



BEN-GURION UNIVERSITY OF THE NEGEV  
FACULTY OF ENGINEERING SCIENCES  
DEPARTMENT OF INDUSTRIAL ENGINEERING AND MANAGEMENT

**DYNAMIC SWITCHING BETWEEN  
COLLABORATION LEVELS OF A HUMAN-  
ROBOT SYSTEM IN TARGET RECOGNITION  
TASKS**

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE M.Sc. DEGREE

Tkach Itshak

October 2008



BEN-GURION UNIVERSITY OF THE NEGEV  
FACULTY OF ENGINEERING SCIENCES  
DEPARTMENT OF INDUSTRIAL ENGINEERING AND MANAGEMENT

**DYNAMIC SWITCHING BETWEEN  
COLLABORATION LEVELS OF A HUMAN-  
ROBOT SYSTEM IN TARGET RECOGNITION  
TASKS**

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE M.Sc. DEGREE

By: Tkach Itshak

Supervised by: Prof. Edan Yael, Dr. Bechar Avital

Author: Tkach Itshak

Date:.....

Supervisor: Prof. Edan Yael

Date:.....

Supervisor: Dr. Bechar Avital

Date:.....

Chairman of Graduate Studies Committee: Prof. Schechtman Edna

Date:.....

October 2008

This work was carried out under the supervision of

Prof. Edan Yael and Dr. Bechar Avital

In the Department of Industrial Engineering and Management

Faculty of Engineering Sciences

## *Acknowledgments*

When taking a glance back, the outcome of this research has strongly been influenced by many people who accompanied me during the whole or part of the work in my private and academic life. Therefore, it is also the occasion to very warmly thank these people and to acknowledge their support.

In the first place I thank my parents with all my heart. They have given me continuous support during all the years, have guided me through the early years of my life, and always have provided a stable environment. Also, I want to warmly thank my brother, David, for his constant encouragement and for buying the computer that assisted me in performing this research.

I wish to express my deepest appreciation to my advisors, Prof. Yael Edan and Dr. Avital Bechar for their help, guidance, great assistance, and for the opportunity to conduct this research. I am also very grateful for their support during this work, for providing the necessary scientific freedom to carry out this research, and for their patience during this time.

# Table of Contents

<b>List of Figures.....</b>	<b>III</b>
<b>List of Tables .....</b>	<b>IV</b>
<b>List of Symbols .....</b>	<b>V</b>
<b>Abstract.....</b>	<b>VII</b>
<b>1 Introduction .....</b>	<b>1</b>
1.1 Problem description .....	1
1.2 Research significance and contributions .....	2
1.3 Thesis structure .....	3
<b>2 Scientific and technological background.....</b>	<b>4</b>
2.1 Automation and autonomous Robots .....	4
2.2 Teleoperation and Human – Robot collaboration .....	6
2.3 Signal Detection Theory .....	14
2.4 Filtering and Control.....	18
2.5 Previous work on collaboration levels in target recognition tasks .....	19
<b>3 Methodology.....</b>	<b>27</b>
3.1 Problem objective .....	27
3.2 Outline.....	27
3.3 Human-robot system definition .....	27
3.4 Controller design.....	29
3.5 Dynamic switching methodology .....	30
3.6 Assumptions:.....	31
3.7 Switching objective function .....	31
3.8 Switching algorithms .....	32
3.9 Numerical Analysis.....	32
<b>4 Switching objective function.....</b>	<b>33</b>
4.1 Switching objective function formulation.....	33
<b>5 Switching algorithms.....</b>	<b>37</b>
5.1 Basic switching algorithm for immediate response time of the system.....	37
5.2 Switching algorithms formulation .....	37
5.3 “RLSA” algorithm .....	40
5.4 “CTSA” algorithm .....	40
5.5 “CFSA” algorithm .....	41

5.6	“PRSA” algorithm .....	41
5.7	Parameters classification.....	45
<b>6</b>	<b>Numerical analysis.....</b>	<b>46</b>
6.1	Numerical simulation of switching algorithms .....	46
6.2	Analysis of the switching points .....	52
<b>7</b>	<b>Conclusions and future research.....</b>	<b>53</b>
<b>8</b>	<b>Bibliography.....</b>	<b>56</b>
<b>9</b>	<b>Appendixes .....</b>	<b>59</b>
	Appendix I: The simulation program .....	60
	Appendix II: Additional numerical analysis results .....	61
	Appendix III: Dynamic switching pseudocodes .....	90
	Appendix IV: Matlab programs .....	95

## List of Figures

Figure 1: Three paradigms for organizing intelligence in robots: (a) Hierarchical, (b) Reactive, and (c) Hybrid deliberative/reactive (Murphy, 2000). .....	6
Figure 2: Organizing chart of a telesystem (Murphy 2000).....	7
Figure 3: Types of function performed by man or machine components of man-machine systems (McCormick, 1993).....	9
Figure 4: Schematic illustration of manual, semiautomatic, and automatic man-machine systems. (McCormick, 1993).....	11
Figure 5: Internal response probability density functions for noise alone and for signal plus noise trials. ....	14
Figure 6: Example of a criterion line that divides the density functions into four sections .....	15
Figure 7: Modified SDT model for the robot.....	16
Figure 8: Modified SDT model for both human and robot.....	16
Figure 9: Objective function score for different human and robot sensitivities of the four collaboration levels for $P_s=0.2$ . H – blue, HR – cyan, HOR yellow and R – red. ....	25
Figure 10: Best collaboration level map for different human and robot sensitivities. The colors represent different collaboration levels: HR – cyan, HOR - yellow and R – red.....	25
Figure 11: Block diagram of a closed loop control method.....	29
Figure 12: System’s block diagram. ....	29
Figure 13: System's block diagram with collaboration switching priorities. In the upper part presented taxonomy of the switching layers and below illustration of blocks of collaboration level switch for each layer. ....	31
Figure 14: Basic switching algorithm for immediate response time of the system .....	37
Figure 15: Block diagram of the control system algorithms.....	39
Figure 16: Block diagram of “RLSA” algorithm.....	43
Figure 17: Block diagram of “CTSA” algorithm.....	43
Figure 18: Block diagram of “CFSA” algorithm .....	44
Figure 19: Block diagram of “PRSA” algorithm .....	44
Figure 20: The cost for non optimal work with switching in normally distributed random values of $P_s$ distribution – algorithm “RLSA” .....	47
Figure 21: The cost for non optimal work with switching in normally distributed random values of $P_s$ distribution – algorithm “CTSA” .....	47
Figure 22: The cost for non optimal work with switching in normally distributed random values of $P_s$ distribution – algorithm “CFSA” .....	48
Figure 23: The cost for non optimal work with switching in normally distributed random values of $P_s$ distribution – algorithm “PRSA” .....	48
Figure 24: Graphical presentation of number of switches vs. switching gain in random distributed $P_s$ . ....	52

