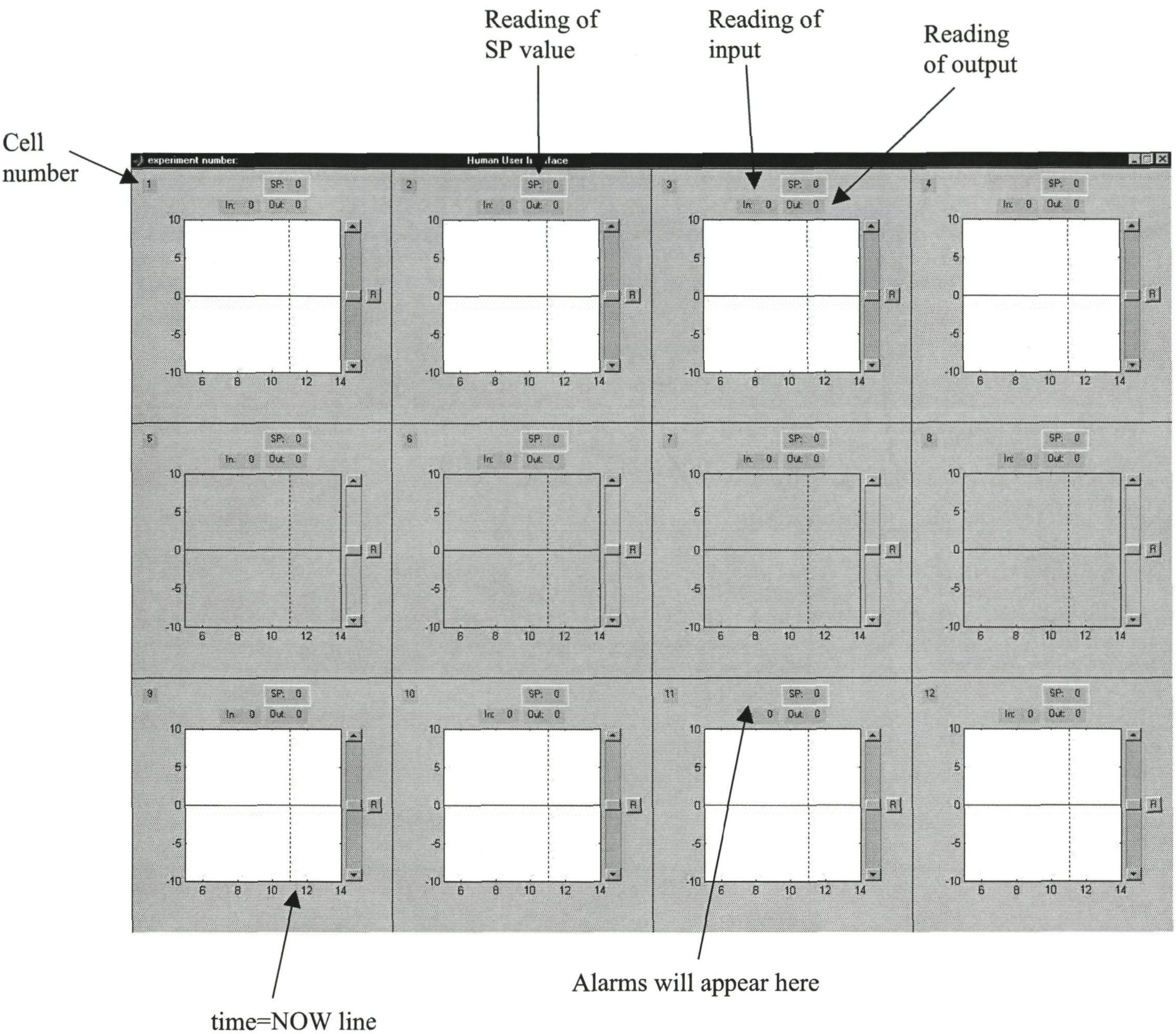
The highest score you can get is 100.
For a **better score** keep the blue line (your output) as close as you can to the orange line (SP). A negative score means your mean error was greater than one.

Appendix F: Subject Instructions

Cell Functions

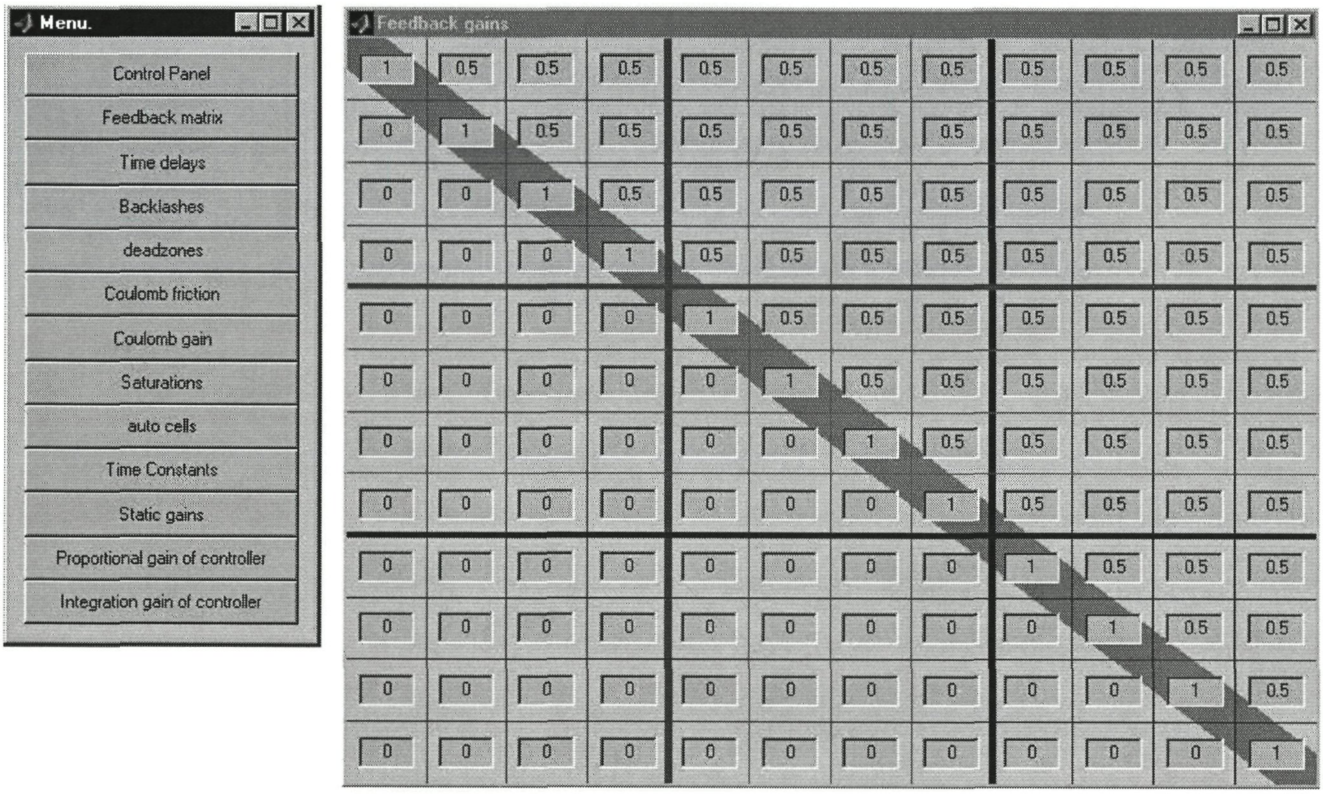


Trend information display



The layout of the trend-information-display is a little different. The left part of the graph displays history, and the right part of the graph displays a prediction for the future.

Depending on the purpose of the test, the tester can alter the appearance, but he also can alter the system. There are a lot of parameters possible to change the behavior of the simulink system. An example is shown below.



For this experiment, no changes are made in the system behaviour.

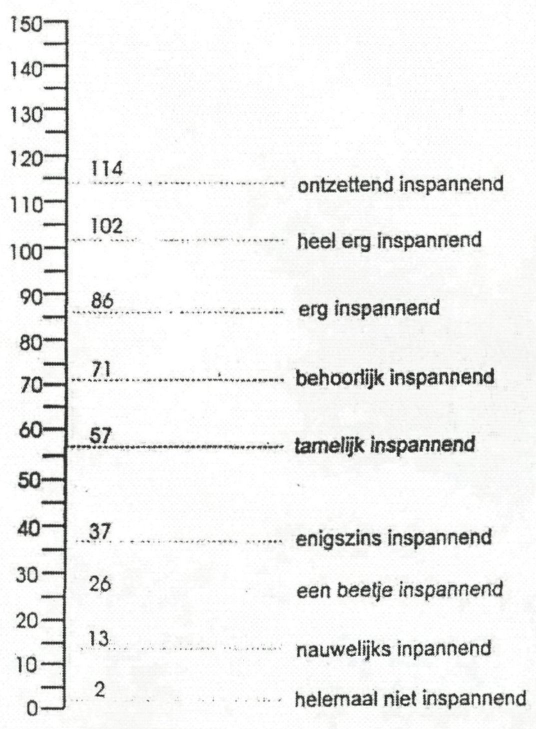
Appendix 2

RSME Rating Form Used in Chapter 5

Rating Scale Mental Effort (RSME)

Instructions: Choose a point on the scale that represents the magnitude of the mental effort in the task you just performed and write the number related to the point chosen to **Table 1** (right).

The RSME measures an operator's mental effort invested when he/she performs a task. Mental effort may be defined as the total amount of controlled cognitive process in which an operator is engaged. Simply to say, mental effort is related to attention. The more attention a task demands, the more effort will be required. In the experiment, you will assess how much effort you have invested during task performance, i.e. how much cognitive costs and how hard you have to work to perform one or all tasks.



Rating Scale Mental Effort (Zijlstra, 1993)

Table 1. Workload Assessment Record

Subject number:_________

Session number:_________

Task (Cell) Number	RSME
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
Overall mental load in operating the whole system	

Note: For automated cells, you do not need to assess mental load.

Testers manual

Contents.

Before beginning

To understand the experiments and the idea behind the program, read a few pages of Zhi-Gang Wei's thesis: "Mental Load and Performance at different Automation Levels" Pages 57 to 67 are in Appendix1.

To understand the program, look at appendix 2. Hre you will find a flowchart and a brief description of the programs. There are 13 Matlab programs and 1 simulink program. Please read these before going further.

To run the program, you need to download the Humusoft real time toolbox (free demo version available on the internet).

Beginning

To run the program, you need to download the Humusoft real time toolbox (free demo version available on the internet).

Matlab Commands

There are three Matlab commands that can be typed at the command prompt:

1. Matrix
2. Starten
3. Results

Matrix

What it does

Matrix creates a 12x12 matrix. The numbers in the display determine the gain for feedback! feed-forward between cells. E.g. cell 3-across-2-down gives the gain between the output of cell 2 and the input of cell 3.

How to use it

Type Matrix at the command prompt
To change a number, enter the desired value in the right box. Keep this window open and return to the Matlab command prompt

Default

The Default is 0 in the lower left triangle and 0.5 in the upper right triangle

Note:

The values on the diagonal of this matrix (-1) are not currently used. It would be possible to link these numbers to kii (the cell's internal feedback) but this also increases the risk of an error occuring

Starten

How to use it

Type Starten at the command prompt This begins the simulation

Variables

There is a full list of variables included later. Here are some of the most important variables that will need to be changed, and where they can be found.

List of variables that can be changed:

Stop Time

Length of time of simulation
In SimProg, simulation, parameters. Default =900

Sp time

Interval between sp calls .In HUlgraph
Find sp_time=rand(1)*x (line 272)
Change 'x' to set the average number of seconds between sp_times .Default = 40

Auto_cells

Number and position of automatically controlled cells

In starten
Approx line 43
Change the number after test_no to choose one of the pre-defined set-ups
Or choose "test_no == 0" to define your own choice of automatically controlled cells.

Results

How to use it

Type Results at the command prompt after the 25 tests have been done. This gives three graphs of the mean error, SE and SPF for the 25 tests.

Note:

It will only give a correct graph if all tests (1-25) have been completed

Running an experiment

1. Open Matlab 6.1
2. Create folder in s:\horst\HCoCS\testdata\people\ with the subjects name
3. Let subject read the instructions
4. Change to the directory where the programs are (e.g. s:\horst\HCoCS\matsim)
5. Subject has 1 hour to learn to use the system
6. Check list:
   - Time is right in the simulink parameters
   - Subject has mental load report sheet (and enough for all the tests)
   - Feedback matrix, and all other matrices, have the right values
   - test_no is the right number
7. Run simulations
   - Run the simulations designed in Starten. Before each simulation, increase the value of test_no by one. Each subject has 28 sessions of 15 minutes each.

8. After each Simulation, save the data by entering the folder name (Name of the subject) and the file name (Test number) e.g. s:\horst\HCoCS\testdata\people\Arjen\1
9. After all simulations are completed Make sure you are in the subject's directory e.g. s:\horst\HCoCS\testdata\people\Arjen Type 'Results' at the command prompt

Apendix I

PP 57 -67 of Wei's thesis

Apendix II

The programs in brief

a) A brief decription
b) A flow chart

A brief description

MATRIX
This program creates a GUI (Graphical User Interface) of 12 x 12 cells to fill in numbers. These numbers are read by MatrixRead. The values are used in the HUI programmes for the feedback/feed-forward in simulink.

MATRIX MAKE
This programme creates a matrix of 12 x 12 of initial values to put in the matrix. These values determine which simulink cells feed forward/back and with what constant they do this.

STARTEN
Works in base workspace so simulink can get the variable values. Starts simulation. Sets which cells are automatic cells. Called by the user typing "Starten" at the command prompt.

MATRIX READ
This programme reads the values typed into the GUI (Graphic User Interface) and makes a matrix of these values. The matrix is used in the HUI programmes. The values of the matrix determine which simulink cells feedback to which, and with what constant.

sHUI
This is the first s-function in full_function4mask3. It has the standard s-function setup and calls HUlgraph.

HUIGraph
This is called by the first s-function in full_function3mask4. It creates the general layout for the GUI (Graphical User Interface) and starts the clock.
It initialises the slider to react to user input and gives random SP values to the interface

RANDOM SP
Creates random SP value for cells and frames them

SliderVal
This function creates the series of sliders in the visual display. The information from the slider is displayed in the "in" box. It feeds the information into HUlgraph, where it is repetitively called back.

sGraph
This is the second s-function in full_function4mask3. It has the standard s-function setup and calls GraphVal.

GraphVal
This function creates the series of graphs in the visual display. It updates them with the new information once a second. It is called by the second s-function in full_function3mask4.

Cell_Error
Calculates the error between the expected output (SP) and actual output (simout). Automatically controls cells specified in Starten. Called by sGraph.

SCORE
After the simulation has run, shows user how effective they were and allows them to save the data in a specified file. Called automatically at the termination of the simulation.

SAVE DATA
Called by the SaveTo pushbutton callback (Score). Saves sum of mean error, SE and SPF to designated file.

RESULTS
Retrieves data from .mat files and plots it.
Assumes you are in the current subjects subdirectory Assumes you have named the 25 files as 1, 2, 3, etc.
Run after all the tests are done through Starten (otherwise It will give you incorrect information).
Run by typing Results at the command prompt.

SimProg
Simulink programme which calls the Matlab functions and operates on their values. Consists of two s functions and a subsystem. Is called by Starten

System
Subsystem in SimProg
Is made up of cells such as drawn on page 59 of Wei's thesis, and many nonlinear blocks. Their values are defined in matrix and matrixmake.

Apendix III

Variables

i) a brief decription

ii) a flow chart

List* of *Variables and programs they are used in.

Alphabetically listed by name

[Italics] indicates it is a local variable

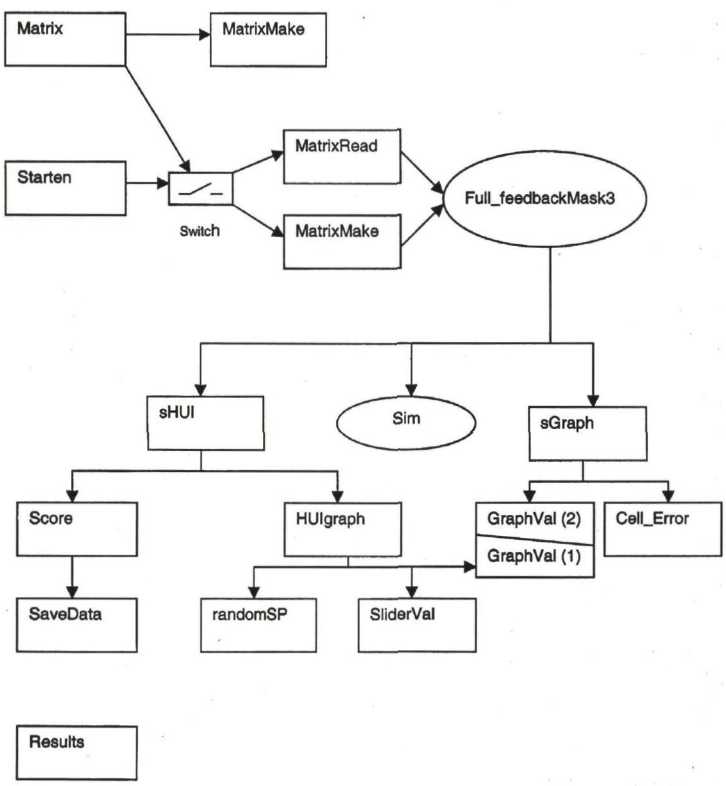
the list of variables is omitted here

Variables from Wei

and thier names in the programs

K -connection between cells inputs an outputs = feedb Kii -internal feedback gain = kii N -number of cells: defined as fixed number throughout programs
t -time constant = t_sim
u -input matrix into simulation = input_sim k -kth sampling period, used in Cell_Error (calculated from time, in secs) = t_g_old e -error between output and sp, used in Cell_Error = c_error