## List of Tables

Table 1: Robot primitives defined in terms of inputs and outputs. ....	5
Table 2: Levels of automation of decision and action selection (Sheridan, 1992). ....	13
Table 3: Classification of detections. ....	15
Table 4: Performance parameters ....	28
Table 5: Switching decisions. ....	31
Table 6: Summary of the terms.....	42
Table 7: Summary of the results for all distributions with 200 independent simulations with different starting conditions. ....	51



## List of Symbols

$\beta$	likelihood ratio, cutoff ratio
$\mu$	mean
$\sigma$	standard deviation
$\tilde{\sigma}$	sum of switching objective functions at switching points - gross switching gain
$\bar{\sigma}$	gain in global objective function score achieved by each system's operation (image sampling) compared to a particular collaboration level that used as a reference
$\zeta_k$	image sample indexed k
$\omega_k$	switch point indexed k
$\tau$	time between switches
$\psi$	the nominal time between switching operations
$\Psi$	the deviation value from switching frequency
B	threshold for switching operation
CCL	current collaboration level
CR index	correct rejection
d'	sensitivity
f	cumulative gain of the entire operation of the system – net switching gain
F	density function
FA index	false alarm
$F_{CRs}$	the correct rejection density function for the system
$F_{FAs}$	the number of system false alarm objects
F(s)	fitness function of the model
H	manual collaboration level
H index	hit
h index	human
HOR	collaboration level where the human supervise the robot
HR	collaboration level where the robot recommends the human
m	total number of image samples
n	total number of switches
N	the number of objects
N index	noise
OCL	optimal collaboration level
$p_{FAh}$	the human false alarm probability
$p_{FAr}$	the robot false alarm probability
$p_{FArh}$	the human probability of not correcting the robot false alarm
$p_{Hh}$	the human probability of detecting a target which the robot did not detect
$p_{Hr}$	the robot probability of a hit
$p_{Hrh}$	the human probability of confirming a robot hit
$p_{Hs}$	the system probability for a hit
$p_{Ms}$	the probability of a system misses
$P_s$	target probability
R	fully autonomous collaboration level
r index	robot
rh index	human operation on robot operations
ROC	receiver operating characteristic
s	possible solution for F
S	solution space for F
S index	signal
SDT	signal detection theory

$t_{CRh}$	the HO correct rejection time
$t_{CRrh}$	the human time to correctly reject a robot false alarm
$t_D$	decision time
$t_{FAh}$	the human false alarm time
$t_{FArh}$	the human time needed to correct a robot false alarm
$t_{Hh}$	the human time required to hit a target which the robot did not hit
$t_{Hrh}$	the human time required to confirm a robot hit
$t_{h\_response}$	time for the human to identify and adapt to switching
$t_{h\_1}$	time for the human to adapt to switching to a nearest collaboration level
$t_{h\_2}$	time for the human to adapt to jumping for two collaboration levels
$t_{h\_3}$	time for the human to adapt to jumping for three collaboration levels
$t_M$	motoric time
$t_{Mh}$	the human time invested when missing a target which the robot did not hit
$t_{Mrh}$	the human time lost when a robot hit is missed
$t_r$	the robot time
$t_{response}$	system's response time
$t_{r\_response}$	time for the robot to switch between collaboration levels
$t_s$	the system time that is required to perform the task
$V_{AR}$	payoff ratio $V_{AR} = -\frac{V_{FA}}{V_H}$
$V_C$	the cost of one object recognition operation
$V_{CR}$	the gain from a single correct rejection
$V_{CRs}$	the system gain for correct rejection
$V_{FA}$	the damage from a single false alarm
$V_{FAs}$	the system penalty for false alarms
$V_H$	the gain from a single hit
$V_{Hs}$	the system gain for target detection (hit)
$V_{Is}$	the system objective function
$V_{Iswitch}$	the switching objective function
$V_{I\text{Optimal}}$	objective function score in optimal collaboration level
$V_{I\text{Current}}$	objective function score in current collaboration level
$V_{I\text{reference}}$	objective function score of particular collaboration level that used as a reference
$V_M$	the penalty of a single miss
$V_{Ms}$	the system penalty for missing the target (miss)
$V_p$	penalty for switching earlier than the nominal value of switching frequency
$V_t$	the cost of one time unit
$V_{Ts}$	the system operation cost
$X$	the measurement unit $X$ of object features
$x$	a position along coordinate $X$ , represents the cutoff point
$Z$	the distance in standard deviation units
*	indicates optimal value

# *Abstract*

This work presents development, analysis and evaluation of algorithms for dynamic switching between collaboration levels in a human-robot target recognition system to maintain maximum system performance despite deviations in the performances of both human and robot during task performance. This research is based on a system objective function that was designed to enable determination of the expected value of task performance, given the parameters of the system, the task, and the environment using signal detection theory (Bechar et al., 2006). The collaboration levels were based on ten degrees of autonomy from Sheridan's (Sheridan, 1992) scale of "action selection and automation of decision".