Program Interconnections



```
%                   HUIGraph
% This is called by the first s-function in
% SimProg. It creates the general layout for the
% GUI (Graphical User Interface) and the time bases which
% the simulation works in. It initialises the slider to react
% to user input and gives random SP values to the interface
%
% Sally Gardner 22-02-01
%
% Changed by Martin Langelaar in order to create
% several User Interfaces (april 2001)

function [input_sim]=HUIgraph(varargin)
% input_sim is passed out to sHUI. Number of input arguments
% (varargin) dictates which part of HUIgraph runs

% Variables used in run_sim
global kii t_sim
% Variables used in GraphVal
global Alarm frame_toc Grf
% Variables used in SliderVal and GraphVal
global HUI tictoc CellNumber
% Variables used in SliderVal and run_sim
global input_sim simout
% Variables used in randomSP
global Frame Call Schuiver sp sp_count
% Variables used in here
global ftoc old_ftoc sp_time test_no
% Variables from Starten
global system
% Variable for visible
global CellNo
% Variable for Cell_Error
global auto_control_Ts in_put

global spmat tsimulation seemouse
global GUinterface

switch nargin,
case 1,

          % Start clock running
  tic;

  % Initialise variables
  sp_time = spmat(1,1);
  input_sim=zeros(1,12);
  sp=zeros(1,12);
  ftoc=1;
  old_ftoc=ftoc-1;
  sp_count=1;
  auto_control_Ts=1;
  seemouse=[];


   %%%%%%%%%%%%%%%%%%%%%%%%%%%
          % Setting up the display %
          %%%%%%%%%%%%%%%%%%%%%%%%%%%



          % Creates the figure base and position
   % the GUI cannot be resized, and always is fullscreen.

   scrsize = get(0,'screensize');
   scrposition = scrsize - [0 0 0 19];

   HUI.Fig = figure;
                    set(HUI.Fig,'Parent',0,...
                              'MenuBar','none',...
        'Position',scrposition,...
                         'Units','normalized',...
        'resize','off',...
```

```
        'Name',['experiment number: ' num2str(test_no) '
Human User Interface'],...
        'BackingStore','off',...
        'DoubleBuffer','on',...
        'RendererMode','manual',...
        'Renderer','painters',...
        'NumberTitle','off',...
        'WindowButtonMotionFcn','mouse');
%         'WindowButtonMotionFcn','seemouse = get(HUI.Fig,"CurrentPoint");');
         % Creates place for a graph
           HUI.Ax = axes; axis off;
                       set(HUI.Ax,'Parent',HUI.Fig,...
                                 'Position',[0 0 1 1],...
                          'Units','normalized',...
                                 'XLim',[0 1],'YLim',[0 1]);

  % The base Matlab 0,0 coordinate is at the lower left corner
  % The user expects the first cell to be in the top left corner
  % This codes organises a matrix so that the cells are numbered correctly
           c=1;
           for a=1:3;
                       for b=1:4;
                       CellNumber(a,b)=c;
                         c=c+1;
                       end
           end

           %Create a loop to make all the elements in all the cells
           for i=1:4
             for j=1:3

        switch GUinterface

           case 0
                                 %Create an index number for the current cell
                                 CellIndex = CellNumber(4-j,i);

           %%%%% LINES %%%%%
           %Create 2 horizontal white lines in the cells
           %                     HUI.Line1(CellIndex)=line;
             %                                         set(HUI.Line1(CellIndex),'Parent',HUI.Ax,...
             %                                           'Color','w',...
             %                               'LineWidth',1,...
             %'XData',[0 1],'YData',[i-0.3 i-0.3]/3);
                                 HUI.Line2(CellIndex)=line;
                                        set(HUI.Line2(CellIndex),'Parent',HUI.Ax,...
                                             'Color','w',...
                                 'LineWidth',1,...
                                 'XData',[0 1],'YData',[i-0.15 i-0.15]/3);

                                 %Create horizontal and vertical lines between the cells
                                 HUI.LineHorz(CellIndex)=line;
                                   set(HUI.LineHorz(CellIndex),'Parent',HUI.Ax,...
                                               'Color','k',...
                                 'LineWidth',1,...
                                 'XData',[0 1],'YData',[j j]/3);
                                 HUI.LineVert(CellIndex)=line;
                                   set(HUI.LineVert(CellIndex),'Parent',HUI.Ax,...
                                             'Color','k',...
                                 'LineWidth',1,...
               'XData',[i i]/4,'YData',[0 1]);

            % Create white box around SP value
                                 HUI.LineSPbottom(CellIndex)=line;
                                   set(HUI.LineSPbottom(CellIndex),'Parent',HUI.Ax,...
                                             'Color','w',...
                                 'LineWidth',1,...
                                 'XData',[i-0.97 i-0.70]/4,'YData',[j-0.29 j-0.29]/3);
                                 HUI.LineSPleft(CellIndex)=line;
                                   set(HUI.LineSPleft(CellIndex),'Parent',HUI.Ax,...
                                             'Color','w',...
                                 'LineWidth',1,...
                                 'XData',[i-0.97 i-0.97]/4,'YData',[j-0.29 j-0.17]/3);
                                 HUI.LineSPtop(CellIndex)=line;
```

```
    set(HUI.LineSPtop(CellIndex),'Parent',HUI.Ax,...
            'Color','w',...
'LineWidth',1,...
'XData',[i-0.97 i-0.70]/4,'YData',[j-0.17 j-0.17]/3);
HUI.LineSPright(CellIndex)=line;
    set(HUI.LineSPright(CellIndex),'Parent',HUI.Ax,...
            'Color','w',...
'LineWidth',1,...
'XData',[i-0.70 i-0.70]/4,'YData',[j-0.29 j-0.17]/3);


%%%%% TEXT BOXES %%%%%
        % Note: the sliders text box is created in sliders


        % Call box shows sp values (gets values from randomSP
        Call.Edit(CellIndex)=uicontrol;
          set(Call.Edit(CellIndex),'Parent',HUI.Fig,...
                  'Units','normalized',...
        'Position',[(i-0.86)/4 (j-0.28)/3 0.15/4 0.1/3],...
    'Style','text',...
        'String',num2str(sp(CellIndex)),...
'FontWeight','normal',...
'FontSize',[12]);
%       [(i-0.925)/4 (j-0.925)/3 0.2/4 0.1/3]

        % Cell number (set and fixed for the entire experiment)
        CellNo.Edit(i,j)=uicontrol;
          set(CellNo.Edit(i,j),'Parent',HUI.Fig,...
                  'Units','normalized',...
        'Position',[(i-0.95)/4 (j-0.125)/3 0.1/4 0.1/3],...
            'Style','text',...
                  'String','C',...
        'String',num2str(CellIndex),...
        'FontSize',[12]);

% Alarm (Made visible in GraphVal
  Alarm.Edit(CellIndex)=uicontrol;
          set(Alarm.Edit(CellIndex),'Parent',HUI.Fig,...
          'Units','normalized',...
        'Position',[(i-0.6)/4 (j-0.14)/3 0.3/4 0.12/3],...
        'Style','text',...
'FontWeight','Bold',...
        'BackgroundColor','red',...
     'string','Alarm',...
'visible','off',...
'FontSize',[14]);



% Text Labels (require no input or output)
        InputLabel.Edit(i,j)=uicontrol;
          set(InputLabel.Edit(i,j),'Parent',HUI.Fig,...
                  'Units','normalized',...
        'Position',[(i-0.3)/4 (j-0.28)/3 0.1/4 0.1/3],...
            'Style','text',...
                  'string','IN',...
 'FontUnits','normalized');
%[(i-0.18)/4 (j-0.33)/3 0.1/4 0.1/3]
%[(i-0.97)/4 (j-0.26)/3 0.1/4 0.06/3]
        OutputLabel.Edit(i,j)=uicontrol;
          set(OutputLabel.Edit(i,j),'Parent',HUI.Fig,...
                    'Units','normalized',...
                  'Position',[(i-0.66)/4 (j-0.28)/3 0.15/4 0.1/3],...
        'Style','text',...
          'string','OUT:',...
'FontUnits','normalized');
%[(i-0.66)/4 (j-0.33)/3 0.15/4 0.1/3]
%[(i-0.925)/4 (j-0.49)/3 0.15/4 0.06/3]
        CallLabel.Edit(i,j)=uicontrol;
        set(CallLabel.Edit(i,j),'Parent',HUI.Fig,...
                  'Units','normalized',...
        'Position',[(i-0.96)/4 (j-0.28)/3 0.1/4 0.1/3],...
            'Style','text',...
```

```
                                'string','SP:',...
          'FontUnits','normalized');
        %[(i-0.925)/4 (j-0.815)/3 0.1/4 0.06/3]

      case {1,2,3,4,5,6} %GUinterface

                    %Create an index number for the current cell
                    CellIndex = CellNumber(4-j,i);

      %%%%% LINES %%%%%
      %Create 2 horizontal white lines in the cells
      %             HUI.Line1(CellIndex)=line;
        %                           set(HUI.Line1(CellIndex),'Parent',HUI.Ax,...
        %                             'Color','w',...
        %                   'LineWidth',1,...
        %'XData',[0 1],'YData',[i-0.3 i-0.3]/3);
%                           HUI.Line2(CellIndex)=line;
%                      set(HUI.Line2(CellIndex),'Parent',HUI.Ax,...
 %                          'Color','w',...
 %                  'LineWidth',1,...
 %                  'XData',[0 1]/4,'YData',[i-0.15 i-0.15]/3);

                    %Create horizontal and vertical lines between the cells
                    HUI.LineHorz(CellIndex)=line;
                      set(HUI.LineHorz(CellIndex),'Parent',HUI.Ax,...
                              'Color','k',...
                    'LineWidth',1,...
                    'XData',[i-1 i]/4,'YData',[j j]/3);
                    HUI.LineVert(CellIndex)=line;
                      set(HUI.LineVert(CellIndex),'Parent',HUI.Ax,...
                            'Color','k',...
                    'LineWidth',1,...
        'XData',[i-1 i-1]/4,'YData',[j-1 j]/3);


            %%%%% TEXT BOXES %%%%%
                    % Note: the sliders text box is created in sliders


                    % Call box shows sp values (gets values from randomSP)
                    Call.Edit(CellIndex)=uicontrol;
                      set(Call.Edit(CellIndex),'Parent',HUI.Fig,...
                              'Units','normalized',...
                    'Position',[(i-1+0.6)/4 (j-1+0.9)/3 0.07/4 0.06/3],...
              'Style','text',...
                    'String',num2str(sp(CellIndex)),...
          'FontWeight','normal',...
        'FontSize',[9]);
   %    [(i-0.925)/4 (j-0.925)/3 0.2/4 0.1/3]

                    % Cell number (set and fixed for the entire experiment)
                    CellNo.Edit(i,j)=uicontrol;
                      set(CellNo.Edit(i,j),'Parent',HUI.Fig,...
                              'Units','normalized',...
                    'Position',[(i-1+0.04)/4 (j-1+0.9)/3 0.06/4 0.06/3],...
                      'Style','text',...
                              'String','C',...
                    'String',num2str(CellIndex),...
                    'FontSize',[9]);

            % Alarm (Made visible in GraphVal
              Alarm.Edit(CellIndex)=uicontrol;
                      set(Alarm.Edit(CellIndex),'Parent',HUI.Fig,...
                      'Units','normalized',...
                    'Position',[(i-1+0.33)/4 (j-1+0.9)/3 0.16/4 0.06/3],...
                    'Style','text',...
        'FontWeight','Bold',...
                    'BackgroundColor','red',...
                'string','Alarm',...
        'visible','off',...
        'FontSize',[9]);
```

```
                % Text Labels (require no input or output)
                        InputLabel.Edit(i,j)=uicontrol;
                          set(InputLabel.Edit(i,j),'Parent',HUI.Fig,...
                                 'Units','normalized',...
                        'Position',[(i-1+0.33)/4 (j-1+0.82)/3 0.09/4 0.06/3],...
                             'Style','text',...
                                 'string','In:',...
    'FontUnits','normalized');
  %[(i-0.18)/4 (j-0.33)/3 0.1/4 0.1/3]
  %[(i-0.97)/4 (j-0.26)/3 0.1/4 0.06/3]
                        OutputLabel.Edit(i,j)=uicontrol;
                          set(OutputLabel.Edit(i,j),'Parent',HUI.Fig,...
                                   'Units','normalized',...
                                 'Position',[(i-1+0.51)/4 (j-1+0.82)/3 0.09/4 0.06/3],...
                        'Style','text',...
                          'string','Out:',...
   'FontUnits','normalized');
  %[(i-0.66)/4 (j-0.33)/3 0.15/4 0.1/3]
  %[(i-0.925)/4 (j-0.49)/3 0.15/4 0.06/3]
                        CallLabel.Edit(i,j)=uicontrol;
                        set(CallLabel.Edit(i,j),'Parent',HUI.Fig,...
                                 'Units','normalized',...
                        'Position',[(i-1+0.51)/4 (j-1+0.9)/3 0.09/4 0.06/3],...
                             'Style','text',...
                                 'string','SP:',...
    'FontUnits','normalized');
  %[(i-0.925)/4 (j-0.815)/3 0.1/4 0.06/3]

 % Create white box around SP value
                        HUI.LineSPbottom(CellIndex)=line;
                          set(HUI.LineSPbottom(CellIndex),'Parent',HUI.Ax,...
                                 'Color','w',...
                        'LineWidth',1,...
                        'XData',[i-1+0.50 i-1+0.68]/4,'YData',[j-1+0.97 j-1+0.97]/3);
                        HUI.LineSPleft(CellIndex)=line;
                          set(HUI.LineSPleft(CellIndex),'Parent',HUI.Ax,...
                                 'Color','w',...
                        'LineWidth',1,...
                        'XData',[i-1+0.68 i-1+0.68]/4,'YData',[j-1+0.97 j-1+0.89]/3);
                        HUI.LineSPtop(CellIndex)=line;
                          set(HUI.LineSPtop(CellIndex),'Parent',HUI.Ax,...
                                 'Color','w',...
                        'LineWidth',1,...
                        'XData',[i-1+0.68 i-1+0.50]/4,'YData',[j-1+0.89 j-1+0.89]/3);
                        HUI.LineSPright(CellIndex)=line;
                          set(HUI.LineSPright(CellIndex),'Parent',HUI.Ax,...
                                 'Color','w',...
                        'LineWidth',1,...
                        'XData',[i-1+0.50 i-1+0.50]/4,'YData',[j-1+0.89 j-1+0.97]/3);


 end %switch GUinterface

                end %for
   end %for



   %%%%% FRAMES %%%%%
% Frame around cells with a new SP value.
% Dissappears when slider is clicked on
   for i = 1:4;
     for j = 1:3;

 switch GUinterface

   case 0
                        %Create an index number for the current cell
                        CellIndex = CellNumber(4-j,i);

                Frame(CellIndex).Left=uicontrol;
                                 set(Frame(CellIndex).Left,'Parent',HUI.Fig,...
                                   'Units','normalized',...
                          'Position',[(i-1)/4 (j-1)/3 0.02/4 1/3],...
```

```
                    'Style','frame',...
        'BackgroundColor','b',...
      'visible','off');
        Frame(CellIndex).Top=uicontrol;
                    set(Frame(CellIndex).Top,'Parent',HUI.Fig,...
            'Units','normalized',...
                    'Position',[(i-1)/4 (j-0.02)/3 1/4 0.02/3],...
              'Style','frame',...
                    'BackgroundColor','b',...
        'visible','off');
        Frame(CellIndex).Right=uicontrol;
        set(Frame(CellIndex).Right,'Parent',HUI.Fig,...
                    'Units','normalized',...
                'Position',[(i-0.02)/4 (j-1)/3 0.02/4 1/3],...
          'Style','frame',...
              'BackgroundColor','b',...
      'visible','off');
        Frame(CellIndex).Bottom=uicontrol;
                set(Frame(CellIndex).Bottom,'Parent',HUI.Fig,...
              'Units','normalized',...
          'Position',[(i-1)/4 (j-1)/3 1/4 0.02/3],...
                    'Style','frame',...
        'BackgroundColor','b',...
      'visible','off');
case {1,2,3,4,5,6} %GUinterface
              %Create an index number for the current cell
              CellIndex = CellNumber(4-j,i);

      Frame(CellIndex).Left=uicontrol;
                    set(Frame(CellIndex).Left,'Parent',HUI.Fig,...
                      'Units','normalized',...
                'Position',[(i-1)/4 (j-1)/3 0.02/4 1/3],...
                    'Style','frame',...
        'BackgroundColor','b',...
      'visible','off');
        Frame(CellIndex).Top=uicontrol;
                set(Frame(CellIndex).Top,'Parent',HUI.Fig,...
            'Units','normalized',...
                    'Position',[(i-1)/4 (j-0.02)/3 1/4 0.02/3],...
              'Style','frame',...
                    'BackgroundColor','b',...
        'visible','off');
        Frame(CellIndex).Right=uicontrol;
        set(Frame(CellIndex).Right,'Parent',HUI.Fig,...
                    'Units','normalized',...
                'Position',[(i-0.02)/4 (j-1)/3 0.02/4 1/3],...
          'Style','frame',...
              'BackgroundColor','b',...
      'visible','off');
        Frame(CellIndex).Bottom=uicontrol;
                set(Frame(CellIndex).Bottom,'Parent',HUI.Fig,...
              'Units','normalized',...
          'Position',[(i-1)/4 (j-1)/3 1/4 0.02/3],...
                    'Style','frame',...
        'BackgroundColor','b',...
      'visible','off');

end % switch
end %for
end %for

%%%%%%%%%%%%%%%%%%%%%%%%%
% Setting up the slider %
%%%%%%%%%%%%%%%%%%%%%%%%%
SliderVal;

%%%%%%%%%%%%%%%%%%%%%%%%%
% Setting up the graphs %
%%%%%%%%%%%%%%%%%%%%%%%%%
GraphVal(1);


%%%%%%%% AUTOMATIC %%%%%%%%
```

```
  % Define the automatic cells and name them

  for auto = system.auto_cells
     switch GUinterface
        case 0
          set(Schuiver(auto),'BackgroundColor',[0.8 0.8 0.8]);
          set(Grf.Ax(auto),'Color',[0.8 0.8 0.8],'UserData','grey');
        case {1,2,3,4,5,6}
          set(Schuiver(auto),'BackgroundColor',[0.8 0.8 0.8]);
          set(Grf.Ax(auto),'Color',[0.8 0.8 0.8],'UserData','grey');
       end %switch
   end %for


case 2, %executes if the display is already set up
   % Current time rounded to one decimal place
   ftoc=((floor(toc*10))/10);

            % runs randomSP and find where & when new sp value is called
            elapsed_toc=ftoc-old_ftoc;
   if tsimulation >= sp_time
      randomSP;
      random_toc = ftoc;
      frame_toc = ftoc;
      sp_time = spmat(1,sp_count);
      %sp_time = 35 +(rand(1)*2*5);
      %sp_time = 20;
               old_ftoc=ftoc;
            end %if
end %switch
```

```
%                   GraphVal
%
% This function creates the series of graphs in the visual
% display. It updates them with the new information as it
% comes from the simulink programme. It is called by the
% second s-function in SimProg.
%
% Sally Gardner 22-02-01
%
% Altered by Martijn Langelaar april 2002 to create several user interfaces.


function GraphVal(varargin);

% Variables that come from HUIgraph
global Alarm gh1 gh2 Frame Call
% Variables that are used in HUIgraph and SliderVal
global HUI tictoc CellNumber
% Variables that are needed for this function
global InvoerG Y_data n_old Grf sp_graph
% Variables from randomSP
global sp
%Variables for runsim
global simout
% Veriables from sGraph
global t_g_old

global GUinterface

%create a matrix of the last ten seconds
n=floor(toc);
m=n-9;
graph_time=[m:n]';

% This determines if it is the first time the programme has
% been run (in which case it creates the graphs and the visual
% display of the last value on the graph) or if it has been
% run before (in which case it updates the data and then loops
% back on itself)

switch nargin,
case 1,
  % Initialises values
  n_old=floor(toc);
  Y_data=zeros(10,12);
  sp_graph=zeros(10,12);

         %In the loop the visual displays for the twelve cells are created
  for i=1:4
         for j=1:3

     CellIndex=CellNumber(4-j,i);


                 % Makes axes, describes them and plots information from the output of the
                   % simulation programme

       switch GUinterface
         case 0
           Grf.Ax(CellIndex) = axes; axis on;
                                   set(Grf.Ax(CellIndex),'Parent',HUI.Fig,...
                                     'Units','normalized',...
                                     'Position',[(i-0.86)/4 (j-0.86)/3 0.5/4 0.5/3],...
             'XLimMode','manual',...
             'YLimMode','manual',...
             'YLim',[-10 10],...
             'XLim',[-10 0],...
             'Userdata','white',...
             'Box','on',...
             'YaxisLocation','left');

         % Plot data
```

```
gh2(CellIndex)=line('XData',graph_time,'YData',sp_graph(:,CellIndex),'Color',[1 .5 0]);
gh1(CellIndex)=line('XData',graph_time,'YData',Y_data(:,CellIndex),'Color',[0 0 1]);

%output box gets it's values from sim
                InvoerG(CellIndex)=uicontrol;
                  set(InvoerG(CellIndex),'Parent',HUI.Fig,...
                'Units','normalized',...
                    'Position',[(i-0.51)/4 (j-0.28)/3 0.15/4 0.1/3],...
                  'Style','text',...
                'String',num2str(round(0)),...
    'FontSize',[8],...
    'Callback','GraphVal(0)');
  %[(i-0.51)/4 (j-0.33)/3 0.15/4 0.1/3]
  %[(i-0.925)/4 (j-0.6)/3 0.2/4 0.1/3]
case {1,2,3,4,5,6} %GUInterface
  Grf.Ax(CellIndex) = axes; axis on;
                          set(Grf.Ax(CellIndex),'Parent',HUI.Fig,...
                            'Units','normalized',...
                            'Position',[(i-1+0.2)/4 (j-1+0.2)/3 0.6/4 0.6/3],...
    'XLimMode','manual',...
    'YLimMode','manual',...
    'YLim',[-10 10],...
    'XLim',[-10 0],...
    'Userdata','white',...
    'Box','on',...
    'YaxisLocation','left');

% Plot data
gh2(CellIndex)=line('XData',graph_time,'YData',sp_graph(:,CellIndex),'Color',[1 .5 0]);
gh1(CellIndex)=line('XData',graph_time,'YData',Y_data(:,CellIndex),'Color',[0 0 1]);

%output box gets it's values from sim  (Number beheind Out)
                InvoerG(CellIndex)=uicontrol;
                  set(InvoerG(CellIndex),'Parent',HUI.Fig,...
                'Units','normalized',...
                    'Position',[(i-1+0.6)/4 (j-1+0.82)/3 0.07/4 0.06/3],...
                  'Style','text',...
                'String',num2str(round(0)),...
    'FontSize',[8],...
    'Callback','GraphVal(0)');
  %[(i-0.51)/4 (j-0.33)/3 0.15/4 0.1/3]
  %[(i-0.925)/4 (j-0.6)/3 0.2/4 0.1/3]

    % Create horizontal line at t=now to seperate history from prediction
    HUI.TimeLine(CellIndex)=line('Color','k','LineWidth',1,'XData',[(n) (n)],'YData',[-11
10],'LineStyle',':');

      end %switch GUinterface
  end %for
 end %for

case 2,
  % make the time on the graph the same as the time in the simulation
  % try to precict value in several secs (1,2,3,5,10 ?)

  n=t_g_old;
  m=n-9;
  graph_time=[m:n]';


  % Get values of simout (1x12 matrix) from sim
        load simout;

 %%%%%% REPLOT GRAPHS %%%%%
 % replot for each second

 % Values to be plotted are nine recently plotted + 1 latest simout value
 Y_data(1:9,:)=Y_data(2:10,:);
 Y_data(10,:)=simout(1,:);

  Y_data = round(Y_data * 10) / 10;
 %   afronden Arjen !
```

```
% voorsp_vorm=zeros(3,12);
% voorsp_waarden=ones(5,12);
% voorsp_vorm_2=zeros(2,12);
% voorsp_waarden_2=ones(5,12);
%  for loop=1:12
%     if n>8
%         vorm=polyfit(Y_data(6:10, loop)',(m+5):n,2);
%         voorsp_vorm(1:3,loop)=vorm';
%         vorm_2=polyfit(Y_data(8:10, loop)',(m+7):n,1);
%         voorsp_vorm(1:2,loop)=vorm_2';
%
%         voorsp_waarden(1:5,loop)=[polyval(voorsp_vorm(1:3,loop)',n+1) ;
polyval(voorsp_vorm(1:3,loop)',n+2 ) ; polyval(voorsp_vorm(1:3,loop)',n+3) ;
polyval(voorsp_vorm(1:3,loop)',n+5) ;polyval(voorsp_vorm(1:3,loop)',n+10) ];
%
%         voorsp_waarden_2(1:5,loop)=[polyval(voorsp_vorm_2(1:2,loop)',n+1) ;
polyval(voorsp_vorm_2(1:2,loop)',n+2) ; polyval(voorsp_vorm_2(1:2,loop)',n+3) ;
polyval(voorsp_vorm_2(1:2,loop)',n+5) ; polyval(voorsp_vorm_2(1:2,loop)',n+10) ];
%     else
%         voorsp_waarden(1:5,loop)=[Y_data(10,loop) ; Y_data(10,loop) ; Y_data(10,loop); Y_data(10,loop)
;Y_data(10,loop) ];
%         voorsp_waarden_2(1:5,loop)=[Y_data(10,loop) ; Y_data(10,loop) ; Y_data(10,loop);
Y_data(10,loop) ;Y_data(10,loop) ];
%     end %for
% end %if


  sp_graph(1:9,:)=sp_graph(2:10,:);
  sp_graph(10,:)=sp(1,:);

  y_height=10;
  y_height_min = -y_height;

  % For each cell
          for graaf_count=1:12
      switch GUinterface
        case 0
          % Write latest output value in InvoerG
           set(InvoerG(graaf_count),'String',num2str((round((simout(1,graaf_count))*10))/10));

             % set graph axes
          set(Grf.Ax(graaf_count),'XLim',[m n]);

          % Plot new information
           set(gh2(graaf_count),'XData',graph_time,...
                              'YData',sp_graph(1:10,graaf_count),'Color',[1 0.5 0],'LineStyle','--');
           set(gh1(graaf_count),'XData',graph_time,...
                              'YData',Y_data(1:10,graaf_count),'color',[0 0 1]);


          %%%%% ALARM %%%%%
          % Makes alarm show red if output value exceeds error bounds

          is_auto=get(Grf.Ax(graaf_count),'UserData');
          if strcmp(is_auto,'white')

            error_max=sp(graaf_count)+0.4;
            error_min=sp(graaf_count)-0.4;
            error_max_warn=sp(graaf_count)+0.3;
            error_min_warn=sp(graaf_count)-0.3;


            if simout(1,graaf_count)<error_min
                     set(Alarm.Edit(graaf_count),'Visible','on',...
                     'BackgroundColor','r',...
                       'String','Low');
            elseif simout(1,graaf_count)>error_max
                       set(Alarm.Edit(graaf_count),'Visible','on',...
                       'BackgroundColor','r',...
                              'String','High');
                     elseif simout(1,graaf_count)<error_min_warn
                     set(Alarm.Edit(graaf_count),'Visible','on',...
                     'BackgroundColor','y',...
                       'String','Low');
```

```
                elseif simout(1,graaf_count)>error_max_warn
                set(Alarm.Edit(graaf_count),'Visible','on',...
                'BackgroundColor','y',...
                        'String','High');

                else
                        % Turn off alarm if the output is now correct
            set(Alarm.Edit(graaf_count),'Visible','off');
            FrameHide1=get(Frame(graaf_count).Top,'visible');
            if strcmp(FrameHide1,'on')
                        set(Frame(graaf_count).Top,'visible','off');
                                                set(Frame(graaf_count).Bottom,'visible','off');
                                      set(Frame(graaf_count).Left,'visible','off');
                                                set(Frame(graaf_count).Right,'visible','off');
            end
            CallHide=get(Call.Edit(graaf_count),'BackgroundColor');
            if CallHide==[1 0.7 0.1]
              set(Call.Edit(graaf_count),'BackgroundColor',[1 0.9 0.5]);
            end
          end %if simout
        end %if is_auto

      case {1,2,3,4,5,6} % GUinterface
        % Write latest output value in InvoerG
         set(InvoerG(graaf_count),'String',num2str((round((simout(1,graaf_count))*10))/10));
% set graph axes



%       graph_time=(m+3:n+3)';

% dit stond er oorspronkelijk
%          % Plot new information
%                   set(gh2(graaf_count),'XData',graph_time,...
%                           'YData',sp_graph(1:10,graaf_count),'Color',[1 0.5 0],'LineStyle','--');
%         set(gh1(graaf_count),'XData',graph_time,...
%                           'YData',Y_data(1:10,graaf_count),'color',[0 0 1]);

%plot new information zelf maar rekenen



        helling=( ( Y_data(9,graaf_count)-Y_data(8,graaf_count) ) + 2* (Y_data(10,graaf_count)-
Y_data(9,graaf_count)) ) / 3 ;

        switch GUinterface %2
        case 1
          % geen voorspelling
           set(Grf.Ax(graaf_count),'XLim',[m n]);
           set(gh2(graaf_count),'XData',graph_time,'YData',[sp_graph(1:10,graaf_count) ],'Color',[1 0.5
0],'LineStyle','--') ;
             set(gh1(graaf_count),'XData',graph_time,'YData',[Y_data(1:10,graaf_count) ],'color',[0 0 1]);

        case 2
          % 2,5 s voorspelling
           set(Grf.Ax(graaf_count),'XLim',[m+2.5 n+2.5]);
           prediction = Y_data(10,graaf_count) + 2.5 * helling ;
           prediction_time = n+2.5;
           set(gh2(graaf_count),'XData',[graph_time; prediction_time]
,'YData',[sp_graph(1:10,graaf_count) ; sp_graph(10,graaf_count)],'Color',[1 0.5 0],'LineStyle','--') ;
             set(gh1(graaf_count),'XData',[graph_time;
prediction_time],'YData',[Y_data(1:10,graaf_count) ; prediction],'color',[0 0 1]);

        case 3
          % 3s voorspelling
           set(Grf.Ax(graaf_count),'XLim',[m+3 n+3]);
           prediction = Y_data(10,graaf_count) + 3 * helling ;
           prediction_time = n+3;
           set(gh2(graaf_count),'XData',[graph_time ;
prediction_time],'YData',[sp_graph(1:10,graaf_count) ; sp_graph(10,graaf_count)],'Color',[1 0.5
0],'LineStyle','--') ;
             set(gh1(graaf_count),'XData',[graph_time ; prediction_time
],'YData',[Y_data(1:10,graaf_count) ; prediction],'color',[0 0 1]);
         case 4
```

```
            % 5s voorspelling
            set(Grf.Ax(graaf_count),'XLim',[m+5 n+5]);
            prediction = Y_data(10,graaf_count) + 5 * helling ;
            prediction_time = n+5;
            set(gh2(graaf_count),'XData',[graph_time;
prediction_time],'YData',[sp_graph(1:10,graaf_count) ; sp_graph(10,graaf_count)],'Color',[1 0.5
0],'LineStyle','--') ;
              set(gh1(graaf_count),'XData',[graph_time;
prediction_time],'YData',[Y_data(1:10,graaf_count) ; prediction],'color',[0 0 1]);

          case 5
            % 2s voorspelling
            set(Grf.Ax(graaf_count),'XLim',[m+2 n+2]);
            prediction = Y_data(10,graaf_count) + 2 * helling ;
            prediction_time = n+2;
            set(gh2(graaf_count),'XData',[graph_time;
prediction_time],'YData',[sp_graph(1:10,graaf_count) ; sp_graph(10,graaf_count)],'Color',[1 0.5
0],'LineStyle','--') ;
              set(gh1(graaf_count),'XData',[graph_time;
prediction_time],'YData',[Y_data(1:10,graaf_count) ; prediction],'color',[0 0 1]);
          case 6
            % 4s voorspelling
            set(Grf.Ax(graaf_count),'XLim',[m+4 n+4]);
            prediction = Y_data(10,graaf_count) + 4 * helling ;
            prediction_time = n+4;
            set(gh2(graaf_count),'XData',[graph_time;
prediction_time],'YData',[sp_graph(1:10,graaf_count) ; sp_graph(10,graaf_count)],'Color',[1 0.5
0],'LineStyle','--') ;
              set(gh1(graaf_count),'XData',[graph_time;
prediction_time],'YData',[Y_data(1:10,graaf_count) ; prediction],'color',[0 0 1]);


          end %switch GUinterfzce 2



         %%%%% ALARM %%%%%
         % Makes alarm show red if output value exceeds error bounds

         is_auto=get(Grf.Ax(graaf_count),'UserData');
         if strcmp(is_auto,'white')

           error_max=sp(graaf_count)+0.4;
           error_min=sp(graaf_count)-0.4;
           error_max_warn=sp(graaf_count)+0.3;
           error_min_warn=sp(graaf_count)-0.3;


           if simout(1,graaf_count)<error_min
                   set(Alarm.Edit(graaf_count),'Visible','on',...
                   'BackgroundColor','r',...
                     'String','Low');
           elseif simout(1,graaf_count)>error_max
                     set(Alarm.Edit(graaf_count),'Visible','on',...
                     'BackgroundColor','r',...
                             'String','High');
                   elseif simout(1,graaf_count)<error_min_warn
                   set(Alarm.Edit(graaf_count),'Visible','on',...
                   'BackgroundColor','y',...
                     'String','Low');
                     elseif simout(1,graaf_count)>error_max_warn
                     set(Alarm.Edit(graaf_count),'Visible','on',...
                     'BackgroundColor','y',...
                             'String','High');

                     else
                             % Turn off alarm if the output is now correct
             set(Alarm.Edit(graaf_count),'Visible','off');
             FrameHide1=get(Frame(graaf_count).Top,'visible');
             if strcmp(FrameHide1,'on')
                             set(Frame(graaf_count).Top,'visible','off');
                                                   set(Frame(graaf_count).Bottom,'visible','off');
                                         set(Frame(graaf_count).Left,'visible','off');
                                                   set(Frame(graaf_count).Right,'visible','off');
```

```
            end
            CallHide=get(Call.Edit(graaf_count),'BackgroundColor');
            if CallHide==[1 0.7 0.1]
              set(Call.Edit(graaf_count),'BackgroundColor',[1 0.9 0.5]);
            end
          end %if simout
        end %if is_auto
  %      HUI.TimeLine(CellIndex)=line('Color','k','LineWidth',1,'XData',[(n) (n)],'YData',[-11
10],'LineStyle',':');

        set(HUI.TimeLine(graaf_count),'XData',[n n]);
      end%switch GUinterface
         end %for
end %switch
drawnow
```

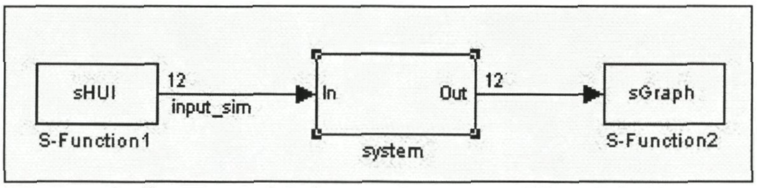


Figure 4: System with in- and output (SimuLink)

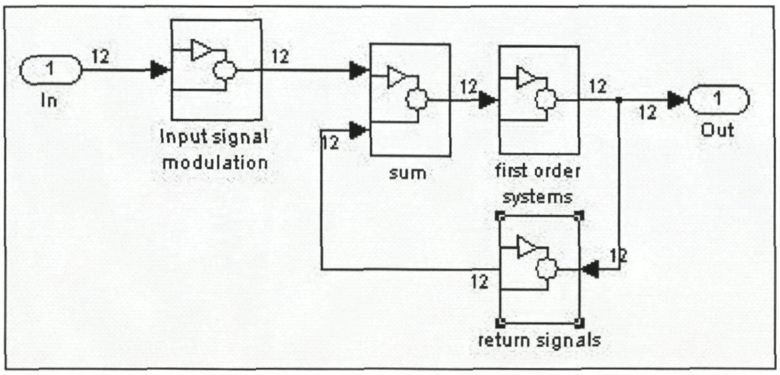


Figure 5: System (SimuLink)

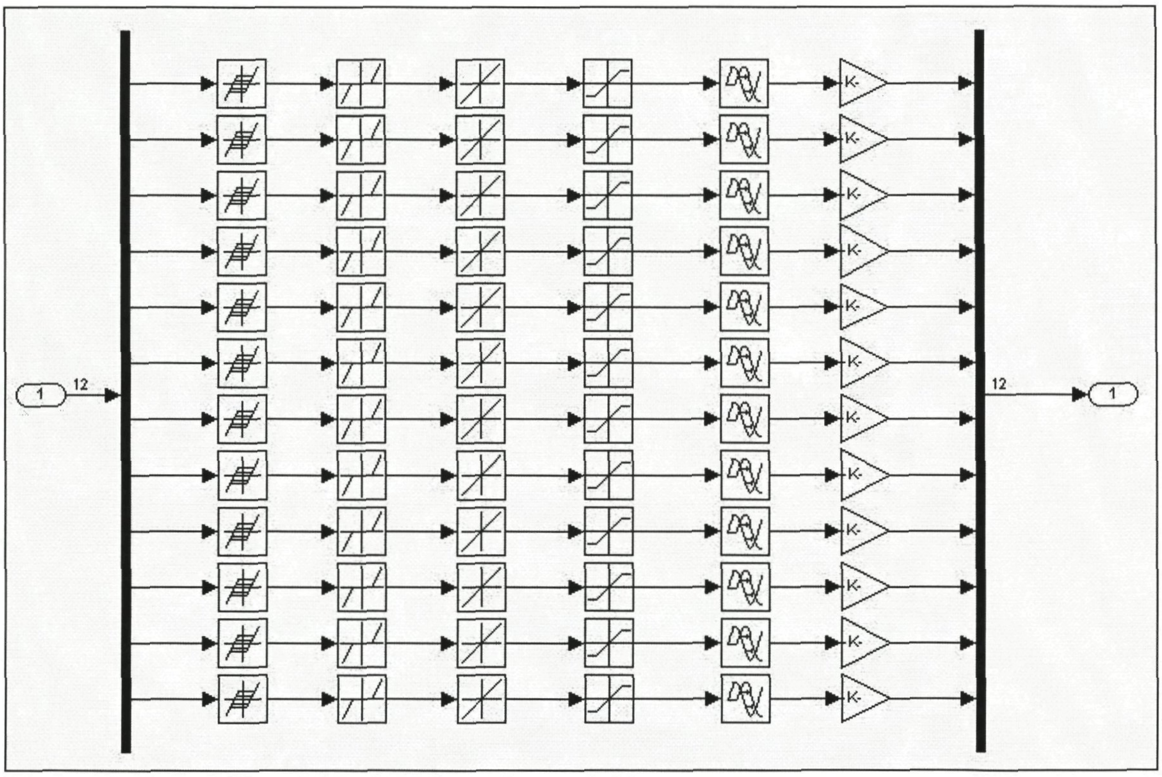


Figure 6: Input Signal Modulation (SimuLink)

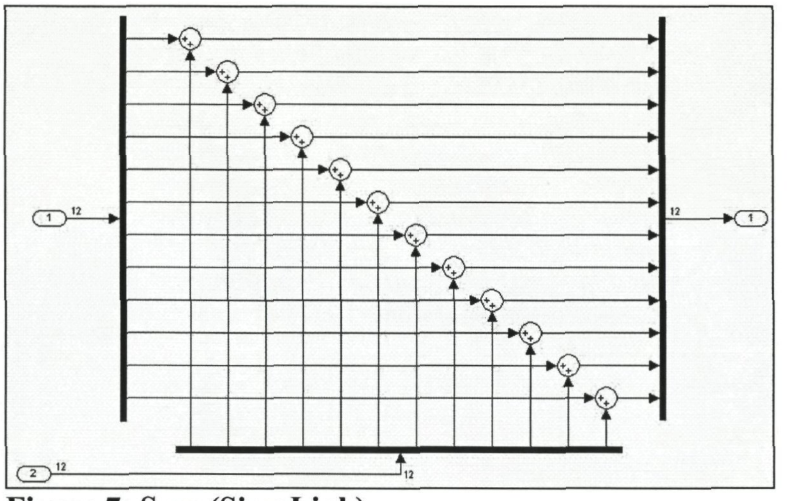


Figure 7: Sum (SimuLink)

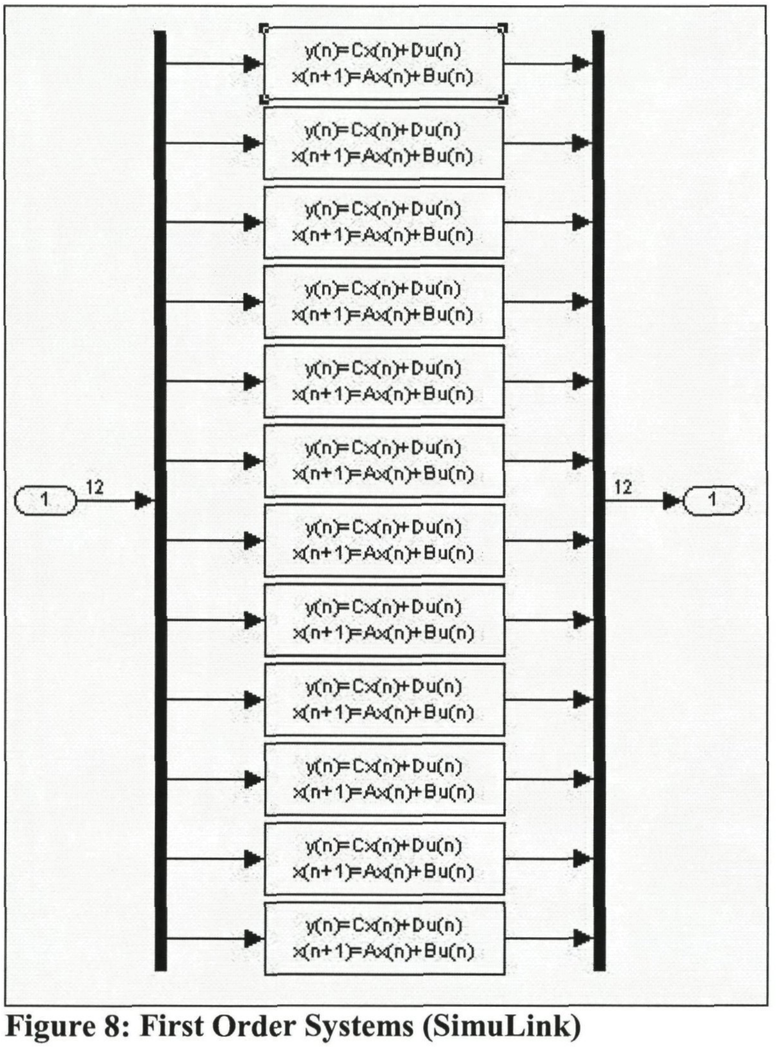


Figure 8: First Order Systems (SimuLink)

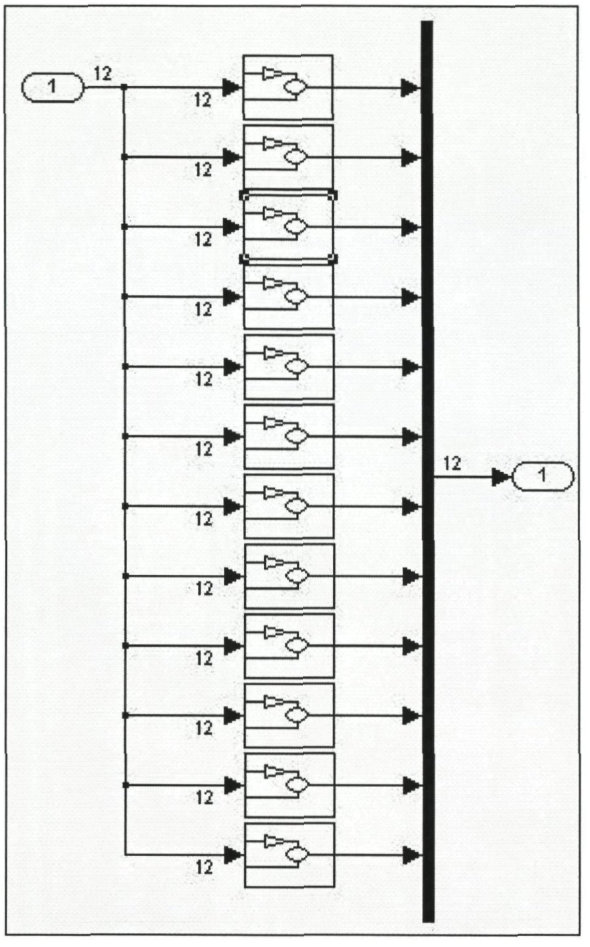


Figure 9: Return Signals (SimuLink)

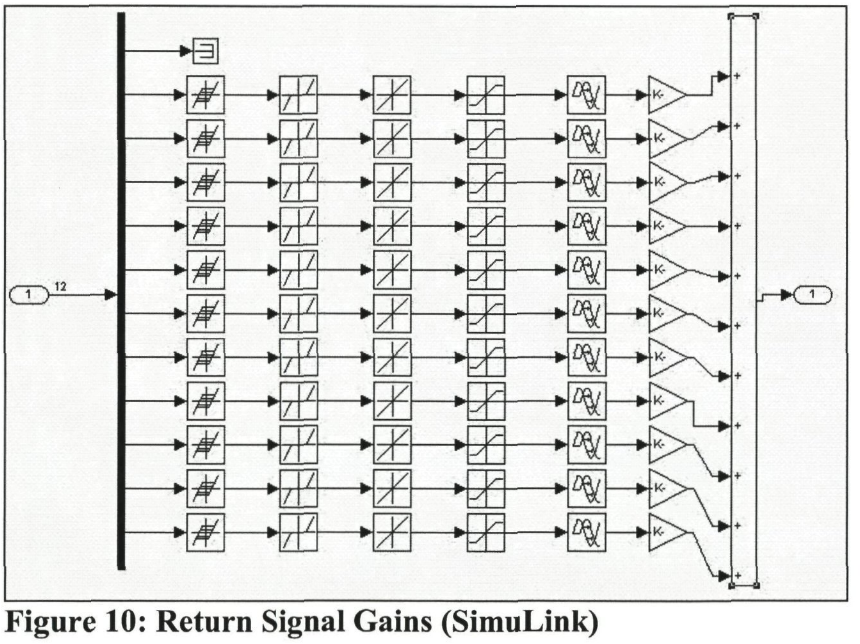


Figure 10: Return Signal Gains (SimuLink)

Appendix J: (MatLab/SimuLink) pictures of the system

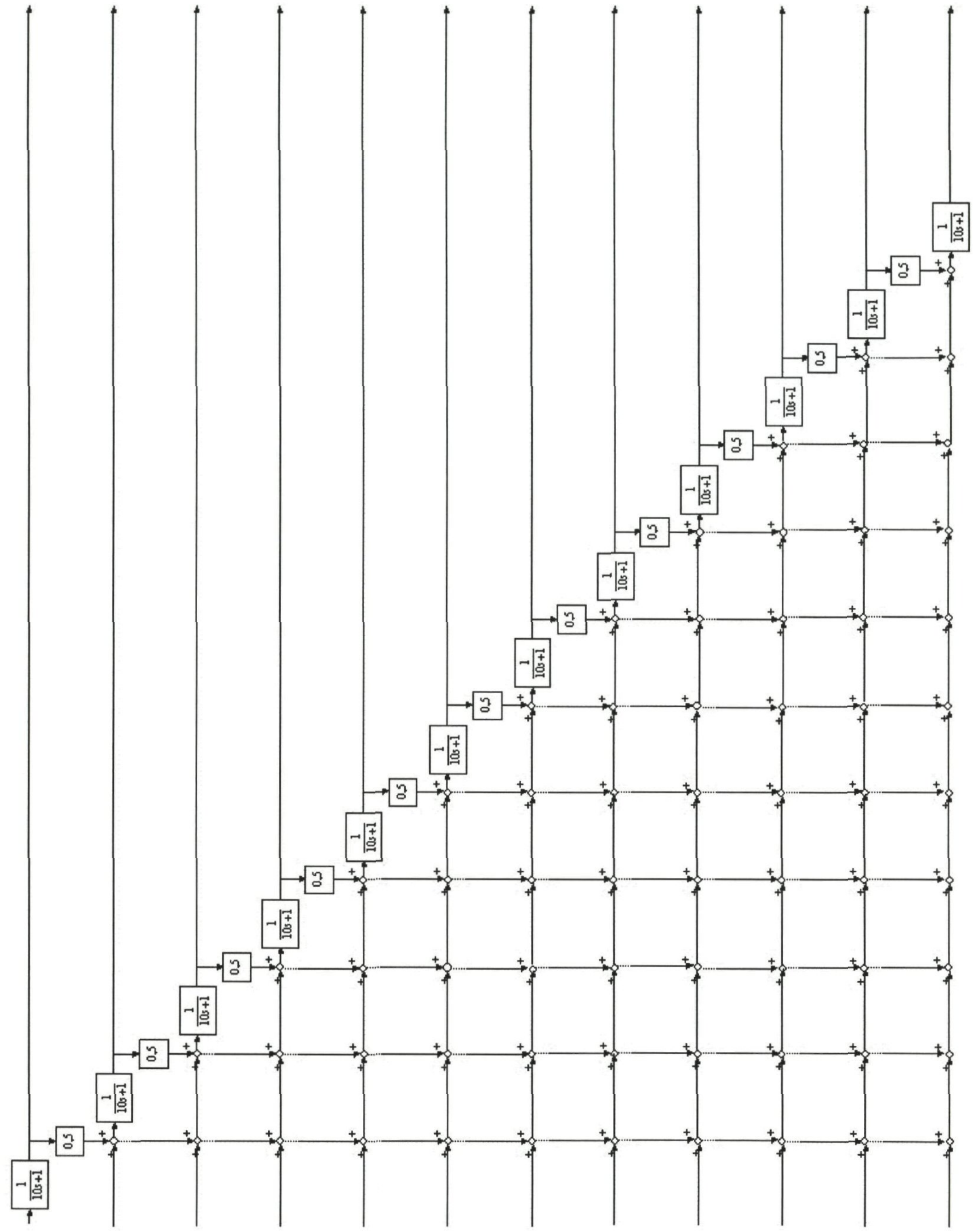


Figure 11: Entire 12-cell system (MARTIJN LANGELAAR)

Subject number: 1
ChemTech

Experiment	test number (in sequence)	RSME	Score	Remarks
Training		71	72	
E1	**6**	**70**	**82**	
E2	**7**	**65**	**86**	
E3	1	71	78	
E4	2	86	63	slider response is bad
E5	3	60	79	
E6	4	50	83	
E7	**5**	**50**	**88**	

slider scale

gets easier and easier
learningcurve ?

total time (appx) : 3hrs talks to computer?: no

trendinformation usefull : yes (not in the beginning)
difference 2 and 4 s ? : doesn't matter

best layout: doesn't matter, likes smaller font, alarms are visible anyway
(alarm/numbers)in/out))

Subject number: 2
ChemTech

Experiment	test number (in sequence)	RSME	Score	Remarks
Training		65	64	
E1	5	80	36	hard: first 4 cells start later
E2	6	70	57	
E3	**7**	**80**	**47**	
E4	1	60	67	
E5	**2**	**70**	**59**	
E6	**3**	**55**	**70**	coffeebreak helped ; left alone helped
E7	4	70	44	demotivated by errors in the beginning

2s clearly calmest.. also best/easiest

total time (appx) : 3,5 hr talks to computer?: yes

trendinformation usefull : yes
difference 2 and 4 s ? : none noticed

best layout: new (ML) stetpoint value above out value. vertical time=now line is nice/aesthetic ("Geeft rust") in Dutch (alarm/numbers)in/out)) Hi/Low warning old interface (Sally) too big.