Four different algorithms for dynamic switching were developed and tested using numerical analysis for different scenarios and different distributions of the parameters. The algorithms were designed to switch the system to the best collaboration level by calculating the best objective function score with the current human, robot, environment and task parameters. The design of these algorithms considers the limitation of the switching execution frequency, the number of levels to be shifted at one time, the system's response time and the changes of the parameters due to this response time. The four algorithms differ by the conditions for making the switch between collaboration levels. Algorithm "RLSA" - regular switch algorithm, makes a switch whenever the gain of the switching is positive. Algorithm "CTSA" - constrained switch algorithm, makes switches whenever the gain of the switching is higher than a desired threshold. Algorithm "CFSA" - constrained without frequency limitation switch algorithm, also makes switches whenever the gain of the switching is higher than a desired threshold like in "CTSA" algorithm but with no limitation of the switching execution frequency. Algorithm "PRSA" - predictive switch algorithm, makes switches by including predictions from past data to operation of "CFSA" algorithm. It switches the system to the optimal collaboration level only when this collaboration level was optimal in most image samples analyzed by the system.

System performance was analyzed in simulations for a variety of target probabilities distributions. Improvements that can be achieved by each algorithm were calculated as a mean value for 200 independent simulation 'runs' for each target probability distribution. Results indicated that when there is a limitation of the switching execution frequency, the switching does not always improve results but sometimes decreases the overall system performance. Thus, introducing dynamic switching mechanism may actually lower performance if the system uses non-optimal methods for the switching operation (like manual or stochastic switching). Therefore, well designed algorithms are necessary to make dynamic switching in human-robot system effective. The detailed design of such algorithms is presented in this work. The numerical analysis results indicated that the developed algorithms for dynamic switching achieved improved system performance.

## **Keywords:**

human-robot interaction, dynamic switching, collaboration levels, target recognition.

# 1 Introduction

## 1.1 Problem description

In recent years intelligent machines, especially computers, can perform many functions that at one time could only be done by humans. Machine execution of such functions or automation has also been extended to functions that humans do not wish to perform, or cannot perform as accurately or reliably as machines (Parasuraman et al., 2000). This led to the introduction of automation into virtually all aspects of human-machine systems.

Today, robotic systems perform well only when all conditions are known and well defined. Autonomous robots do not perform well in real-world environments which are dynamic and unpredictable (Al-Jumaily and Amin, 2000; Fletcher et al., 2005). They cannot cope with unexpected situations encountered in unstructured, changing “real-world” conditions (Bechar and Edan, 2003). Inadequacies of sensor technologies further impair the capabilities of autonomous robotics (Everett and Dubey, 1998). Complexity is further increased when dealing with natural objects such as in medical and agricultural environments, due to the object’s high degree of variability in shape, texture, color, size, orientation and position. Consequently, the robotic systems become increasingly cumbersome, thereby creating a complicated system which is expensive to develop and operate (Bechar, 2006) and not robust enough.

Target recognition is an essential part of most robotic systems (Bicho et al., 2000; Ye and Tsotsos, 1999). However, target recognition in unstructured environments is characterized by low detection rates and high false alarm rates (Bechar et al., 2006).

Humans excel in recognition capabilities (Ayanna, 2006) and can easily adapt to changing environmental and objective conditions (Pook and Ballard, 1996). Human perception, acting and thinking capabilities in dynamic environments are unmatched to those of robots. However, a human operator is inconsistent, tends to fatigue and suffer from distractions (Swets et al., 2000), and ultimately might reduce the system’s production rate relative to that of a fully autonomous system in a structured environment (Bechar et al., 2006).

By combining the advantages of human perception and recognition skills with the autonomous robots’ accuracy and consistency a cooperative human-robotic system can

increase target identification rate and reduce the complexity of the robotic system (Parasuraman et al., 2000), and handle unpredictable conditions that autonomous systems are incompetent to deal with (Pook and Ballard, 1996). Different researches focus on different applications of human-robot collaboration. Sheridan (1992) divides automation into ten levels, from fully autonomous, with no human intervention to fully manual. Scholtz (2002) describes five roles that a human may take when interacting with a robot: supervisor, operator, mechanic, peer and bystander. Ayanna (2006) focused on role allocation in Human-Robot collaboration for space missions. Bechar et al. (2006) defined four human-robot collaboration levels for target recognition tasks in unstructured environments (Bechar et al., 2006). An objective function was developed to determine the expected value of task performance given the parameters of the system, the task, and the environment (Bechar et al., 2006).

## **1.2 Research significance and contributions**

Combined human-robot target recognition system performance depends on environmental conditions (e.g., illumination, visibility, terrain type), human conditions (fatigue, stress, workload), and system parameters (error, accuracy, reliability, Bechar et al., 2006). Thus, the system performance might vary rapidly during task performance. This brings a strong interest in making the human-robot system dynamic and flexible, allowing it to make switches between different collaboration levels in order to maintain adequate results.

This research introduces a dynamic control mechanism for an uncontrolled human-robot system that is responsible for switching between different collaboration levels for target recognition tasks in unstructured environments in form of a logical controller. The control methodology considers the most important limitations of a combined human-robot system, processing time and bandwidth. This methodology allows to maintain high system performance despite possible deviations in the parameter values during task performance.

Several algorithms were developed for the logical controller to allow real-time dynamic switching of the collaboration levels. The algorithms were implemented in a closed-loop control process and their performances were simulated numerically for different parameters and conditions.

These developments enable smooth real-time adaptation of the combined human-robot system to many possible changes of the conditions and parameters during system's task performance, like changes in the environment, human operator time lags and robot performance. It was shown that the overall system performance was increased in case of

rapid changes of the environment and the system and can be implemented to many combined human-robot systems to increase performance.