Subject number: 3
Applied Physics

Experiment	test number (in sequence)	RSME	Score	Remarks
Training		55	73	
E1	**4**	**22**	**84**	
E2	5	35	84	
E3	6	17	87	
E4	7	38	77	
E5	1	35	80	
E6	**2**	**25**	**86**	
E7	**3**	**13**	**87**	

numbers are usefull for finetuning

4s little calmer, also easiest, best

total time (appx) : 3hr talks to computer?: no

trendinformation usefull : not really noticed
difference 2 and 4 s ? : not really noticed

best layout: new one - out above SP useful big alarms are ugly
(alarm/numbers)in/out))

Subject number: 4
Technical Informatics

Experiment	test nuber (in sequence)	RSME	Score	Remarks
Training		35	69	
E1	3	75	61	
E2	4	75	74	
E3	5	35	77	
E4	6	100	19	
E5	**7**	**40**	**65**	
E6	**1**	**55**	**78**	
E7	**2**	**70**	**78**	

2s calmest, good performance, not easiest

total time (appx) : 3,5 hr talks to computer?: yes

trendinformation usefull :yes
difference 2 and 4 s ? : no

best layout: old one (Sally) likes the big alarms
(alarm/numbers)in/out))
Subject number: 5

Mechanical Engineering

Experiment	test nuber (in sequence)	RSME	Score	Remarks
Training		26	43	
E1	2	26	76	
E2	3	26	79	easy; can calculate it all
E3	**4**	**26**	**73**	
E4	5	26	80	larger slider step, so calculates less
E5	**6**	**13**	**80**	**becomes boring**
E6	**7**	**13**	**81**	
E7	1	26	44	

can I use a calculator ??

0s calmest relatively easy and relatively good performance

4s difficult

total time (appx) : 2h talks to computer?: yes

trendinformation usefull : didn't look at trendinfo
difference 2 and 4 s ? : didn't look at trend info

best layout: new (ML) .. "overzichtelijker"
(alarm/numbers)in/out))
Subject number: 6
Applied Physics

Experiment	test number (in sequence)	RSME	Score	Remarks
Training		30	-	
E1	**1**	**32**	**84**	
E2	**2**	**58**	**73**	**i'm getting tired**
E3	**3**	**51**	**84**	
E4	4	49	81	
E5	5	61	68	
E6	6	66	73	
E7	7	58	73	

thinks RSME overall is average !!

2s NOT calm (tired) Highest RSME, lowest score

total time (appx) : 3h talks to computer?: no

trendinformation usefull : yes
difference 2 and 4 s ? : likes 4s better ; thinks history doesn't matter

best layout: new (ML) ; likes the smaller letter (alarm/numbers)in/out))
Subject number: 7
Mechanical Engineering

Experiment	test number (in sequence)	RSME	Score	Remarks
Training		57	74	
E1	**7**	**35**	**89**	
E2	1	65	79	
E3	2	37	85	
E4	3	70	81	
E5	4	70	78	
E6	**5**	**30**	**89**	
E7	**6**	**25**	**91**	

4s calmest, difficult, best performance

total time (appx) : 3 hrs talks to computer?: a little

trendinformation usefull : yes
difference 2 and 4 s ? : 2s better, 4s is too nervous

best layout: new (ML) … contra's old (Sally) *slider step *location numbers (alarm/numbers)in/out))

Best experiments of each type:

Experiments

E1	4
E2	2
E3	3
E5	3
E6	5
E7	4

Sessions

S1	2
S2	4
S3	3
S4	2
S5	2
S6	3
S7	7

best interface

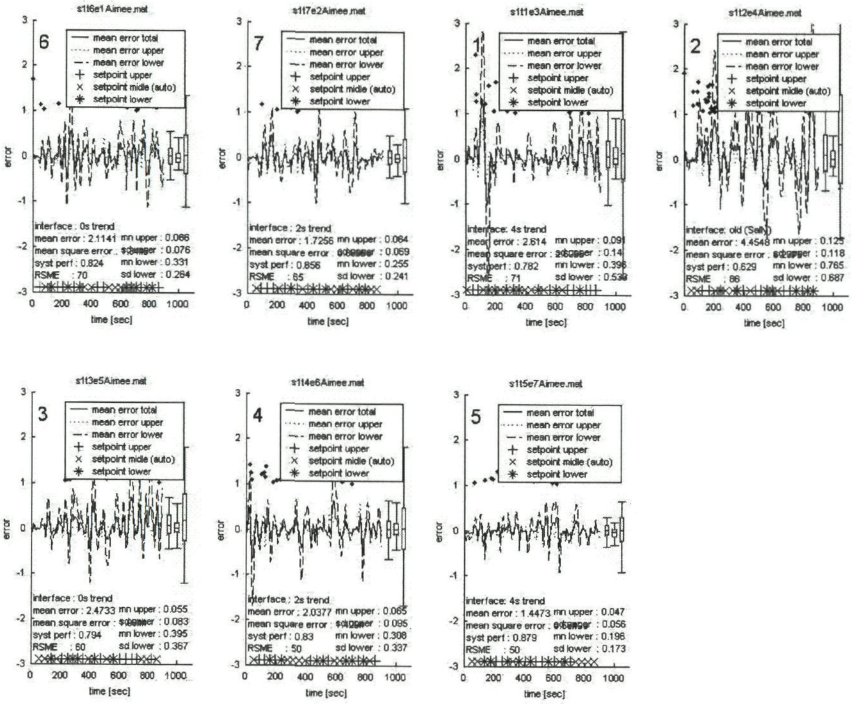
new	5
old	1
doesn’t matter	1

Best trend info

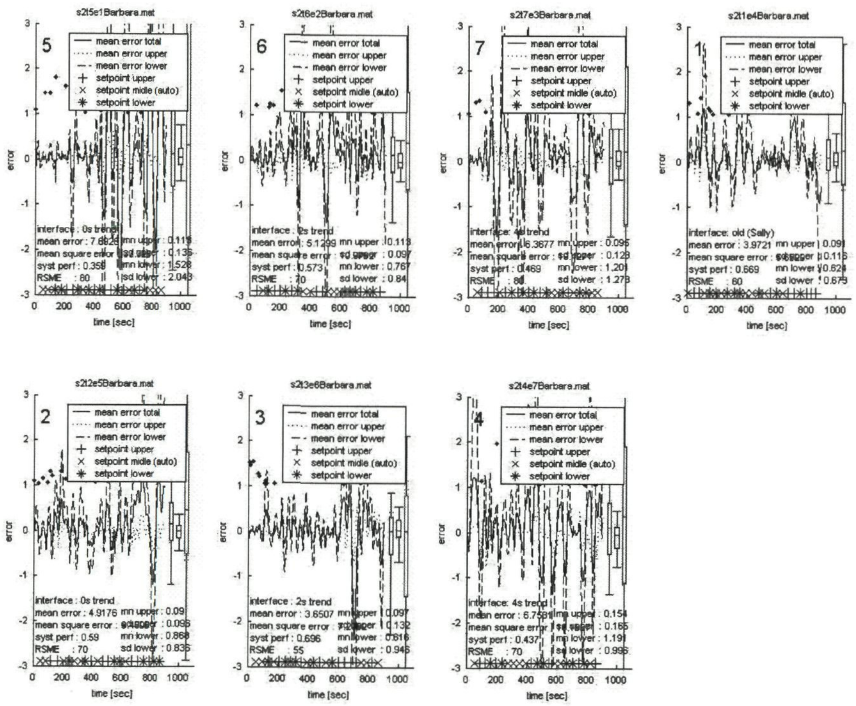
2s	1
4s	1
doesn’t matter	5

Appendix L: General Behavior Plots

Subject 1 :

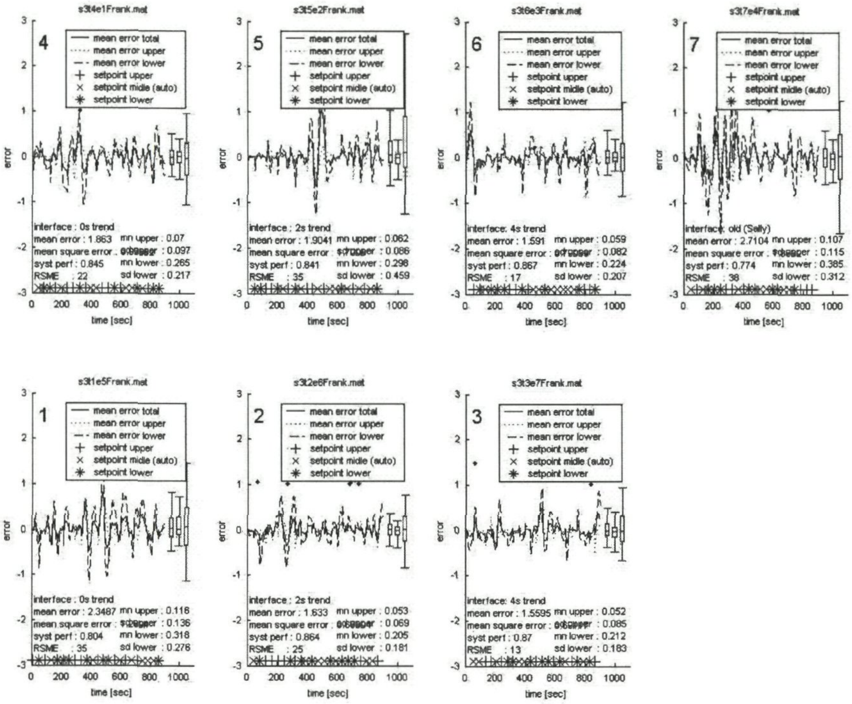


Subject 2:

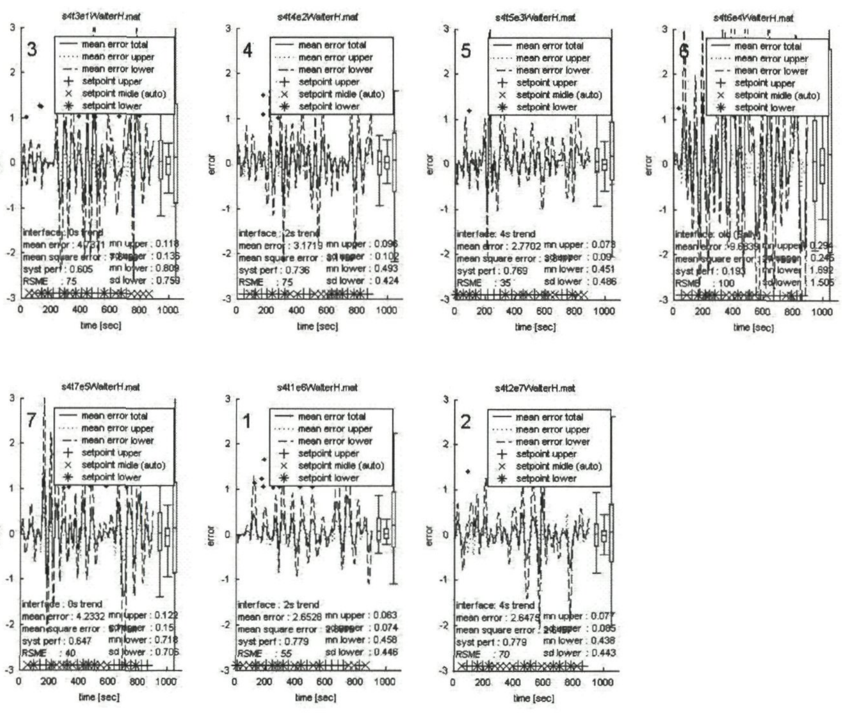


Appendix L: General Behavior Plots

Subject 3:

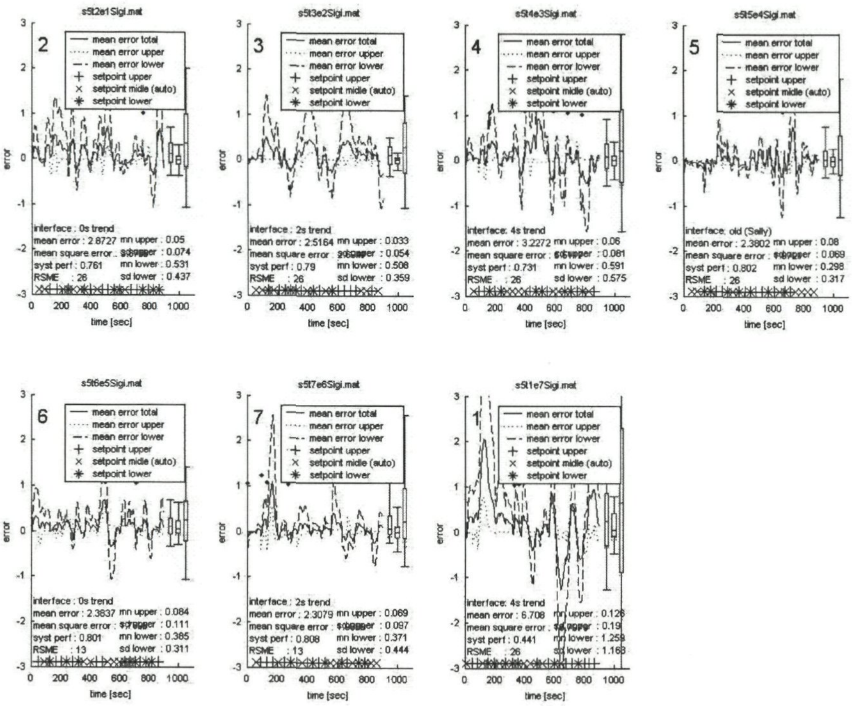


Subject 4:

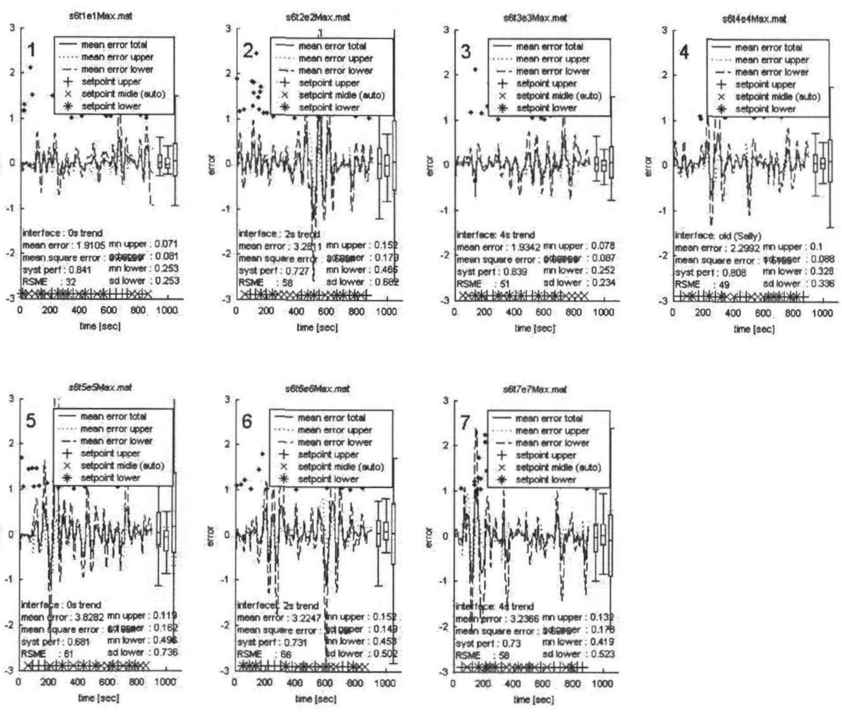


Appendix L: General Behavior Plots

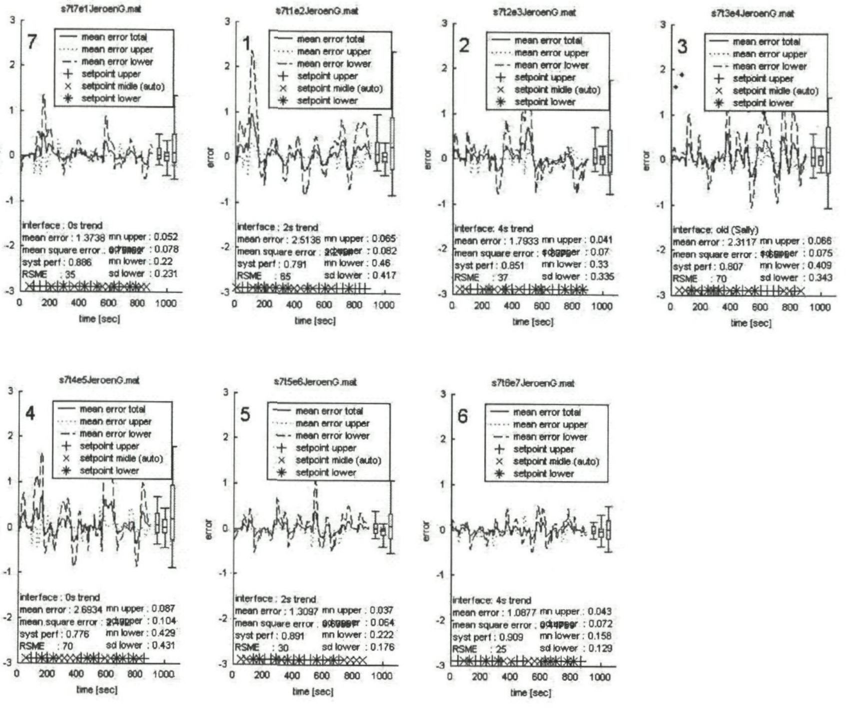
Subject 5:



Subject 6:



Subject 7:



Appendix M: Script to create general behavior plots

%M2Lanalyse.m

```
for subject=1:7
  for experiment=1:7
    filename_search=['s' num2str(subject) 't?e' num2str(experiment) '*.mat'];
    dir_found=dir(['s:\martijn\testdata\' filename_search]);
    filename=dir_found.name;
    load(['s:\martijn\testdata\' filename]);

tijd=transpose(1:length(simout_hist));

fout=simout_hist-sp_hist ;

fout_boven=(fout(:,1)+fout(:,2)+fout(:,3)+fout(:,4))/4 ;
fout_onder=(fout(:,9)+fout(:,10)+fout(:,11)+fout(:,12))/4 ;

fout_totaal=(fout(:,1)+fout(:,2)+fout(:,3)+fout(:,4)+fout(:,5)+fout(:,6)+fout(:,7)+fout(:,8)+fout(:,9)+fout(:,10)+fo
ut(:,11)+fout(:,12))/12;

disp(filename);
mn_fout_boven=mean(abs(fout_boven));
sd_fout_boven=std(abs(fout_boven));
mn_fout_onder=mean(abs(fout_onder));
sd_fout_onder=std(abs(fout_onder));

 sp_boven=[];
 sp_midden=[];
 sp_onder=[];

for loopje = 2:length(simout_hist);
  if (~ (sp_hist(loopje,1)==sp_hist(loopje-1,1) & sp_hist(loopje,2)==sp_hist(loopje-1,2) &
sp_hist(loopje,3)==sp_hist(loopje-1,3) & sp_hist(loopje,4)==sp_hist(loopje-1,4) ) )
    sp_boven=[sp_boven ; loopje];
  end % if
  if (~ (sp_hist(loopje,5)==sp_hist(loopje-1,5) & sp_hist(loopje,6)==sp_hist(loopje-1,6) &
sp_hist(loopje,7)==sp_hist(loopje-1,7) & sp_hist(loopje,8)==sp_hist(loopje-1,8) ) )
    sp_midden=[sp_midden ; loopje];
  end % if
  if (~ (sp_hist(loopje,9)==sp_hist(loopje-1,9) & sp_hist(loopje,10)==sp_hist(loopje-1,10) &
sp_hist(loopje,11)==sp_hist(loopje-1,11) & sp_hist(loopje,12)==sp_hist(loopje-1,12) ) )
    sp_onder=[sp_onder ; loopje];
  end % if
end %for

sp_boven_y=-2.9*ones(length(sp_boven),1);
sp_midden_y=-2.9*ones(length(sp_midden),1);
sp_onder_y=-2.9*ones(length(sp_onder),1);

 speed=speed';
 time_speed=tijd(1:length(simout_hist)-1);
 speed_1_25=[];
 speed__25=[];
  %for i =1 : length(speed)
 for i=1:(length(simout_hist)-1);
   if speed(i)>1
     if speed(i)>2.5
       speed__25=[speed__25 ; i 3.9];
     else
       speed_1_25=[speed_1_25 ; i speed(i) ];
     end%if
   end%if
 end%for

   l=size(sp_hist);
   l=l(1);
   c_error= sp_hist - simout_hist ;
   c_error_kw=c_error.*c_error;
   sum_error_kw=sum(c_error_kw);
   for i=1:12
     mean_error_kw(i)=sum_error_kw(i)/(l);
   end
   sum_mean_kw=sum(mean_error_kw);
   system_error_kw=sum_mean_kw/12;
```

Appendix M: Script to create general behavior plots

```
    score_val=(round(system_perf_fact*100));score_val=(round(system_perf_fact*100));
color='k';

 args_totaal=[color,'-'];
 args_boven=[color,':'];
 args_onder=[color,'--'];
 args_sp_boven=[color,' +'];
 args_sp_midden=[color,' x'];
 args_sp_onder=[color,' *'];

figure(subject);
 orient landscape;
 subplot(2,4,experiment);
 hold on;
 set(gca,'FontSize',5);
 plot
(tijd,fout_totaal,args_totaal,tijd,fout_boven,args_boven,tijd,fout_onder,args_onder,sp_boven,sp_boven_y,a
rgs_sp_boven,sp_midden,sp_midden_y,args_sp_midden,sp_onder,sp_onder_y,args_sp_onder);
 %plot (time_speed,speed,'y')
 legend('mean error total','mean error upper','mean error lower','setpoint upper','setpoint midle
(auto)','setpoint lower');
 args_speed_1_25=[color,' .'];
 args_speed__25=[color,' *'];

 if length(speed_1_25) >0
    plot(speed_1_25(:,1),speed_1_25(:,2),args_speed_1_25);
 end %if
 if length(speed__25)>0
    plot(speed__25(:,1),speed__25(:,2),args_speed__25)
 end %if
 xlim([0 1100]);
 YLim([-3 3]);
 set(gcf,'name',filename)
 title(filename);
 mn_fout_boven=round(mn_fout_boven*1000)/1000;
 sd_fout_boven=round(sd_fout_boven*1000)/1000;
 mn_fout_onder=round(mn_fout_onder*1000)/1000;
 sd_fout_onder=round(sd_fout_onder*1000)/1000;
 txt(1)={['mn upper : ' num2str(mn_fout_boven)]};
 txt(2)={['sd upper  : ' num2str(sd_fout_boven)]};
 txt(3)={['mn lower : ' num2str(mn_fout_onder)]};
 txt(4)={['sd lower  : ' num2str(sd_fout_onder)]};
text(600,-2.2,txt,'FontSize',5);
system_perf_fact=round(system_perf_fact*1000)/1000;
if GUinterface==1
   txt2(1)={'interface : 0s trend'};
elseif GUinterface==5
   txt2(1)={'interface : 2s trend'};
elseif GUinterface==6
   txt2(1)={'interface: 4s trend'};
elseif GUinterface==0
   txt2(1)={'interface: old (Sally)'};
  end %if GUinterface
txt2(2)={['mean error : ' num2str(sum_mean)]};
txt2(3)={['mean square error : ' num2str(sum_mean_kw)]};
txt2(4)={['syst perf : ' num2str(system_perf_fact)]};
txt2(5)={['RSME     : ' num2str(RSME.overall)]};
text(18,-2.1,txt2,'FontSize',5);
text(45,2.5,filename(4),'FontSize',10);
XLabel('time [sec]');
YLabel('error');
boxplot_av(25,950,fout_totaal');
boxplot_av(25,1000,fout_boven');
boxplot_av(25,1050,fout_onder');
end %for subject
end %for exerpiment
```

%analyseml2.m

```
%     save(fname,'sum_mean','system_error','system_perf_fact',...
%
'sp_hist','input_sim_hist','simout_hist','seemouse_hist','time','speed','system','name','GUinterface','slider_ch
ange','slider_reset');
%   save([pathname filename], 'RSME','filename','pathname','-APPEND');
%     save([pathname filename], 'seemouse_hist','smh_old','-APPEND');
% sum_mean is the sum of all errors. It is used to calculate the score,

% c_error((t_g_old + 1),:) = (sp-simout);
% sum_error=sum(abs(c_error));
%for i=1:12
%   mean_error(i)=sum_error(i)/(simtime+1);
%end
% sum_mean=sum(mean_error);
% system_error=sum_mean/12;
%       l=size(sp_hist);
%       l=l(1);
%       c_error= sp_hist - simout_hist ;
%       sum_error=sum(abs(c_error));
%       for i=1:12
%         mean_error(i)=sum_error(i)/(l);
%       end
%       sum_mean=sum(mean_error);
%       system_error=sum_mean/12;
%       system_perf_fact=1-system_error;
%       score_val=(round(system_perf_fact*100));score_val=(round(system_perf_fact*100));

%analyseml2

%m2lanalyse

global data

for subject=1:7
  for experiment=1:7
    filename_search=['s' num2str(subject) 't?e' num2str(experiment) '*.mat'];
    dir_found=dir(['s:\martijn\testdata\' filename_search]);
    filename=dir_found.name;
    load(['s:\martijn\testdata\' filename]);
    s_l=str2num([num2str(subject) num2str(experiment)]);
    data(s_l).sum_mean=sum_mean;
    data(s_l).system_error=system_error;
    data(s_l).system_perf_fact=system_perf_fact;
    data(s_l).sp_hist=sp_hist;
    data(s_l).input_sim_hist=input_sim_hist;
    data(s_l).seemouse_hist=seemouse_hist;
    data(s_l).time=time;
    data(s_l).speed=speed;
    data(s_l).system=system;
    data(s_l).name=name;
    data(s_l).GUinterface=GUinterface;
    data(s_l).slider_change=slider_change;
    data(s_l).silder_reset=slider_reset;
    data(s_l).RSME=RSME;
    data(s_l).test=filename(4); % test nummer in volgorde waarin ze zijn uitgevoerd
    disp([filename ' ingelezen']);

    % kwadratische fout en score berekenen

    l=size(sp_hist);
    l=l(1);
    c_error= sp_hist - simout_hist ;
    c_error_kw=c_error.*c_error;
    sum_error_kw=sum(c_error_kw);
    for i=1:12
      mean_error_kw(i)=sum_error_kw(i)/(l);
    end
    sum_mean_kw=sum(mean_error_kw);
    system_error_kw=sum_mean_kw/12;
    score_val=(round(system_perf_fact*100));score_val=(round(system_perf_fact*100));
```

```
data(s_l).sum_mean_kw=sum_mean_kw;
data(s_l).system_error_kw=system_error_kw;
data(s_l).score=score_val;

%muisbewegingen analyseren
% mouse_movement
%
% seemouse_hist :   seemouse  ([x y])  toc tsimulation
% slider_change :    waarde toc tsimulation mouse_loc ([x y])
% slider_reset : waarde toc tsimulation mouse_loc
% mouse_matrix: [ x y toc tsimulation ]


numb_reset_one=size(slider_reset.one); numb_reset_one=numb_reset_one(1);
numb_reset_two=size(slider_reset.two); numb_reset_two=numb_reset_two(1);
numb_reset_three=size(slider_reset.three); numb_reset_three=numb_reset_three(1);
numb_reset_four=size(slider_reset.four); numb_reset_four=numb_reset_four(1);
numb_reset_five=size(slider_reset.five); numb_reset_five=numb_reset_five(1);
numb_reset_six=size(slider_reset.six); numb_reset_six=numb_reset_six(1);
numb_reset_seven=size(slider_reset.seven); numb_reset_seven=numb_reset_seven(1);
numb_reset_eight=size(slider_reset.eight); numb_reset_eight=numb_reset_eight(1);
numb_reset_nine=size(slider_reset.nine); numb_reset_nine=numb_reset_nine(1);
numb_reset_ten=size(slider_reset.ten); numb_reset_ten=numb_reset_ten(1);
numb_reset_eleven=size(slider_reset.eleven); numb_reset_eleven=numb_reset_eleven(1);
numb_reset_twelve=size(slider_reset.twelve); numb_reset_twelve=numb_reset_twelve(1);
numb_reset_upper=numb_reset_one+numb_reset_two+numb_reset_three+numb_reset_four;
numb_reset_middle=numb_reset_five+numb_reset_six+numb_reset_seven+numb_reset_eight;
numb_reset_lower=numb_reset_nine+numb_reset_ten+numb_reset_eleven+numb_reset_twelve;
numb_reset_total=numb_reset_upper+numb_reset_middle+numb_reset_lower;

numb_sl_ch_one=size(slider_change.one); numb_sl_ch_one=numb_sl_ch_one(1);
numb_sl_ch_two=size(slider_change.two); numb_sl_ch_two=numb_sl_ch_two(1);
numb_sl_ch_three=size(slider_change.three); numb_sl_ch_three=numb_sl_ch_three(1);
numb_sl_ch_four=size(slider_change.four); numb_sl_ch_four=numb_sl_ch_four(1);
numb_sl_ch_five=size(slider_change.five); numb_sl_ch_five=numb_sl_ch_five(1);
numb_sl_ch_six=size(slider_change.six); numb_sl_ch_six=numb_sl_ch_six(1);
numb_sl_ch_seven=size(slider_change.seven); numb_sl_ch_seven=numb_sl_ch_seven(1);
numb_sl_ch_eight=size(slider_change.eight); numb_sl_ch_eight=numb_sl_ch_eight(1);
numb_sl_ch_nine=size(slider_change.nine); numb_sl_ch_nine=numb_sl_ch_nine(1);
numb_sl_ch_ten=size(slider_change.ten); numb_sl_ch_ten=numb_sl_ch_ten(1);
numb_sl_ch_eleven=size(slider_change.eleven); numb_sl_ch_eleven=numb_sl_ch_eleven(1);
numb_sl_ch_twelve=size(slider_change.twelve); numb_sl_ch_twelve=numb_sl_ch_twelve(1);
numb_sl_ch_upper=numb_sl_ch_one+numb_sl_ch_two+numb_sl_ch_three+numb_sl_ch_four;
numb_sl_ch_middle=numb_sl_ch_five+numb_sl_ch_six+numb_sl_ch_seven+numb_sl_ch_eight;
numb_sl_ch_lower=numb_sl_ch_nine+numb_sl_ch_ten+numb_sl_ch_eleven+numb_sl_ch_twelve;
numb_sl_ch_total=numb_sl_ch_upper+numb_sl_ch_middle+numb_sl_ch_lower;
numb_sl_ch_total_man=numb_sl_ch_upper+numb_sl_ch_lower;

mouse_clicks=numb_sl_ch_total +  numb_reset_total;
mouse_matrix=[];

mouse_matrix=seemouse_hist;
    if length(slider_change.one)>0 mouse_matrix=[mouse_matrix ; slider_change.one(:,4:5)
slider_change.one(:,2:3)]; end;
    if length(slider_change.two)>0 mouse_matrix=[mouse_matrix ; slider_change.two(:,4:5)
slider_change.two(:,2:3)]; end;
    if length(slider_change.three)>0 mouse_matrix=[mouse_matrix ; slider_change.three(:,4:5)
slider_change.three(:,2:3)]; end;
    if length(slider_change.four)>0 mouse_matrix=[mouse_matrix ; slider_change.four(:,4:5)
slider_change.four(:,2:3)]; end;
    if length(slider_change.five)>0 mouse_matrix=[mouse_matrix ; slider_change.five(:,4:5)
slider_change.five(:,2:3)]; end;
    if length(slider_change.six)>0 mouse_matrix=[mouse_matrix ; slider_change.six(:,4:5)
slider_change.six(:,2:3)]; end;
    if length(slider_change.seven)>0 mouse_matrix=[mouse_matrix ; slider_change.seven(:,4:5)
slider_change.seven(:,2:3)]; end;
    if length(slider_change.eight)>0 mouse_matrix=[mouse_matrix ; slider_change.eight(:,4:5)
slider_change.eight(:,2:3)]; end;
    if length(slider_change.nine)>0 mouse_matrix=[mouse_matrix ; slider_change.nine(:,4:5)
slider_change.nine(:,2:3)]; end;
    if length(slider_change.ten)>0 mouse_matrix=[mouse_matrix ; slider_change.ten(:,4:5)
slider_change.ten(:,2:3)]; end;
    if length(slider_change.eleven)>0 mouse_matrix=[mouse_matrix ; slider_change.eleven(:,4:5)
slider_change.eleven(:,2:3)]; end;
```

```
    if length(slider_change.twelve)>0 mouse_matrix=[mouse_matrix ; slider_change.twelve(:,4:5)
slider_change.twelve(:,2:3)]; end;

    if length(slider_reset.one)>0 mouse_matrix=[mouse_matrix ; slider_reset.one(:,4:5)
slider_reset.one(:,2:3)]; end;
    if length(slider_reset.two)>0 mouse_matrix=[mouse_matrix ; slider_reset.two(:,4:5)
slider_reset.two(:,2:3)]; end;
    if length(slider_reset.three)>0 mouse_matrix=[mouse_matrix ; slider_reset.three(:,4:5)
slider_reset.three(:,2:3)]; end;
    if length(slider_reset.four)>0 mouse_matrix=[mouse_matrix ; slider_reset.four(:,4:5)
slider_reset.four(:,2:3)]; end;
    if length(slider_reset.five)>0 mouse_matrix=[mouse_matrix ; slider_reset.five(:,4:5)
slider_reset.five(:,2:3)]; end;
    if length(slider_reset.six)>0 mouse_matrix=[mouse_matrix ; slider_reset.six(:,4:5)
slider_reset.six(:,2:3)]; end;
    if length(slider_reset.seven)>0 mouse_matrix=[mouse_matrix ; slider_reset.seven(:,4:5)
slider_reset.seven(:,2:3)]; end;
    if length(slider_reset.eight)>0 mouse_matrix=[mouse_matrix ; slider_reset.eight(:,4:5)
slider_reset.eight(:,2:3)]; end;
    if length(slider_reset.nine)>0 mouse_matrix=[mouse_matrix ; slider_reset.nine(:,4:5)
slider_reset.nine(:,2:3)]; end;
    if length(slider_reset.ten)>0 mouse_matrix=[mouse_matrix ; slider_reset.ten(:,4:5)
slider_reset.ten(:,2:3)]; end;
    if length(slider_reset.eleven)>0 mouse_matrix=[mouse_matrix ; slider_reset.eleven(:,4:5)
slider_reset.eleven(:,2:3)]; end;
    if length(slider_reset.twelve)>0 mouse_matrix=[mouse_matrix ; slider_reset.twelve(:,4:5)
slider_reset.twelve(:,2:3)]; end;

    mouse_matrix=sortrows(mouse_matrix,[3 4]);
    dista=0;
    dista_tot=0;
    ang=0;
    ang_tot=0;
    ang_count_0_45=0;
    ang_count_45_90=0;
    ang_count_90_135=0;
    ang_count_135_180=0;
    ang_vector = [];
    maxang=0;
    for counter=1:(length(mouse_matrix)-1)
      counter2=counter+1;
      dista= sqrt( ( mouse_matrix(counter,1)-mouse_matrix(counter2,1))^2 +( mouse_matrix(counter,2)-
mouse_matrix(counter2,2))^2   );
      dista_tot=dista_tot+dista;
    end %for
    for counter=1:(length(mouse_matrix)-2)
      counter2=counter+1;
      counter3=counter2+1;


    %vertical up & down

      if (    mouse_matrix(counter2,1) - mouse_matrix(counter,1)       ) ==0
        if (mouse_matrix(counter2,2)   - mouse_matrix(counter,2)    ) > 0
          a=(1/2)*pi;
        else
          a=-1*(1/2)*pi;
        end %if
      else
        a=atan(   (mouse_matrix(counter2,2)   - mouse_matrix(counter,2)     ) / (
mouse_matrix(counter2,1) - mouse_matrix(counter,1)        )                        );
      end %if

      if (    mouse_matrix(counter3,1) - mouse_matrix(counter2,1)       ) ==0
        if (mouse_matrix(counter3,2)   - mouse_matrix(counter2,2)    ) > 0
          b=(1/2)*pi;
        else
          b=-1*(1/2)*pi;
        end %if
      else
        b=atan(   (mouse_matrix(counter3,2)   - mouse_matrix(counter2,2)     ) / (
mouse_matrix(counter3,1) - mouse_matrix(counter2,1)        )                        );
      end %if
```

```
      ang=a-b;
      if ang >pi ang=ang-2*pi; end;
      if ang < -pi ang=ang+2*pi ;end;
      ang=abs(ang);
      ang_tot=ang_tot+ang;
      ang_vector=[ang_vector ; ang];

      if  ang<((1/4)*pi)
        ang_count_0_45=ang_count_0_45+1;
      elseif ang<((1/2)*pi)
        ang_count_45_90=ang_count_45_90+1;
      elseif  ang<((3/4)*pi)
        ang_count_90_135=ang_count_90_135+1;
      else
        ang_count_135_180=ang_count_135_180+1;
      end %if

    end %for

    data(s_l).ang_count_0_45=ang_count_0_45;
    data(s_l).ang_count_45_90=ang_count_45_90;
    data(s_l).ang_count_90_135=ang_count_90_135;
    data(s_l).ang_count_135_180=ang_count_135_180;
    data(s_l).ang_count_0_90=ang_count_0_45+ang_count_45_90;
    data(s_l).ang_count_90_180=ang_count_90_135+ang_count_135_180;
    data(s_l).mouse_clicks=mouse_clicks;
    data(s_l).dist=dista_tot;
    data(s_l).ang_vector=ang_vector;
    data(s_l).control_actions=numb_sl_ch_total_man;
    data(s_l).eff_q_contact_mscl=numb_sl_ch_total_man/mouse_clicks;
    data(s_l).eff_q_msdist_dirch_0_90=dista_tot/(ang_count_0_45+ang_count_45_90);
    data(s_l).eff_q_msdist_dirch_90_180=dista_tot/(ang_count_90_135+ang_count_135_180);
    data(s_l).eff_p_msdist_dirch_0_90=dista_tot*(ang_count_0_45+ang_count_45_90);
    data(s_l).eff_p_msdist_dirch_90_180=dista_tot*(ang_count_90_135+ang_count_135_180);


  end % for experiment
end % for subject

subject_vect=[ 1 2 3 4 5 6 7 1 2 3 4 5 6 7 ];

%% volgorde  bepalen
for subject=1:7
  sequence_n=[];
  for zoeken_naar=1:7
    experi=1;
    experiment_found=0;
    while experiment_found==0
      s_l=str2num([num2str(subject) num2str(experi)]);
      getal=str2num(data(s_l).test);
      if getal==zoeken_naar
        experiment_found=experi;
        sequence_n=[sequence_n experi];
      else
        experi=experi+1;
      end %if
    end %while
  end %for zoeken_naar
  data(subject).sequence=sequence_n;
end %for subject


analyseml3
```