### **1.3 Thesis structure**

The thesis is organized as follows: chapter 2 presents the scientific and technological background on automation and autonomous robots, teleoperation and human-robot collaboration, filtering and control, signal detection theory and review of previous work on collaboration levels in target recognition tasks. Chapter 3 presents the methodology which starts with the description of the problem of dynamic switching of human-robot system for target recognition tasks in unstructured environments, continues with the outline of the methods, definitions of major terms, the research assumptions, a presentation of the controller design, a brief presentation of switching objective function and of the proposed algorithms and the numerical computations conducted. The formulation of the switching objective function which includes mathematical expression of the switching objective function is presented in chapter 4. The development of four switching algorithms for the controlled dynamic switching operation is presented in chapter 5. Chapter 6 presents numerical analysis of switching algorithms and their operation. Conclusions and discussion of future research is discussed in chapter 7.

## 2 Scientific and technological background

The scientific and technological background includes a review of automation and autonomous robots, teleoperation and human-robot collaboration, filtering and control theories. Signal detection theory and previous research methods that served as the basis for the development of the thesis methodologies are also reviewed.

### 2.1 Automation and autonomous Robots

Machines, especially computers, are now capable of performing many functions that at one time could only be performed by humans. Machine execution of such functions or automation has also been extended to functions that humans do not wish to perform, or cannot perform as accurately or reliably as machines (Parasuraman et al., 2000). This led to the introduction of automation into virtually all aspects of human-machine systems.

The Oxford English Dictionary (1989) defines automation as:

- 1) Automatic control of the manufacture of a product through a number of successive stages;
- 2) The application of automatic control to any branch of industry or science;
- 3) By extension, the use of electronic or mechanical devices to replace human labor.

According to Parasuraman et al. (2000), automation is not all or none, but can vary across a continuum of levels, from the lowest level of fully manual performance to the highest level of full automation (see Table 2).

Autonomous robots are robots which can perform desired tasks in unstructured environments without continuous human guidance. They are best-suited for applications that require accuracy in recurrences and high yield under stable conditions. Usually autonomous robots are used in structured environments, such as the production floor in a plant, and are required in fields, which demand reduction in manpower and workload (Holland and Nof, 1999).

A fully autonomous robot has the ability to (Murphy, 2000):

- Gain information about the environment.
- Work for an extended period without human intervention.
- Move either all or part of itself throughout its operating environment without human assistance.

- Avoid situations that are harmful to people, property, or itself unless those are part of its design specifications.

According to Rucci et al. (1999), autonomous robotic systems must possess a high degree of flexibility in order to adapt to the continuously changing conditions of the environment as well as to the information from their own sensors and motors. There are two important challenges which designers of autonomous robotic systems often face (Ng and Trivedi, 1998). The first deals with the nonlinear, real-time response requirements underlying the sensor–motor control formulation. The second deals with how to model and use the approach with which a human will address such a problem (Ng and Trivedi, 1998).

According to Murphy (2000), there are currently three paradigms for organizing intelligence in robots: hierarchical, reactive, and hybrid deliberative/reactive (Figure 1). The paradigms are described in two ways:

1. By the relationship between the three commonly accepted primitives of robotics: SENSE, PLAN, and ACT (Table 1).
2. By the way sensory data is processed and distributed through the system.

“If a function is taking in information from the robot's sensors and producing an output useful by other functions, then that function falls in the SENSE category. If the function is taking in information (either from sensors or its own knowledge about how the world works) and producing one or more tasks for the robot to perform that function is in the PLAN category. Functions which produce output commands to motor actuators fall into ACT.” (Murphy, 2000).

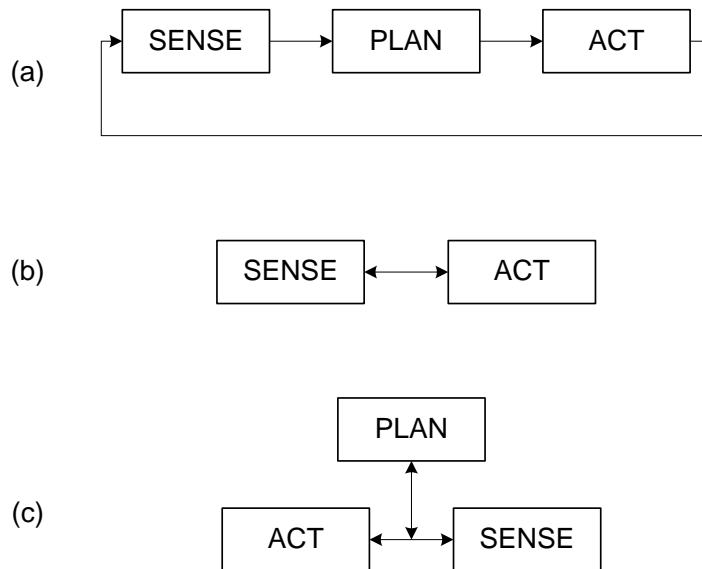
**Table 1: Robot primitives defined in terms of inputs and outputs (Murphy 2000).**

<b>Robot primitives</b>	<b>Input</b>	<b>Output</b>
<b>SENSE</b>	Sensor data	Sensed information
<b>PLAN</b>	Information (sensed and/or cognitive)	Directives
<b>ACT</b>	Sensed information or directives	Actuator commands

Application of autonomous robots in dynamic and changeable environments still produces inadequate results. Therefore, the use of an autonomous robotic device is not advisable (Al-Jumaily and Amin, 2000; Penin et al., 1998). An example can be seen in recognition tasks, where inadequacies in sensor and image processing technology have limited the capabilities of autonomous robotics in complex environments (Everett and Dubey, 1998). Moreover, having an automated system handle all conceivable scenarios is extremely



difficult and the promise of automatic and efficient remote operations has fallen short of expectations (Fletcher et al., 2005; Steinfeld, 2004).



**Figure 1: Three paradigms for organizing intelligence in robots: (a) Hierarchical, (b) Reactive, and (c) Hybrid deliberative/reactive (Murphy, 2000).**

## 2.2 Teleoperation and Human – Robot collaboration

“Teleoperation arose as an intermediate solution to tasks that required automation but for which robots could not be adequately programmed to handle” (Uttal, 1989). “Teleoperation methods typically are cognitive fatiguing, require high communication bandwidths and short communication delays, and require one or more teleoperators per remote” (Uttal, 1989).

“Teleoperation is when a human operator controls a robot from a distance (see Figure 2). The operator and robot have some type of master-slave relationship. In most cases, the human operator sits at a workstation and directs a robot through an interface. The teleoperator cannot look at what the remote is doing directly, either because the robot is physically remote or the local has to be shielded” (Murphy, 2000). Therefore, “the sensors which acquire information about the remote location, the display technology for allowing the operator to see the sensor data, and the communication link between the local and remote are critical components of a telesystem” (Uttal, 1989).

According to Wampler (1990), teleoperation is best suited for applications where:

- The tasks are unstructured and not repetitive.
- The task workspace cannot be engineered to permit the use of industrial manipulators.

- Key portions of the task intermittently require dexterous manipulation, especially hand-eye coordination.
- Key portions of the task require object recognition, situational awareness, or other advanced perception.
- The needs of the display technology do not exceed the limitations of the communication link (bandwidth, time delays).
- The availability of trained personnel is not an issue.

“A teleoperator is a machine that extends a person's sensing and/or manipulating capability to a location remote from that person” (Sheridan, 1992). “Virtually, by its definition, every human-robot collaborative system has a teleoperator. Since their first appearance in the 40’s, many teleoperated systems have been developed and employed for dealing with unstructured environments and in applications where there is clear and unavoidable danger for the human operator” (Sheridan, 1992).

In addition, robots are already increasingly being used in assistive technology, rehabilitation, surgery, therapy, service and entertainment domains. Methods which will enable easy and effective communication between robots and humans are crucial in all of these areas (Salter et al., 2004).

### 2.2.1 Human-robot collaboration models

According to Sheridan (1976), a telerobotics manipulator is a more advanced form of teleoperation in which a human operator supervises a robot through a computer moderator.

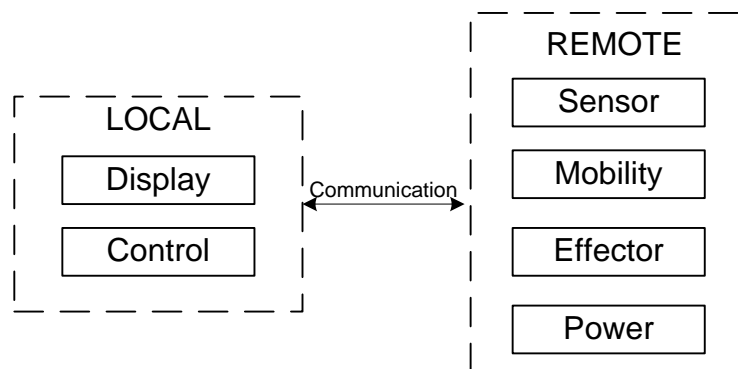


Figure 2: Organizing chart of a telesystem (Murphy 2000).

According to Sheridan (1992), human-robot collaboration means that one or more human operators are intermittently or continuously programming and receiving information from a computer that interconnects through artificial effectors and sensors to the controlled

process or task environment. “A man-machine system is an operating combination of human and equipment components, interacting to bring about, from given inputs, some desired outcome within the constraints of a given environment” (Sanders and McCormick, 1993). Humans have superior recognition capabilities and can easily adapt to changing environmental and object conditions (Rodriguez and Weisbin, 2003). Their acute perception capabilities enable humans to deal with a flexible, vague, changing, and wide scope of definitions (Chang et al., 1998). However, a human operator is not consistent, tends to fatigue, and suffers from distraction (Van Erp et al., 2004). In addition, human operators are known to make mistakes of overlooking collisions with surrounding objects, which result in expensive repairs and limit the system's effectiveness. People seem to be unable to navigate and manipulate remote equipment without colliding with objects in the environment (Ivanisevic and Lurnehky, 1998). The operators of many systems are expected to make control responses to bring about the desired operation of the system as implied by the input. In the absence of any scheme for helping the operator, such control can be complicated with higher-level control orders (Sanders and McCormick, 1993). “With respect to tracking performance, people perform about as well with a zero-order control system as they perform with a first-order control system, but performance really deteriorates when a second-order control system is used. Tracking error can increase from 40 to 100 percent” (Wickens, 1986).

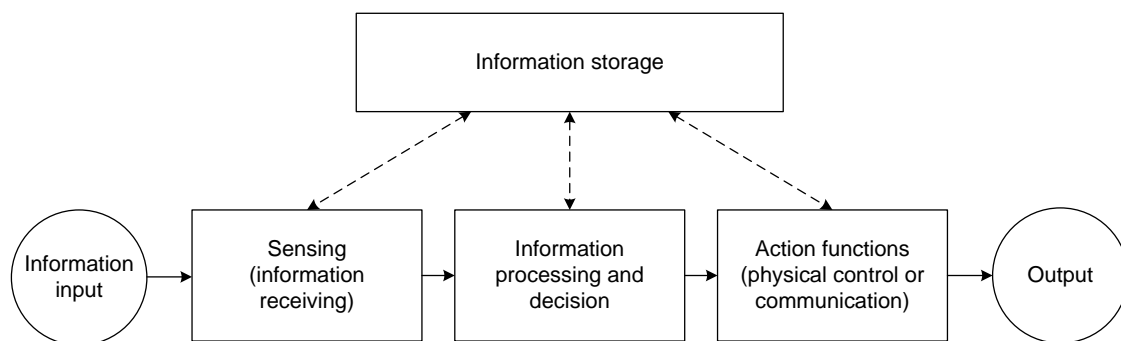
Wickens (1984) identifies specific information processing limitations of a human operator that affect tracking performance, i.e., processing time, bandwidth, and anticipation:

- Processing Time - People do not process information instantaneously, hence there is a time delay between a change in a target and the initiation of the responses required to track the target. The magnitude of the time delay is dependent on the order of the system being controlled (McRuer and Jex, 1967).
- Bandwidth refers to the upper limit of frequency with which corrective decisions can be made, and hence the term defines the maximum frequency of a random input that can be successfully tracked. This bandwidth is normally between 0.5 and 1.0 Hz (Elkind and Sprague, 1961).
- Anticipation - Often operators must track targets by using systems that have time lags or that respond sluggishly to control inputs. This requires that the operator anticipate future errors based on present conditions and then make control responses that are expected to reduce that anticipated future error. Unfortunately, humans are not very good at anticipating future outputs, especially for slow,

sluggish systems. Part of this difficulty is due to the limitations inherent in working memory. Making the calculations necessary to predict the future state of a higher-order system can stress all but the most experienced operators.