%analyseml3.m

```
% sum_mean uitzetten voor tegen 0s, 2s, 4s ... nummers proefpersonen erbij
% evt zelfde doen voor standaarddeviatie.

% verctor voor 0s (E1 en E5)

global data
```

```
% subject_vect=[ 1 2 3 4 5 6 7 1 2 3 4 5 6 7 ];
%
% %% volgorde bepalen
% for subject=1:7
%    sequence_n=[];
%    for zoeken_naar=1:7
%       experi=1;
%       experiment_found=0;
%       while experiment_found==0
%          s_l=str2num([num2str(subject) num2str(experi)]);
%          getal=str2num(data(s_l).test);
%          if getal==zoeken_naar
%             experiment_found=experi;
%             sequence_n=[sequence_n experi];
%          else
%             experi=experi+1;
%          end %if
%       end %while
%    end %for zoeken_naar
%    data(subject).sequence=sequence_n;
% end %for subject

analyseml_1;  %sum_mean
analyseml_2;  %RSME
analyseml_3;  % sum_mean_square
analyseml_14 % system_perf_fact
analyseml_4;  % mouse _dist
analyseml_5; % direction changes 0-90
analyseml_6 % direction changes 90-180
analyseml_7 %mouse clicks
analyseml_8 %control actions
analyseml_9 % eff_q_contact_mscl  (control actions/mouseclicks)
analyseml_10 %eff_q_msdist_dirch_0_90 (mousedistande/direction changes)
analyseml_11 %eff_q_msdist_dirch_90_180 (mousedistande/direction changes)
analyseml_12 %eff_p_msdist_dirch_0_90 (mousedistande/direction changes)
analyseml_13 %eff_p_msdist_dirch_90_180 (mousedistande/direction changes)



% for anal=1:14 fig=((anal-1)*4 + 3); figure (fig); print(fig); end

%RSME tegen sum_mean

global RSME sum_mean_0 sum_mean_2 sum_mean_4 sum_mean_s sum_mean_all

figure;
plot (RSME.overall_0,sum_mean_0,' k+',RSME.overall_2,sum_mean_2,'
kx',RSME.overall_4,sum_mean_4,' k*', RSME.overall_s,sum_mean_s,' k.');
orient landscape;
axis([0 100 0 8]);
Title('performance vs workload');
XLabel('RSME');
YLabel('mean error (sum mean)');
legend('0s','2s','4s','old');

% mouse behavior / efficiency plotjes

global dist_0 dist_2 dist_4 dist_s dist_all ang_count_0_90_0 ang_count_0_90_2 ang_count_0_90_4
ang_count_0_90_s ang_count_0_90_all
global ang_count_90_180_0 ang_count_90_180_2 ang_count_90_180_4 ang_count_90_180_s
ang_count_90_180_all
global mouse_clicks_0 mouse_clicks_2 mouse_clicks_4 mouse_clicks_s mouse_clicks_all
global control_actions_0 control_actions_2 control_actions_4 control_actions_s control_actions_all

figure;
orient landscape;
subplot(2,3,1);
plot(dist_0,ang_count_0_90_0,' k+',dist_2,ang_count_0_90_2,' kx',dist_4,ang_count_0_90_4,'
k*',dist_s,ang_count_0_90_s,' k.');
%axis([0 300 0 4000]);
Title('mouse behavior');
XLabel('mouse dist');
YLabel('mouse direction changes (0-90deg)');
```

```
legend('0s','2s','4s','old');

subplot(2,3,4);
plot(dist_0,ang_count_90_180_0,' k+',dist_2,ang_count_90_180_2,' kx',dist_4,ang_count_90_180_4,'
k*',dist_s,ang_count_90_180_s,' k.');
%axis([0 300 0 700]);
Title('mouse behavior');
XLabel('mouse dist');
YLabel('mouse direction changes (90-180deg)');
legend('0s','2s','4s','old');

subplot(2,3,3)
plot(mouse_clicks_0,control_actions_0,' k+',mouse_clicks_2,control_actions_2,'
kx',mouse_clicks_4,control_actions_4,' k*',mouse_clicks_s,control_actions_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior');
XLabel('mouse clicks');
YLabel('control actions');
legend('0s','2s','4s','old');

subplot(2,3,6)
plot(mouse_clicks_0,dist_0,' k+',mouse_clicks_2,dist_2,' kx',mouse_clicks_4,dist_4,'
k*',mouse_clicks_s,dist_s,' k.')
%axis([200 300 100 200]);
Title('mouse behavior');
XLabel('mouse clicks');
YLabel('mouse distance');
legend('0s','2s','4s','old');

subplot(2,3,2)
plot(mouse_clicks_0,ang_count_0_90_0,' k+',mouse_clicks_2,ang_count_0_90_2,'
kx',mouse_clicks_4,ang_count_0_90_4,' k*',mouse_clicks_s,ang_count_0_90_s,' k.');
%axis([200 300 100 200]);
Title('mouse behavior');
XLabel('mouse clicks');
YLabel('mouse direction changes (0-90deg)');
legend('0s','2s','4s','old');

subplot(2,3,5)
plot(mouse_clicks_0,ang_count_90_180_0,' k+',mouse_clicks_2,ang_count_90_180_2,'
kx',mouse_clicks_4,ang_count_90_180_4,' k*',mouse_clicks_s,ang_count_90_180_s,' k.');
%axis([200 300 100 200]);
Title('mouse behavior');
XLabel('mouse clicks');
YLabel('mouse direction changes (90-180deg)');
legend('0s','2s','4s','old');

figure;
orient landscape;
plot(mouse_clicks_0,control_actions_0,' k+',mouse_clicks_2,control_actions_2,'
kx',mouse_clicks_4,control_actions_4,' k*',mouse_clicks_s,control_actions_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior');
XLabel('mouse clicks');
YLabel('control actions');
legend('0s','2s','4s','old');


% sum_mean_error met boxplots ook zonder verworpen punten
figure;
orient landscape;
% vect_1=[ 1:21  1:21];
% vect_2=[sum_mean_0 sum_mean_2 sum_mean_4 ];
% hold on;
% plot(vect_1,vect_2, ' kx');
% plot(22:28,sum_mean_s,' kx');
plot([1:7 1:7], sum_mean_0,' k+',[10:16 10:16],sum_mean_2,' kx',[19:25 19:25],sum_mean_4,'
k*',[28:34],sum_mean_s,' .k');
axis([-2 37 0 8]);
Title('0, 2, and 4 sec trendinformation (INCLUDING mean, std, min, max without rejected points)');
XLabel('subject number');
YLabel('Mean absolute error (sum mean)');
set(gca,'XTick',1:37);
%label=[];
```

```
% set(gca,'XTickLabel',{'1' ;'2' ; '3' ; '4';'5';'6';'7'; '1' ;'2' ; '3' ; '4';'5';'6';'7';'1' ;'2' ; '3' ; '4';'5';'6';'7' });
set(gca,'XTickLabel',{'1' ;'2' ; '3' ; '4';'5';'6';'7';'';'';'1' ;'2' ; '3' ; '4';'5';'6';'7';'';'';'1' ;'2' ; '3' ; '4';'5';'6';'7';'';'';'1' ;'2' ; '3'
; '4';'5';'6';'7';'' ;'';''});
teksta='(0 sec)';
tekstb='(2 sec)';
tekstc='(4 sec)';
tekstd=('(old)');
legend('0s','2s','4s','old');
text(3,-0.3,teksta);
text(10,-0.3,tekstb);
text(20,-0.3,tekstc);
text(28,-0.3,tekstd);

% BOX-PLOTS  WEERGEVEN
boxplot_av(0.5,8,sum_mean_0);
boxplot_av(0.5,17,sum_mean_2);
boxplot_av(0.5,26,sum_mean_4);
boxplot_av(0.5,35,sum_mean_s);
sum_mean_all=[sum_mean_0 sum_mean_2 sum_mean_4 sum_mean_s];
boxplot_av(0.5,36,sum_mean_all);

%BOX-Plots zonder verworpen punten
acc_sum_mean_0=sort(sum_mean_0);
acc_sum_mean_0=acc_sum_mean_0(1:9);
boxplot_av(0.5,0,acc_sum_mean_0);

acc_sum_mean_4=sort(sum_mean_4);
acc_sum_mean_4=acc_sum_mean_4(1:11);
boxplot_av(0.5,18,acc_sum_mean_4);

boxplot_av(0.5,-1,[acc_sum_mean_0 sum_mean_2 acc_sum_mean_4 sum_mean_s]);


%wat waarden numeriek uitrekenen en weergeven
disp('mean error mean (all points)');
mean(sum_mean_all)
disp('mean error std (all points)');
std(sum_mean_all)
disp('mean error std (all points)');
std_ass(sum_mean_all)

acc_sum_mean_0=sort(sum_mean_0);
acc_sum_mean_0=acc_sum_mean_0(1:9);
acc_sum_mean_4=sort(sum_mean_4);
acc_sum_mean_4=acc_sum_mean_4(1:11);
acc_sum_mean_all=[acc_sum_mean_0 sum_mean_2 acc_sum_mean_4 sum_mean_s];

disp('mean error mean (accepted points)');
mean(acc_sum_mean_all)
disp('mean error std (accepted points)');
std(acc_sum_mean_all)
disp('mean error std (accepted points)');
std_ass(acc_sum_mean_all)



disp('RSME mean (all points)');
mean(RSME.overall_all)
disp('RSME std (all points)');
std(RSME.overall_all)
disp('RSME assymetrical std');
std_ass(RSME.overall_all)

acc_RSME_overall_0=sort(RSME.overall_0);
acc_RSME_overall_0=acc_RSME_overall_0(1:9);
acc_RSME_overall_4=sort(RSME.overall_4);
acc_RSME_overall_4=acc_RSME_overall_4(1:11);
acc_RSME_all=[acc_RSME_overall_0 RSME.overall_2 acc_RSME_overall_4 RSME.overall_s];

disp('RSME mean (accepted points)');
mean(acc_RSME_all)
disp('RSME std (accepted points)');
std(acc_RSME_all)
disp('assymetrical std (accepted points)');
```

```
std_ass(acc_RSME_all)

global system_perf_fact_0 system_perf_fact_2 system_perf_fact_4 system_perf_fact_s
system_perf_fact_all

disp('system performance factor mean (all points)');
mean(system_perf_fact_all)
disp('systpem performance factor (all points)');
std(system_perf_fact_all)
disp('system performance factor std (all points)');
std_ass(system_perf_fact_all)

acc_system_perf_fact_0=sort(system_perf_fact_0);
acc_system_perf_fact_0=acc_system_perf_fact_0(1:9);
acc_system_perf_fact_4=sort(system_perf_fact_4);
acc_system_perf_fact_4=acc_system_perf_fact_4(1:11);
acc_system_perf_fact_all=[acc_system_perf_fact_0 system_perf_fact_2 acc_system_perf_fact_4
system_perf_fact_s];

disp('system performance factor mean (accepted points)');
mean(acc_system_perf_fact_all)
disp('system performance factor std (accepted points)');
std(acc_system_perf_fact_all)
disp('system performance factor assymetrical std (accepted points)');
std_ass(acc_system_perf_fact_all)

% mouse behavior versus human factors

figure;
orient landscape;
subplot(2,3,1)
plot(sum_mean_0,dist_0,' k+',sum_mean_2,dist_2,' kx',sum_mean_4,dist_4,' k*',sum_mean_s,dist_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs performance');
XLabel('mean error');
YLabel('mouse distance');
legend('0s','2s','4s','old');

subplot(2,3,5)
plot(sum_mean_0,control_actions_0,' k+',sum_mean_2,control_actions_2,'
kx',sum_mean_4,control_actions_4,' k*',sum_mean_s,control_actions_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs performance');
XLabel('mean error');
YLabel('control actions');
legend('0s','2s','4s','old');

subplot(2,3,4)
plot(sum_mean_0,mouse_clicks_0,' k+',sum_mean_2,mouse_clicks_2,' kx',sum_mean_4,mouse_clicks_4,'
k*',sum_mean_s,mouse_clicks_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs performance');
XLabel('mean error');
YLabel('mouse clicks');
legend('0s','2s','4s','old');

subplot(2,3,3)
plot(sum_mean_0,ang_count_0_90_0,' k+',sum_mean_2,ang_count_0_90_2,'
kx',sum_mean_4,ang_count_0_90_4,' k*',sum_mean_s,ang_count_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs performance');
XLabel('mean error');
YLabel('mouse clicks');
legend('0s','2s','4s','old');

subplot(2,3,6)
plot(sum_mean_0,ang_count_90_180_0,' k+',sum_mean_2,ang_count_90_180_2,'
kx',sum_mean_4,ang_count_90_180_4,' k*',sum_mean_s,ang_count_90_180_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs performance');
XLabel('mean error');
YLabel('mouse clicks');
legend('0s','2s','4s','old');
```

```
% mouse behavior vs RSME

figure;
orient landscape;
subplot(2,3,1)
plot(RSME.overall_0,dist_0,' k+',RSME.overall_2,dist_2,' kx',RSME.overall_4,dist_4,'
k*',RSME.overall_s,dist_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs mental load');
XLabel('RSME');
YLabel('mouse distance');
legend('0s','2s','4s','old');

subplot(2,3,5)
plot(RSME.overall_0,control_actions_0,' k+',RSME.overall_2,control_actions_2,'
kx',RSME.overall_4,control_actions_4,' k*',RSME.overall_s,control_actions_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs mental load');
XLabel('RSME');
YLabel('control actions');
legend('0s','2s','4s','old');

subplot(2,3,4)
plot(RSME.overall_0,mouse_clicks_0,' k+',RSME.overall_2,mouse_clicks_2,'
kx',RSME.overall_4,mouse_clicks_4,' k*',RSME.overall_s,mouse_clicks_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs mental load');
XLabel('RSME');
YLabel('mouse clicks');
legend('0s','2s','4s','old');

subplot(2,3,3)
plot(RSME.overall_0,ang_count_0_90_0,' k+',RSME.overall_2,ang_count_0_90_2,'
kx',RSME.overall_4,ang_count_0_90_4,' k*',RSME.overall_s,ang_count_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs mental load');
XLabel('RSME');
YLabel('mouse clicks');
legend('0s','2s','4s','old');

subplot(2,3,6)
plot(RSME.overall_0,ang_count_90_180_0,' k+',RSME.overall_2,ang_count_90_180_2,'
kx',RSME.overall_4,ang_count_90_180_4,' k*',RSME.overall_s,ang_count_90_180_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior vs mental load');
XLabel('RSME');
YLabel('mouse clicks');
legend('0s','2s','4s','old');


%analyseml_9 % eff_q_contact_mscl  (control actions/mouseclicks)
%analyseml_10 %eff_q_msdist_dirch_0_90 (mousedistande/direction changes)
%analyseml_11 %eff_q_msdist_dirch_90_180 (mousedistande/direction changes)
%analyseml_12 %eff_p_msdist_dirch_0_90 (mousedistande/direction changes)
%analyseml_13 %eff_p_msdist_dirch_90_180 (mousedistande/direction changes)

figure;
orient landscape;

global eff_q_contact_mscl_0 eff_q_contact_mscl_2 eff_q_contact_mscl_4 eff_q_contact_mscl_s
eff_q_contact_mscl_all
subplot(2,3,1)
plot(RSME.overall_0,eff_q_msdist_dirch_0_90_0,' k+',RSME.overall_2,eff_q_msdist_dirch_0_90_2,'
kx',RSME.overall_4,eff_q_msdist_dirch_0_90_4,' k*',RSME.overall_s,eff_q_msdist_dirch_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mental load');
XLabel('RSME');
YLabel('mouse efficiency : control actions/mouseclicks');
legend('0s','2s','4s','old');

global eff_q_msdist_dirch_0_90_0 eff_q_msdist_dirch_0_90_2 eff_q_msdist_dirch_0_90_4
eff_q_msdist_dirch_0_90_s eff_q_msdist_dirch_0_90_all
subplot(2,3,2)
```

```
plot(RSME.overall_0,eff_q_msdist_dirch_0_90_0,' k+',RSME.overall_2,eff_q_msdist_dirch_0_90_2,'
kx',RSME.overall_4,eff_q_msdist_dirch_0_90_4,' k*',RSME.overall_s,eff_q_msdist_dirch_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mental load');
XLabel('RSME');
YLabel('mouse efficiency : control actions/mouse direction changes (0-90)');
legend('0s','2s','4s','old');

global eff_q_msdist_dirch_90_180_0 eff_q_msdist_dirch_90_180_2 eff_q_msdist_dirch_90_180_4
eff_q_msdist_dirch_90_180_s eff_q_msdist_dirch_90_180_all
subplot(2,3,5)
plot(RSME.overall_0,eff_q_msdist_dirch_90_180_0,' k+',RSME.overall_2,eff_q_msdist_dirch_90_180_2,'
kx',RSME.overall_4,eff_q_msdist_dirch_90_180_4,' k*',RSME.overall_s,eff_q_msdist_dirch_90_180_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mental load');
XLabel('RSME');
YLabel('mouse efficiency : control actions/mouse direction changes (90-180)');
legend('0s','2s','4s','old');

global eff_p_msdist_dirch_0_90_0 eff_p_msdist_dirch_0_90_2 eff_p_msdist_dirch_0_90_4
eff_p_msdist_dirch_0_90_s eff_p_msdist_dirch_0_90_all
subplot(2,3,3)
plot(RSME.overall_0,eff_p_msdist_dirch_0_90_0,' k+',RSME.overall_2,eff_p_msdist_dirch_0_90_2,'
kx',RSME.overall_4,eff_p_msdist_dirch_0_90_4,' k*',RSME.overall_s,eff_p_msdist_dirch_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mental load');
XLabel('RSME');
YLabel('mouse efficiency : control actions*mouse direction changes (0-90)');
legend('0s','2s','4s','old');

global eff_p_msdist_dirch_90_180_0 eff_p_msdist_dirch_90_180_2 eff_p_msdist_dirch_90_180_4
eff_p_msdist_dirch_90_180_s eff_p_msdist_dirch_90_180_all
subplot(2,3,6)
plot(RSME.overall_0,eff_p_msdist_dirch_90_180_0,' k+',RSME.overall_2,eff_p_msdist_dirch_90_180_2,'
kx',RSME.overall_4,eff_p_msdist_dirch_90_180_4,' k*',RSME.overall_s,eff_p_msdist_dirch_90_180_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mental load');
XLabel('RSME');
YLabel('mouse efficiency : control actions*mouse direction changes (90-180)');
legend('0s','2s','4s','old');


figure;
orient landscape;

global eff_q_contact_mscl_0 eff_q_contact_mscl_2 eff_q_contact_mscl_4 eff_q_contact_mscl_s
eff_q_contact_mscl_all
subplot(2,3,1)
plot(sum_mean_0,eff_q_msdist_dirch_0_90_0,' k+',sum_mean_2,eff_q_msdist_dirch_0_90_2,'
kx',sum_mean_4,eff_q_msdist_dirch_0_90_4,' k*',sum_mean_s,eff_q_msdist_dirch_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mean error');
XLabel('mean error');
YLabel('mouse efficiency : control actions/mouseclicks');
legend('0s','2s','4s','old');

global eff_q_msdist_dirch_0_90_0 eff_q_msdist_dirch_0_90_2 eff_q_msdist_dirch_0_90_4
eff_q_msdist_dirch_0_90_s eff_q_msdist_dirch_0_90_all
subplot(2,3,2)
plot(sum_mean_0,eff_q_msdist_dirch_0_90_0,' k+',sum_mean_2,eff_q_msdist_dirch_0_90_2,'
kx',sum_mean_4,eff_q_msdist_dirch_0_90_4,' k*',sum_mean_s,eff_q_msdist_dirch_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mean error');
XLabel('mean error');
YLabel('mouse efficiency : control actions/mouse direction changes (0-90)');
legend('0s','2s','4s','old');

global eff_q_msdist_dirch_90_180_0 eff_q_msdist_dirch_90_180_2 eff_q_msdist_dirch_90_180_4
eff_q_msdist_dirch_90_180_s eff_q_msdist_dirch_90_180_all
subplot(2,3,5)
plot(sum_mean_0,eff_q_msdist_dirch_90_180_0,' k+',sum_mean_2,eff_q_msdist_dirch_90_180_2,'
kx',sum_mean_4,eff_q_msdist_dirch_90_180_4,' k*',sum_mean_s,eff_q_msdist_dirch_90_180_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mean error');
```

```
XLabel('mean error');
YLabel('mouse efficiency : control actions/mouse direction changes (90-180)');
legend('0s','2s','4s','old');

global eff_p_msdist_dirch_0_90_0 eff_p_msdist_dirch_0_90_2 eff_p_msdist_dirch_0_90_4
eff_p_msdist_dirch_0_90_s eff_p_msdist_dirch_0_90_all
subplot(2,3,3)
plot(sum_mean_0,eff_p_msdist_dirch_0_90_0,' k+',sum_mean_2,eff_p_msdist_dirch_0_90_2,'
kx',sum_mean_4,eff_p_msdist_dirch_0_90_4,' k*',sum_mean_s,eff_p_msdist_dirch_0_90_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mean error');
XLabel('mean error');
YLabel('mouse efficiency : control actions*mouse direction changes (0-90)');
legend('0s','2s','4s','old');

global eff_p_msdist_dirch_90_180_0 eff_p_msdist_dirch_90_180_2 eff_p_msdist_dirch_90_180_4
eff_p_msdist_dirch_90_180_s eff_p_msdist_dirch_90_180_all
subplot(2,3,6)
plot(sum_mean_0,eff_p_msdist_dirch_90_180_0,' k+',sum_mean_2,eff_p_msdist_dirch_90_180_2,'
kx',sum_mean_4,eff_p_msdist_dirch_90_180_4,' k*',sum_mean_s,eff_p_msdist_dirch_90_180_s,' k.')
%axis([200 300 200 300]);
Title('mouse behavior efficiency vs mean error');
XLabel('mean error');
YLabel('mouse efficiency : control actions*mouse direction changes (90-180)');
legend('0s','2s','4s','old');
```

%analyseml_10.m

```
global data eff_q_msdist_dirch_0_90_0 eff_q_msdist_dirch_0_90_2 eff_q_msdist_dirch_0_90_4
eff_q_msdist_dirch_0_90_s eff_q_msdist_dirch_0_90_all

y_max_coord=0.13;
y_min_coord=0;
y_title='efficiency: distance / dircection changes (0-90deg)';
%% 4 plotjes

eff_q_msdist_dirch_0_90_0=[];
for experiment=[1 5]
  for subject=1:7
    s_l=str2num([num2str(subject) num2str(experiment)]);
    eff_q_msdist_dirch_0_90_0=[ eff_q_msdist_dirch_0_90_0 data(s_l).eff_q_msdist_dirch_0_90 ];
  end %for subject
end %for experiment

eff_q_msdist_dirch_0_90_2=[];
for experiment=[ 2 6 ]
  for subject=1:7
    s_l=str2num([num2str(subject) num2str(experiment)]);
    eff_q_msdist_dirch_0_90_2=[ eff_q_msdist_dirch_0_90_2 data(s_l).eff_q_msdist_dirch_0_90 ];
  end %for subject
end %for experiment

eff_q_msdist_dirch_0_90_4=[];
for experiment=[ 3 7 ]
  for subject=1:7
    s_l=str2num([num2str(subject) num2str(experiment)]);
    eff_q_msdist_dirch_0_90_4=[ eff_q_msdist_dirch_0_90_4 data(s_l).eff_q_msdist_dirch_0_90 ];
  end %for subject
end %for experiment

eff_q_msdist_dirch_0_90_s=[];
for experiment=[4]
  for subject=1:7
    s_l=str2num([num2str(subject) num2str(experiment)]);
    eff_q_msdist_dirch_0_90_s=[ eff_q_msdist_dirch_0_90_s data(s_l).eff_q_msdist_dirch_0_90 ];
  end %for subject
end %for experiment

figure;
orient landscape
subplot(2,2,1);
plot ( subject_vect, eff_q_msdist_dirch_0_90_0, ' +k');
axis([0 8.5 y_min_coord y_max_coord]);
Title('0 sec trendinformation');
XLabel('subject number');
YLabel(y_title);
set(gca,'XTick',[ 1 2 3 4 5 6 7 ]);
boxplot_av(0.5,8,eff_q_msdist_dirch_0_90_0);

subplot(2,2,2);
plot(subject_vect, eff_q_msdist_dirch_0_90_2,' xk');
axis([0 8.5 y_min_coord y_max_coord]);
Title('2 sec trendinformation');
XLabel('subject number');
YLabel(y_title);
set(gca,'XTick',[ 1 2 3 4 5 6 7 ]);
boxplot_av(0.5,8,eff_q_msdist_dirch_0_90_2);

subplot(2,2,3);
plot(subject_vect, eff_q_msdist_dirch_0_90_4,' *k');
axis([0 8.5 y_min_coord y_max_coord]);
Title('4 sec trendinformation');
XLabel('subject number');
YLabel(y_title);
set(gca,'XTick',[ 1 2 3 4 5 6 7 ]);
boxplot_av(0.5,8,eff_q_msdist_dirch_0_90_4);

subplot(2,2,4)
plot(1:7, eff_q_msdist_dirch_0_90_s,' .k');
axis([0 8.5 y_min_coord y_max_coord]);
```

```
Title('old interface (Sally)');
XLabel('subject number');
YLabel(y_title);
set(gca,'XTick',[ 1 2 3 4 5 6 7 ]);
boxplot_av(0.5,8,eff_q_msdist_dirch_0_90_s);

% per proefpersoon

figure;
orient landscape;
schuiven=0.1*ones(1,14);
schuiven_s=0.1*ones(1,7);
plot ( subject_vect-2*schuiven, eff_q_msdist_dirch_0_90_0, ' +k',subject_vect-schuiven,
eff_q_msdist_dirch_0_90_2,' xk',subject_vect, eff_q_msdist_dirch_0_90_4,' *k'
,[1:7]+schuiven_s,eff_q_msdist_dirch_0_90_s,' .k');
axis([0 8 y_min_coord y_max_coord]);
Title('0,2, and 4 sec trendinformation');
XLabel('subject number ');
YLabel(y_title);
set(gca,'XTick',[ 1 2 3 4 5 6 7 ]);
legend('0s','2s','4s','old');
%% boxplot gaat fout voor proefpersoon 1 :((
for i=1:7
   boxplot_av(0.2,i+0.4,[eff_q_msdist_dirch_0_90_0([i i+7]) eff_q_msdist_dirch_0_90_2([i i+7])
eff_q_msdist_dirch_0_90_4([i i+7]) eff_q_msdist_dirch_0_90_s(i) ] );
end % for i

% per experiment
figure;
orient landscape;
% vect_1=[ 1:21  1:21];
% vect_2=[eff_q_msdist_dirch_0_90_0 eff_q_msdist_dirch_0_90_2 eff_q_msdist_dirch_0_90_4 ];
% hold on;
% plot(vect_1,vect_2, ' kx');
% plot(22:28,eff_q_msdist_dirch_0_90_s,' kx');
plot([1:7 1:7], eff_q_msdist_dirch_0_90_0,' k+',[9:15 9:15],eff_q_msdist_dirch_0_90_2,' kx',[17:23
17:23],eff_q_msdist_dirch_0_90_4,' k*',[25:31],eff_q_msdist_dirch_0_90_s,' .k');
axis([0 34 y_min_coord y_max_coord]);
Title('0, 2, and 4 sec trendinformation');
XLabel('subject number');
YLabel(y_title);
set(gca,'XTick',1:33);
%label=[];
% set(gca,'XTickLabel',{'1' ;'2' ; '3' ; '4';'5';'6';'7'; '1' ;'2' ; '3' ; '4';'5';'6';'7';'1' ;'2' ; '3' ; '4';'5';'6';'7' });
set(gca,'XTickLabel',{'1' ;'2' ; '3' ; '4';'5';'6';'7';'';'1' ;'2' ; '3' ; '4';'5';'6';'7';'';'1' ;'2' ; '3' ; '4';'5';'6';'7';'';'1' ;'2' ; '3' ;
'4';'5';'6';'7';'' ;'';''});
teksta='(0 sec)';
tekstb='(2 sec)';
tekstc='(4 sec)';
tekstd=('(old)');
legend('0s','2s','4s','old');
text(3,-0.05*y_max_coord,teksta);
text(10,-0.05*y_max_coord,tekstb);
text(20,-0.05*y_max_coord,tekstc);
text(28,-0.05*y_max_coord,tekstd);

% BOX-PLOTS  WEERGEVEN
boxplot_av(0.5,8,eff_q_msdist_dirch_0_90_0);
boxplot_av(0.5,16,eff_q_msdist_dirch_0_90_2);
boxplot_av(0.5,24,eff_q_msdist_dirch_0_90_4);
boxplot_av(0.5,32,eff_q_msdist_dirch_0_90_s);
eff_q_msdist_dirch_0_90_all=[eff_q_msdist_dirch_0_90_0 eff_q_msdist_dirch_0_90_2
eff_q_msdist_dirch_0_90_4 eff_q_msdist_dirch_0_90_s];
boxplot_av(0.5,33,eff_q_msdist_dirch_0_90_all);


% leer effect ??

figure;
orient landscape;
for subject=1:7
   subplot(2,4,subject);
   title(['subject ',num2str(subject)]);
```

```
    sequence=data(subject).sequence;
    vector_0_x=[];
    vector_0_y=[];
    vector_2_x=[];
    vector_2_y=[];
    vector_4_x=[];
    vector_4_y=[];
    vector_s_x=[];
    vector_s_y=[];

    for testje=1:7;
      experiment=sequence(testje);
      s_l=str2num([num2str(subject) num2str(experiment)]);
%     eff_q_msdist_dirch_0_90_s=[ eff_q_msdist_dirch_0_90_s data(s_l).eff_q_msdist_dirch_0_90 ];
      if experiment==1
        vector_0_x=[vector_0_x testje];
        vector_0_y=[vector_0_y data(s_l).eff_q_msdist_dirch_0_90];
      elseif experiment==2
        vector_2_x=[vector_2_x testje];
        vector_2_y=[vector_2_y data(s_l).eff_q_msdist_dirch_0_90];
      elseif experiment==3
        vector_4_x=[vector_4_x testje];
        vector_4_y=[vector_4_y data(s_l).eff_q_msdist_dirch_0_90];
      elseif experiment==4
        vector_s_x=[vector_s_x testje];
        vector_s_y=[vector_s_y data(s_l).eff_q_msdist_dirch_0_90];
      elseif experiment==5
        vector_0_x=[vector_0_x testje];
        vector_0_y=[vector_0_y data(s_l).eff_q_msdist_dirch_0_90];
      elseif experiment==6
        vector_2_x=[vector_2_x testje];
        vector_2_y=[vector_2_y data(s_l).eff_q_msdist_dirch_0_90];
      elseif experiment==7
        vector_4_x=[vector_4_x testje];
        vector_4_y=[vector_4_y data(s_l).eff_q_msdist_dirch_0_90];
      end %if
    end %for
    plot(vector_0_x,vector_0_y,' k+',vector_2_x,vector_2_y,' kx',vector_4_x,vector_4_y,'
k*',vector_s_x,vector_s_y,' k.')
    axis([0 9 y_min_coord y_max_coord]);
    Title({'subject',num2str(subject)});
    XLabel('test number');
    YLabel(y_title);
    set(gca,'XTick',[ 1 2 3 4 5 6 7 ]);
    set(gca,'XTickLabel',{'1' ;'2' ; '3' ; '4';'5';'6';'7';''});
    legend('0s','2s','4s','old');
    boxplot_av(0.5,8,[vector_0_y vector_2_y vector_4_y vector_s_y]);
end %for
```

System error or Mean absolute error (sum_mean)

Figure 1 for mean error:

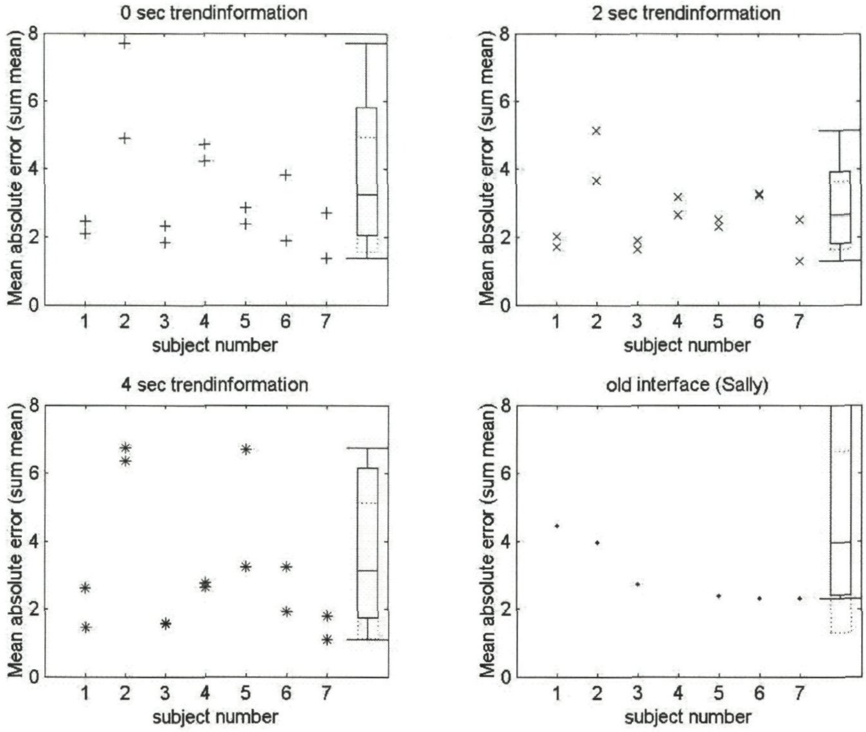


Figure 2 for mean error:

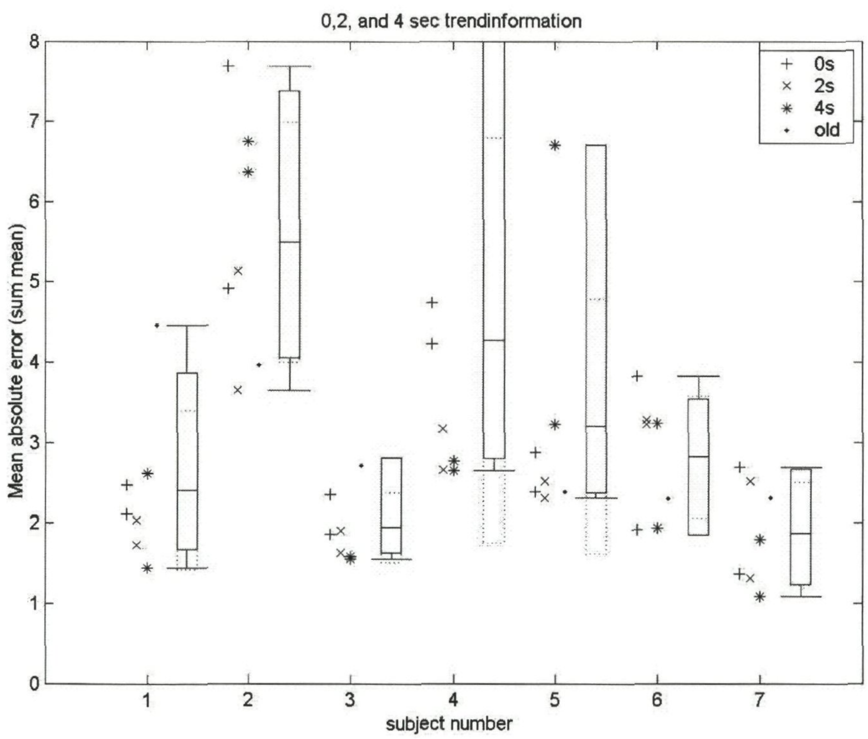


Figure 3 for mean error:

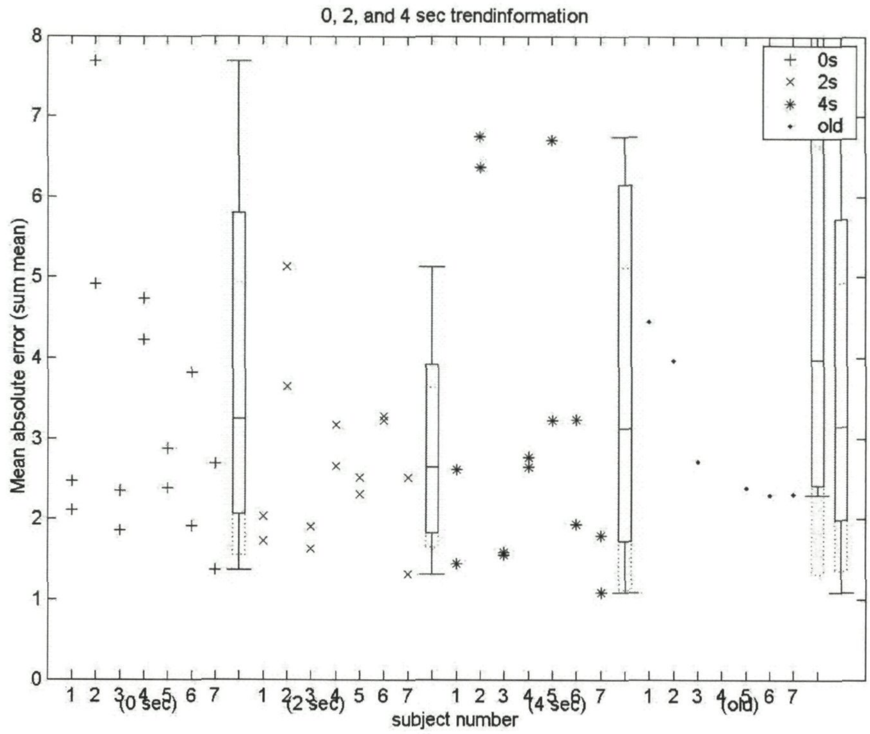
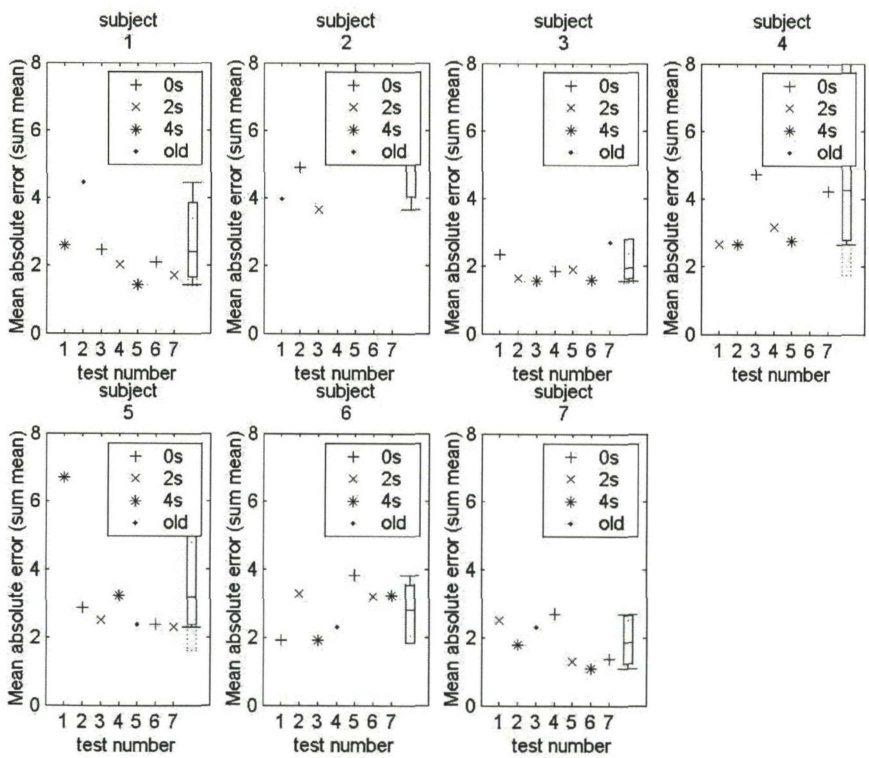


Figure 4 for mean error:



RSME (Rating Scale for Mental Effort)

Figure 1 for RSME:

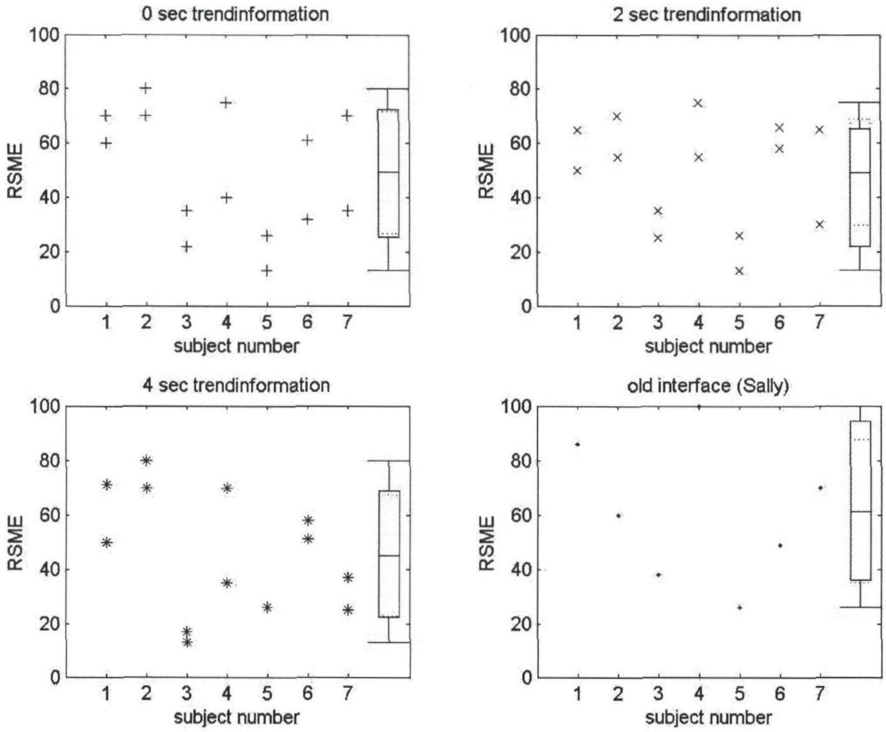


Figure 2 for RSME:

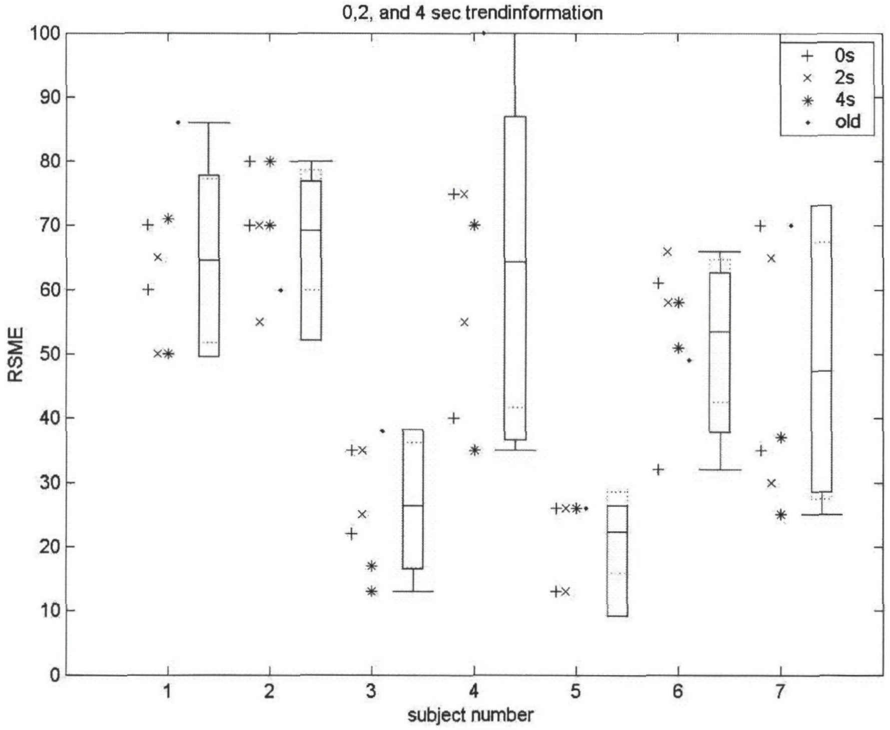


Figure 3 for RSME:

Appendix O: Figures for mean error, square error and RSME

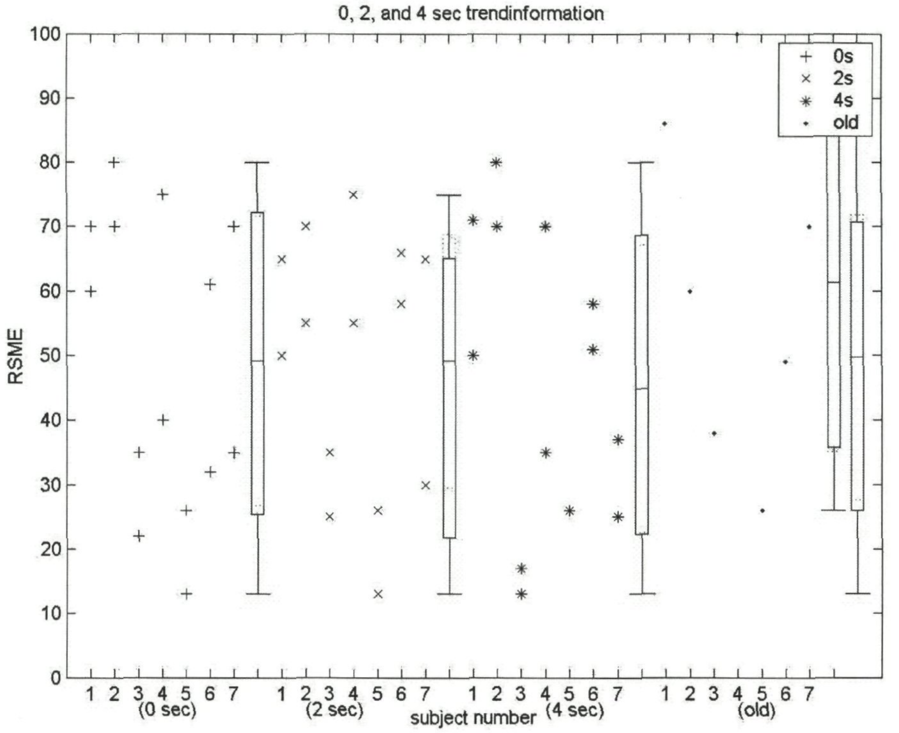
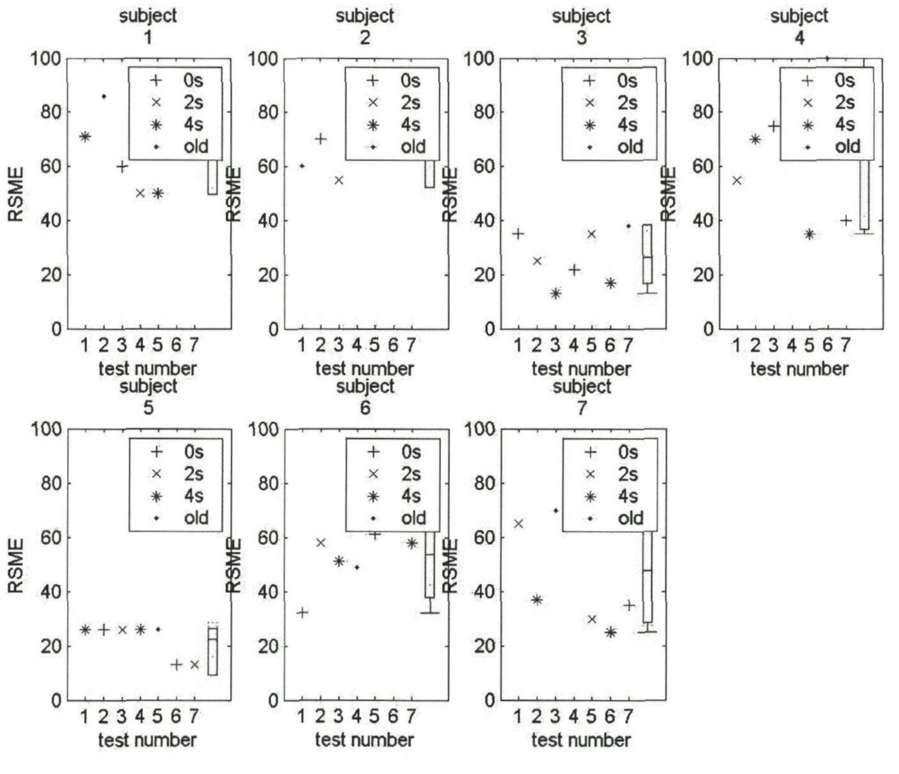


Figure 4 for RSME:



Square error

Figure 1 for square error:

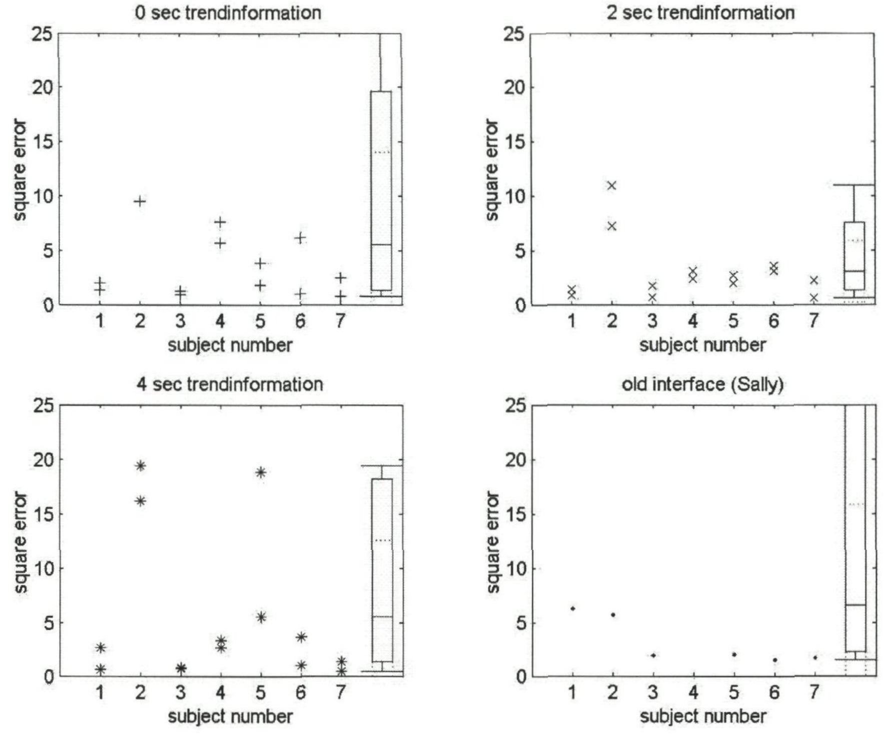


Figure 2 for square error:

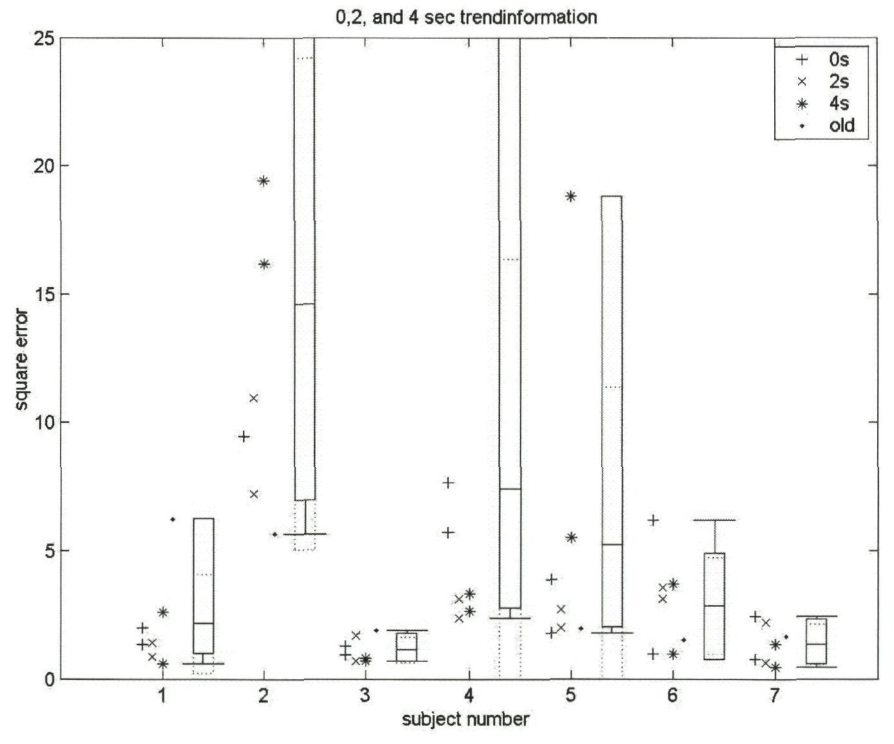


Figure 3 for square error:

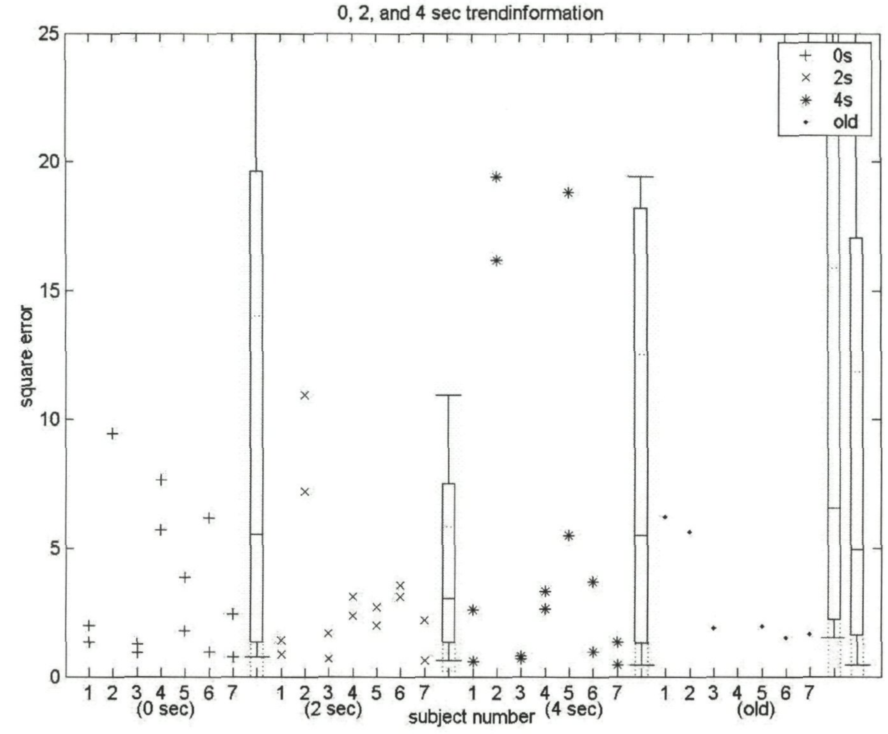


Figure 4 for square error:

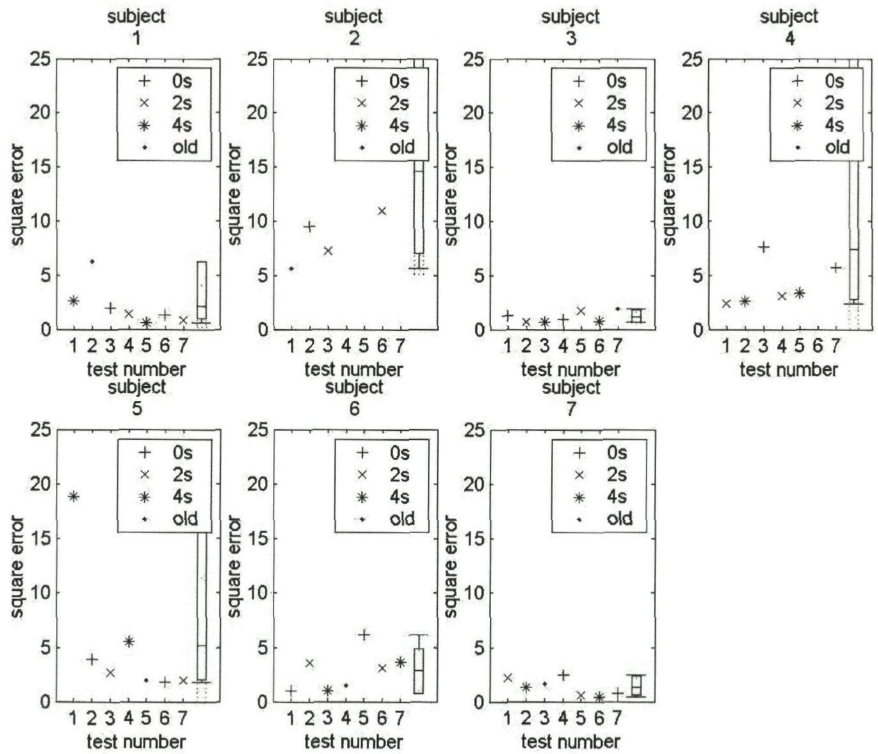


Figure A:Performance vs Workload

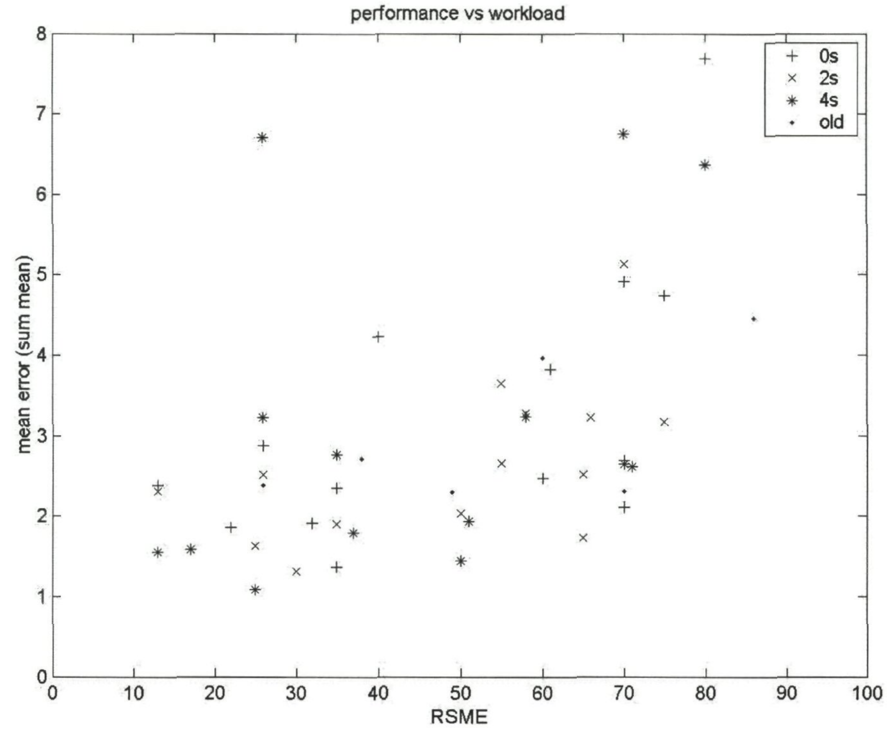


Figure B: Mouse behavior vs performance

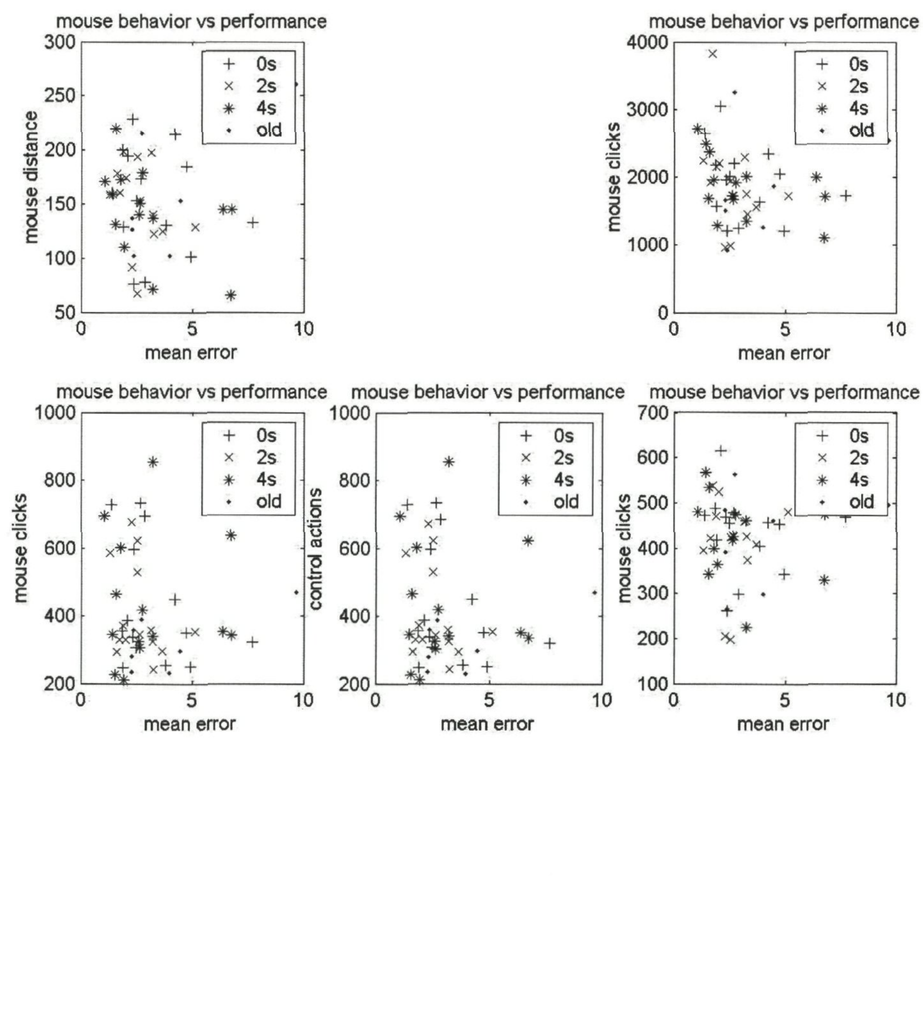
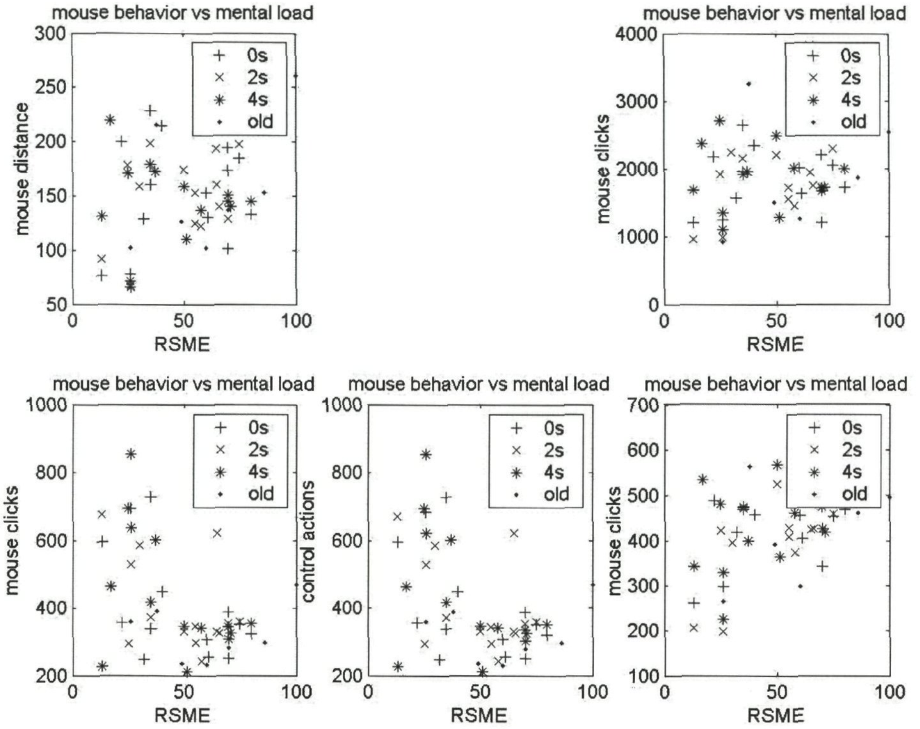


Figure C: Mouse behavior vs workload (or mental load) (RSME)



Appendix Q: Student t test [REF 17, PAGES 34,35,36,37,225,276,277]

Voor de steekproef geldt:

$$t = \frac{31,75 - 28,67}{2,85\ (1/12 + 1/12)^{\frac{1}{2}}} = \frac{3,08 \cdot \sqrt{6}}{2,85} = 2,65 \qquad (2.33)$$

Deze waarde ligt dus in het kritieke gebied. De nulhypothese dient verworpen te worden. Er is dus duidelijk verschil tussen de twee soorten voer.

Opmerking: In bovenstaand voorbeeld is zonder meer aangenomen, dat beide varianties gelijk zijn. Als hieraan getwijfeld wordt, dient men eerst de variantieverhoudingstoets (zie toets 2.18) uit te voeren om na te gaan of de populatievarianties significant verschillen.
In ons voorbeeld laat de variantieverhoudingstoets zien, dat het verschil in de twee varianties niet significant is.

2.10. De t-toets voor twee populatiegemiddelden.

(σ_A^2 en σ_B^2 onbekend en ongelijk)

Doel:
Onderzoek naar het significant zijn van het verschil tussen twee populatiegemiddelden (varianties σ_A^2 en σ_B^2 zijn onbekend en ongelijk).

Vorm:
Toetsing van de nulhypothese, dat het populatiegemiddelde μ_A een constante c (in de praktijk vaak gelijk 0) verschilt van het populatiegemiddelde μ_B.

Toepasbaarheid:
Deze toets is toepasbaar, als:
- de steekproeven geheel onafhankelijk van elkaar zijn;
- de populaties een normale verdeling volgen (of, als n_A en n_B maar groot genoeg zijn).

Methode:
We gebruiken dezelfde notatie als in voorgaande toets reeds gedefinieerd. Zonder bewijs, stellen wij hier dat de toetsingsgrootheid

$$t = \frac{(m_A - m_B) - (\mu_A - \mu_B)}{\left\{(s_A^2/n_A) + (s_B^2/n_B)\right\}^{\frac{1}{2}}} \qquad (2.34)$$

een t-verdeling van Student volgt.
Het "effectieve aantal vrijheidsgraden" is:

$$\nu_{eff} = \frac{\left\{\frac{S_A}{n_A(n_A-1)} + \frac{S_B}{n_B(n_B-1)}\right\}^2}{\frac{\left(\frac{S_A}{n_A(n_A-1)}\right)^2}{(n_A+1)} + \frac{\left(\frac{S_B}{n_B(n_B-1)}\right)^2}{(n_B+1)}} - 2 \qquad (2.35)$$

De waarde van ν_{eff} zal in het algemeen niet geheel zijn. De oplossing is òf om in de tabel voor de t-verdeling te gaan interpoleren, òf om ν_{eff} te gaan afronden op het dichtstbijzijnde gehele getal.
Opmerking: Een gemakkelijker toets is beschikbaar, indien de waarnemingen van de twee steekproeven paarsgewijze verkregen zijn (zie de volgende toets).

Voorschrift:
1. Formuleer de nulhypothese zorgvuldig.
2. Kies een geschikte onbetrouwbaarheid α.
3. De toetsingsgrootheid t staat hierboven gedefinieerd (zie 2.34).
4. Bereken s_A^2, s_B^2 en het aantal vrijheidsgraden ν_{eff} volgens (2.35).
5. Bepaal de kritieke waarde van t, bij gekozen α en deze ν_{eff} (tabel 2).
6. Bepaal voor de steekproef de experimentele waarde van t.
7. Indien deze waarde van t ligt in het kritieke gebied, moet de nulhypothese verworpen worden.

Voorbeeld |5|:
Twee onderzoekers hebben met verschillende methoden de volgende resultaten gevonden:

A:	B:
3128	1939
3219	1697
3244	3030
3073	2424
	2020
	2909
	1815
	2020
	2310
$n_A = 4$	$n_B = 9$
$m_A = 3166,0$	$m_B = 2240,4$
$s_A^2 = 6328,67$	$s_B^2 = 221661,3$

Stel men mag aannemen, dat de steekproeven stammen uit normaal verdeelde populaties.

Gevraagd wordt of het gemiddelde voor methode A het gemiddelde voor methode B overtreft.
De nulhypothese luidt dat beide methoden gemiddeld hetzelfde resultaat leveren. De onbetrouwbaarheid bij de toetsing zij gegeven door $\alpha = 0,05$.
De toetsing wordt eenzijdig verricht.
We berekenen eerst de varianties

$$s^2(\bar{m}_A) = s_A^2/n_A \qquad (2.36)$$

en

$$s^2(\bar{m}_B) = s_B^2/n_B \qquad (2.37)$$

$$s_A^2/n_A = 6328,67/4 = 1582,17 \qquad (2.38)$$

$$s_B^2/n_B = 221661,3/9 = 24629,03 \qquad (2.39)$$

Het effectief aantal vrijheidsgraden is:

$$\nu_{eff} = \frac{\left(s^2(\bar{m}_A) + s^2(\bar{m}_B)\right)^2}{\left\{s^4(\bar{m}_A)/(n_A+1) + s^4(\bar{m}_B)/(n_B+1)\right\}} - 2 \qquad (2.40)$$

Dit levert:

$$\nu_{eff} = \frac{(26211,20)^2}{500652,4 + 60658911,9} - 2 \qquad (2.41)$$

$$= 11,233 - 2 = 9,23$$

$$= 9.\ \text{(afgerond)} \qquad (2.42)$$

De toetsingsgrootheid is:

$$t = \frac{(m_A - m_B) - (\mu_A - \mu_B)}{(s_A^2/n_A + s_B^2/n_B)^{\frac{1}{2}}} \qquad (2.34)$$

De kritieke waarde bij $\alpha = 0,05$ en $\nu = 9$, is 1,83 (zie tabel 2).
Voor de steekproef geldt:

$$t = \frac{3166,0 - 2240,4}{(1582,17 + 24629.03)^{\frac{1}{2}}} = \frac{925,6}{(26211,20)^{\frac{1}{2}}} = \frac{925,6}{161,90} = 5,72 \qquad (2.43)$$

Deze waarde ligt duidelijk in het kritieke gebied. Methode A heeft dus een gemiddelde dat het gemiddelde van methode B overtreft.

Opmerking: In dit voorbeeld mocht aangenomen worden, dat de steekproeven afkomstig waren uit normaal verdeelde populaties. Indien men hieraan twijfelt, kan men bijvoorbeeld de normaliteitstoets 2.31 uitvoeren.

Voor ons voorbeeld levert die toets als resultaat, dat met een betrouwbaarheid van 95% beide steekproeven geacht kunnen worden te stammen uit normaal verdeelde populaties.

2.11. De t-toets voor twee populatiegemiddelden

(paarsgewijze waarnemingen; σ_A^2 en σ_B^2 onbekend en ongelijk)

Doel:
Onderzoek naar het significant zijn van het verschil tussen twee populatiegemiddelden (σ_A^2 en σ_B^2 onbekend, eventueel ongelijk).

Vorm:
Toetsing van de nulhypothese dat het populatiegemiddelde μ_A een constante (in de praktijk vaak gelijk 0) verschilt van het populatiegemiddelde μ_B.

Toepasbaarheid:
Deze t-toets is toepasbaar, als:
- de twee populaties normaal verdeeld zijn;
- de waarnemingen bij het nemen van de twee steekproeven steeds paarsgewijze verricht worden; d.w.z. als de tot een paar samengevoegde waarnemingen onder dezelfde (of althans vergelijkbare) omstandigheden tot stand gekomen zijn;
- elk verschil in waarnemingen van elk paar onafhankelijk is van de andere verschillen.

Deze toets kan gebruikt worden als een toets voor de aanwezigheid van een constant verschil (een systematische afwijking; een onzuiverheid) tussen twee populatiegemiddelden, indien een van beide steekproeven betrekking heeft op waarnemingen verkregen met een standaard methode, of verkregen met een reeks standaarden.

Methode:
De waarnemingen x_i en y_i komen nu niet meer onafhankelijk van elkaar tot stand. De waarnemingen worden weloverwogen, onder gelijke omstandigheden en paarsgewijze verricht.
Het verschil $d_i = x_i - y_i$ van een waarnemingspaar wordt veel minder door uitwendige omstandigheden beïnvloed. We nemen aan dat de verschillen d_i

Appendix Q: Student t test [REF 17, PAGES 34,35,36,37,225,276,277]

Tabel 2: Kritieke waarden voor de t-verdeling van Student |1|

De t-verdeling is symmetrisch t.o.v. t=o; ν is het aantal vrijheidsgraden.

onbetrouwbaarheid α							
tweezijdige toetsing	0,20	0,10	0,05	0,02	0,01	0,002	0,001
eenzijdige toetsing	0,10	0,05	0,025	0,01	0,005	0,001	0,0005
ν							
1	3,08	6,31	12,7	31,8	63,7	318	637
2	1,89	2.92	4,30	6,97	9,93	22,3	31,6
3	1,64	2,35	3,18	4,54	5,84	10,2	12,9
4	1,53	2,13	2,78	3,75	4,60	7,17	8,61
5	1,48	2,02	2,57	3,37	4,03	5,89	6,87
6	1,44	1,94	2,45	3,14	3,71	5,21	5,96
7	1,42	1,90	2,37	3,00	3,50	4,79	5,41
8	1,40	1,86	2,31	2,90	3,36	4,50	5,04
9	1,38	1,83	2,26	2,82	3,25	4,30	4,78
10	1,37	1,81	2,23	2,76	3,17	4,14	4,59
11	1,36	1,80	2,20	2,72	3,11	4,03	4,44
12	1,36	1,78	2,18	2,68	3,06	3,93	4,32
13	1,35	1,77	2,16	2,65	3,01	3,85	4,22
14	1,36	1,76	2,15	2,62	2,98	3,79	4,14
15	1,34	1,75	2,13	2,60	2,95	3,73	4,07
16	1,34	1,75	2,12	2,58	2,92	3,69	4,02
17	1,33	1,74	2,11	2,57	2,90	3,65	3,97
18	1,33	1,73	2,10	2,55	2,88	3,61	3,92
19	1,33	1,73	2,09	2,54	2,86	3,58	3,88
20	1,33	1,73	2,09	2,53	2,85	3,55	3,85
21	1,32	1,72	2,08	2,52	2,83	3,53	3,82
22	1,32	1,72	2,07	2,51	2,82	3,51	3,79
23	1,32	1,71	2,07	2,50	2,81	3,49	3,77
24	1,32	1,71	2,06	2,49	2,80	3,47	3,75
25	1,32	1,71	2,06	2,49	2,79	3,45	3,73
26	1,32	1,71	2,06	2,48	2,78	3,44	3,71
27	1,31	1,70	2,05	2,47	2,77	3,42	3,69
28	1,31	1,70	2,05	2,47	2,76	3,41	3,67
29	1,31	1,70	2,05	2,46	2,76	3,40	3,66
30	1,31	1,70	2,04	2,46	2,75	3,39	3,65
35	1,31	1,69	2,03	2,44	2,72	3,34	3,59
40	1,30	1,68	2,02	2,42	2,70	3,31	3,55
60	1,30	1,67	2,00	2,39	2,66	3,23	3,46
120	1,29	1,66	1,98	2,36	2,62	3,16	3,37
200	1,29	1,65	1,97	2,35	2,60	3,13	3,34
∞	1,28	1,65	1,96	2,33	2,58	3,09	3,29

8. Lijst van symbolen

a,b,c	willekeurige parameters
C	toetsingsgrootheid voor de toets van Cochran
D	discriminante functie
f(x)	kansverdeling of frequentieverdeling voor grootheid x
F(x)	cumulatieve kansverdeling of frequentieverdeling voor grootheid x
f^o	waargenomen frequentieverdeling
f^e	theoretisch te verwachten frequentieverdeling
g	statistisch gewicht van een waarneming
F	toetsingsgrootheid behorend bij de F-verdeling
H	toetsingsgrootheid voor de rangensomtoets
k	aantal meetseries; aantal klasse-intervallen; toetsingsgrootheid in de tekentoets
K	keuringskwaliteit
m	gemiddelde waarde voor een steekproef; variabele steekproefgrootte (bij sequente toetsing)
M	moment van een verdeling
n	aantal waarnemingen in een steekproef
N	totaal aantal waarnemingen in alle steekproeven tezamen; aantal geregistreerde pulsen
P(x)	kans op waarde x
Q	toetsingsgrootheid in puntentoets van Wilcoxon
R	toetsingsgrootheid in de rangensomtoets; teltempo
r	correlatiecoëfficiënt
s	standaarddeviatie van een steekproef
s_p	standaarddeviatie van de somverzameling van alle steekproeven
s^2	steekproefvariantie
s^2_{ext}	externe variantie ("tussen de steekproeven")
s^2_{int}	interne variantie ("binnen de steekproeven")
S	som der kwadraten der afwijkingen; puntentotaal
S_{ext}	externe som der kwadraten ("tussen de steekproeven")
S_{int}	interne som der kwadraten ("binnen de steekproeven")
$S_n(X)$	genormeerde verdelingsfunctie voor een steekproef
t	toetsingsgrootheid behorende bij de t-verdeling

276

T	toetsingsgrootheid in de tekenrangtoets
T^2	toetsingsgrootheid in de toets van Hotelling
u	toetsingsgrootheid behorende bij de normale verdeling; gemiddelde van de poissonverdeling
U	toetsingsgrootheid in de inversietoets van Wilcoxon
x_i	i-de waarneming in een steekproef
y_i	i-de waarneming van een tweede variabele in een steekproef
x_{ij}	i-de waarneming in de j-de steekproef
$\bar{x}$	gemiddelde waarde van de x_i
$\bar{y}$	gemiddelde waarde van de y_i
W	concordantiegraad
α	kans op fout van 1e soort; onbetrouwbaarheid
β	kans op fout van 2e soort; onwerkzaamheid
Δ	toetsing bij de toetsen van Kolmogorov en Smirnov
μ	gemiddelde waarde voor een populatie
$\tilde{\mu}$	mediaanwaarde voor een populatie
$\mu_{y.x}$	populatiegemiddelde van y bij gegeven x
ν	aantal vrijheidsgraden
ρ	correlatiecoëfficiënt voor de populatie
σ	standaarddeviatie voor de populatie
σ^2	populatievariantie
$\sigma^2_{y.x}$	populatievariantie
χ^2	toetsingsgrootheid behorende bij χ^2-verdeling

277