In reviewing the research relative to the possible merits of visual analog pursuit versus compensatory displays, Poulton (1974) concluded that when there is a choice between the two, a conventional pursuit (true motion) display is preferable to a compensatory (relative motion) display. Pursuit displays are generally better than compensatory displays since the operator can see the separate effects of target and controlled-element movements on the error generated. This makes it easier to predict the target's course and to learn the consequences of various control actions on the movement of the controlled element (Sanders and McCormick, 1993).

According to McCormick (1993), a man-machine system has certain properties or characteristics, including a purpose or objective; human components; system functions (sensing, information storage, information processing, decision, and action functions); system components (both men and machines); procedures; communication links; and some form of input and output (Figure 3).



**Figure 3: Types of function performed by man or machine components of man-machine systems (McCormick, 1993).**

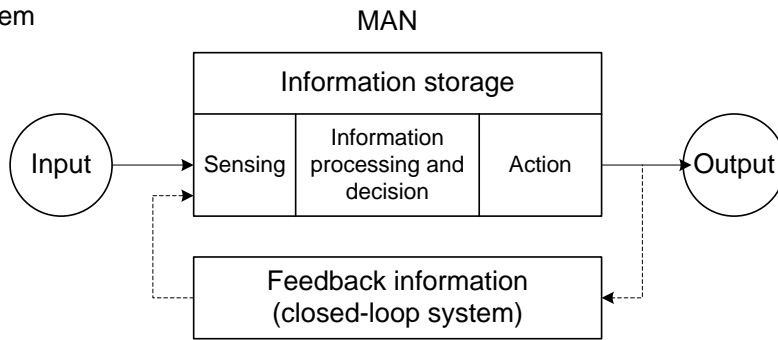
According to Rodriguez and Weisbin (2003), human and robot skills are complementary. The human perception, acting and thinking capabilities in dynamic environments are unmatched to those of robots, but there can be huge potential risks to human safety in getting these benefits. Robots provide complementary skills in being able to work in extremely risky environments, but their ability to perceive, think, and act on their own is far from flawless.

Technical developments in computer hardware and software now make it possible to introduce automation into virtually all aspects of man-machine systems. By taking advantage of the human perception skills and the autonomous systems' accuracy and consistency the

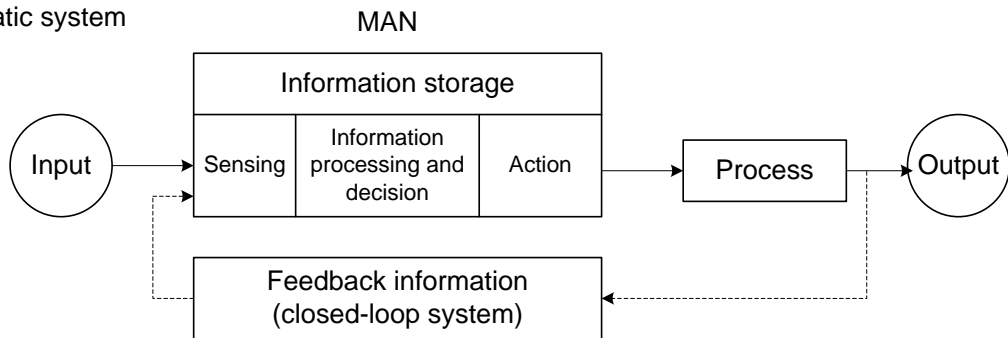
combined human-robotic system can be simplified, resulting in improved performance (Parasuraman et al., 2000).

McCormick (1993) defines different types of man-machine systems. “A closed-loop system is one that involves some continuous process which requires continuous control and in which there is feedback of some form that contributes to the continuous control process. An open-loop system is one in which feedback (if any is provided) does not contribute to subsequent control of the system (an open-loop system may be a continuous system, or non-continuous). A manual system is one in which power is supplied by the human being; typically it involves the use of non powered devices such as hand tools. A semiautomatic system is one in which all functions are performed by a machine component under human control. An automatic system (typically a closed-loop system) is one in which all functions are performed by the machine, including, in particular, the sensing and control functions; it is self-correcting” (McCormick, 1993), (Figure 4).

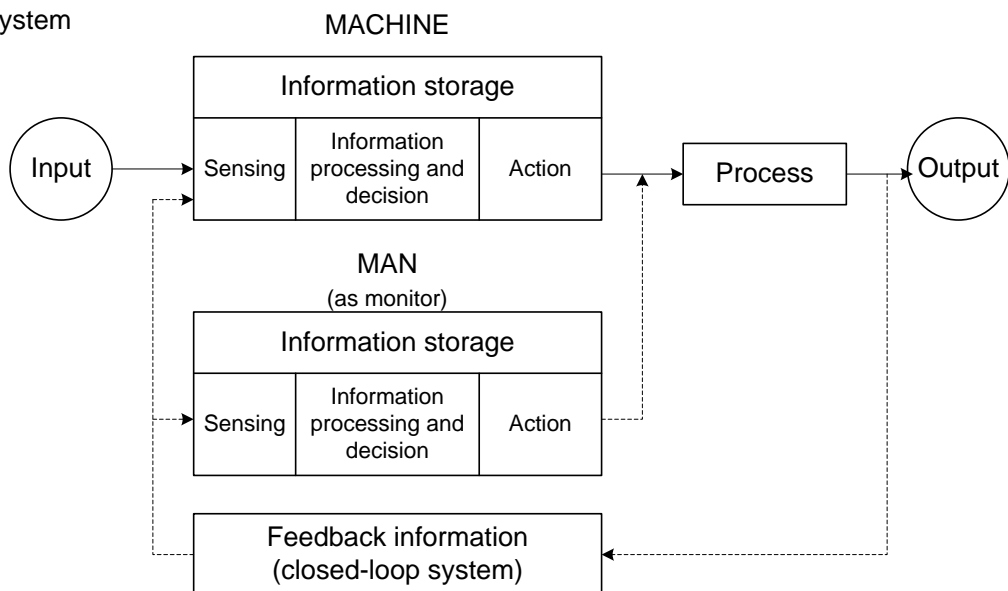
Manual system



Semiautomatic system



Automatic system



**Figure 4: Schematic illustration of manual, semiautomatic, and automatic man-machine systems. In the case of closed-loop (continuous) system, feedback information about conditions of the process is transmitted to the sensor (man or machine) for use in making necessary corrections in the control of the system (McCormick, 1993).**

### 2.2.2 Human-robot collaboration advantages

Penin et al. (1998) state five reasons for using telerobotics: i) ability to do and improve outage-free maintenance in countries with strict regulations regarding the interaction of humans with energized components; ii) increase the safety and comfort of the workers; iii) decrease the cost by eliminating the need for the operator to work in a hazardous environment; iv) ability to work under moderate bad weather conditions; and v) decrease in labor requirements.

The motivation for developing human-robot collaboration control has several reasons (Sheridan, 1992). First, it combines the advantages of the robot with the advantages of the human operator. Specifically, it achieves the accuracy, reliability and high yield of the robot with the cognitive capability and adaptability of the human. Moreover, by collaboration, the workload of the human operator is reduced and in the event of robot or human failure, either can reduce the damage. Second, it makes control possible even where there are time delays in communication between human and robot. Last, it saves lives and reduces cost by eliminating the need for the human operator to be present in hazardous environments (Sheridan, 1992).

According to Parasuraman et al. (2000), robot execution has been extended to functions that humans do not wish to perform, or cannot perform as accurately or reliably as robots. On the other hand, there is a large and rapidly developing class of technical systems that are dependent on human contribution for their operation.

“Telepresence techniques attempt to create a more natural interface for the human to control the robot and interpret what it is doing and seeing, but at a high communication cost” (Murphy, 2000). “Supervisory control attempts to delegate portions of the task to the remote, either to do autonomously (traded control) or with reduced, but continuous, human interaction (shared control)” (Murphy, 2000).

Various teleoperated systems, such as in space, nuclear reactors, and chemical cleanup sites provide excellent examples of human-robot collaboration in which the human operators plan and guide the motion of remotely situated devices through interaction with computer (Ivanisevic and Lurnehky, 1998).

Bechar and Edan (2003) provide proof of the advantage of such collaborations in target recognition tasks. According to their research, collaboration of human and robot increases detection by 4% when compared to a human operator alone and by 14% when compared to a fully autonomous system. In addition, when compared to the human alone, detection times of integrated systems are reduced by 20%.

### 2.2.3 Human-robot collaboration levels

Sheridan (1992) divided human-robot collaboration into ten levels from fully autonomous, without human intervention, to fully manual (Table 2).

**Table 2: Levels of automation of decision and action selection (Sheridan, 1992).**

HIGH 10. The computer decides everything, acts autonomously, ignoring the human

9. Informs the human only if it, the computer, decides to

8. Informs the human only if asked to, or

7. Executes automatically, then necessarily informs the human, and

6. Allows the human a restricted time to veto before automatic execution, or

5. Executes that suggestion if the human approves, or

4. Suggest one alternative

3. Narrows the selection down to a few, or

2. The computer offers a complete set of decision/action alternatives, or

LOW 1. The computer offers no assistance; human must make all decisions and actions

According to Sheridan (1992), human-robot control involves complex and flexible systems where operators have free will in planning, setting goals, and evaluation. Setting goals and decision making seems to be the most difficult aspect of human-robot control to model (Sheridan, 1992). In addition, “modeling free will seems paradoxical, since free will is determined from within the operator and not by an outside source” (Sheridan, 1992). This is also the reason why mental events can only be inferred and cannot be directly measured.

Parasuraman et al. (2000) outlined a model for types and levels of automation which attempt to provide a framework and an objective for making such choices. This model proposes that automation can be applied to four generic classes of functions: information acquisition, information analysis, decision and action selection and action implementation. Each class is independent and has its individual degree of automation which is determined by applying Sheridan's automation scale (Table 2). The model has two evaluation criteria. The primary criterion concerns the reduction in human capabilities due to the degree of automation. The second criterion concerns the automation reliability. Being an iterative model, the degree of automation can be changed after each evaluation.

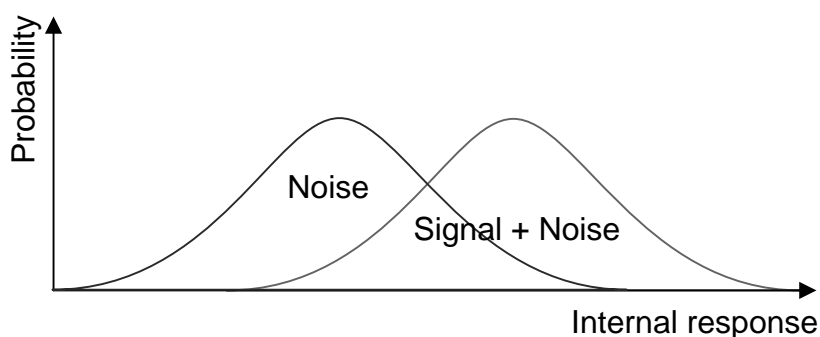


## 2.3 Signal Detection Theory

“Signal Detection Theory (SDT) is used to analyze data coming from experiments where the task is to categorize ambiguous stimuli which can be generated either by a known process, called the signal or be obtained by chance, called the noise in the SDT framework” (Green and Swets, 1966).

The starting point of signal detection theory is that nearly all decision making takes place in the presence of some uncertainty. “Signal detection theory provides a precise language and graphic notation for analyzing decision making in the presence of uncertainty” (Heeger, 1997).

This theory is often used in target recognition tasks, in radars or optical devices where picture sampling and identification of a target in these pictures is always characterized by uncertainty and presence of noise. Typical SDT chart for that purposes is presented in Figure 5.



**Figure 5: Internal response probability density functions for noise alone and for signal plus noise trials.**

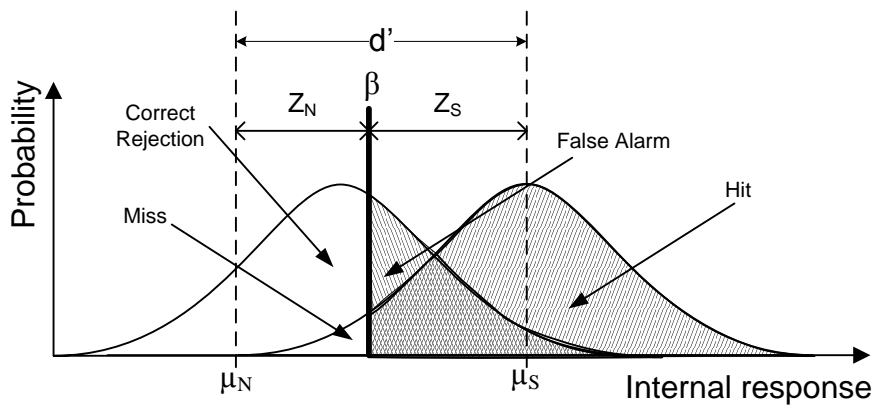
The goal of detection theory is to estimate two main parameters from the experimental data (Green and Swets, 1966). The first parameter, called  $d'$ , indicates the strength of the signal (relative to the noise). The second parameter called  $\beta$  criterion, reflects the strategy of response of the subject. In the detection process there are four types of responses: 1) hit – when a detector recognizes a target; 2) miss – when a detector does not recognize a target; 3) false alarm (FA) – when a detector recognizes a non-target object as a target; and 4) correct rejection (CR) - when a detector detects a non-target object as a non-target.

**Table 3: Classification of detections.**

Reality/Detection	Yes	No
Signal Present	Hit	Miss
Signal Absent	False Alarm	Correct Rejection

Perhaps the simplest strategy that the subject can adopt is to pick a criterion location along the internal response axis. Whenever the internal response is greater than this criterion they respond "yes". Whenever the internal response is less than this criterion they respond "no" (Brown and Davis, 2006).

An example criterion is indicated by the vertical line in Figure 6. The criterion line divides the graph into four sections that correspond to: hits, misses, false alarms, and correct rejections. On both hits and false alarms, the internal response is greater than the criterion, because the detector is responding "yes". Hits correspond to signal-plus-noise trials when the internal response is greater than criterion, as indicated in the Figure. False alarms correspond to noise-alone trials when the internal response is greater than criterion, as indicated in the Figure (Brown and Davis, 2006).



**Figure 6: Example of a criterion line that divides the density functions into four sections**

When the detector chooses has a low criterion, it will respond "yes" to almost everything. Then it will never miss a signal when it is present and therefore result in high hit rate. On the other hand, when responding "yes" to almost everything, the number of false alarms will also be increased. Thus, there is a cost to increasing the number of hits, and that cost is paid in terms of false alarms. If the detector chooses a high criterion then it responds

"no" to almost everything. It will rarely make a false alarm, but it will also miss many real signals (Brown and Davis, 2006).

There is no way that the detector can set its criterion to achieve only hits and no false alarms, unless the sensitivity also changed; it is inevitable that some mistakes will be made (Brown and Davis, 2006). Thus the detector cannot always be right. "It can adjust the kind of errors that it makes by manipulating its criterion, the one part of this diagram that is under their control" (Brown and Davis, 2006).

In this work a human-robot system is described as a system with two detectors corresponding to research by (Bechar, 2006). The performance of the first detector (robot) is determined by its sensitivity  $d'_r$  and its criterion  $\beta_r$  (Figure 7). The performance of the second detector (human) is determined by his sensitivity  $d'_h$  and two criteria; one for objects already marked by the robot,  $\beta_{rh}$ , and one for objects unmarked by the robot,  $\beta_h$  (Figure 8).

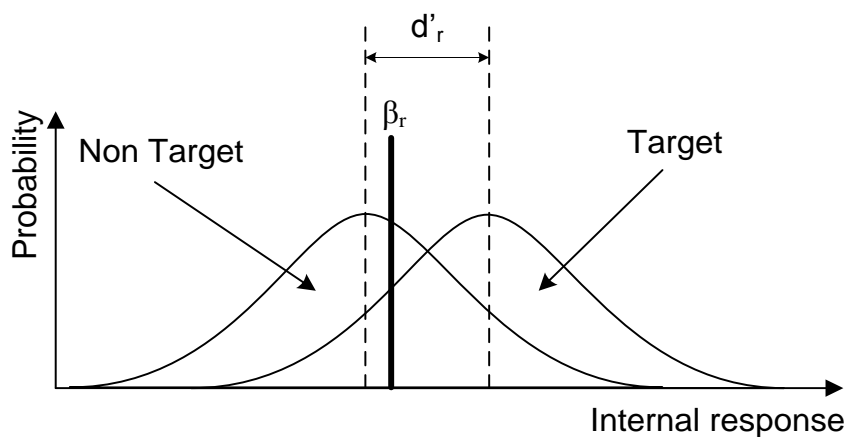


Figure 7: Modified SDT model for the robot

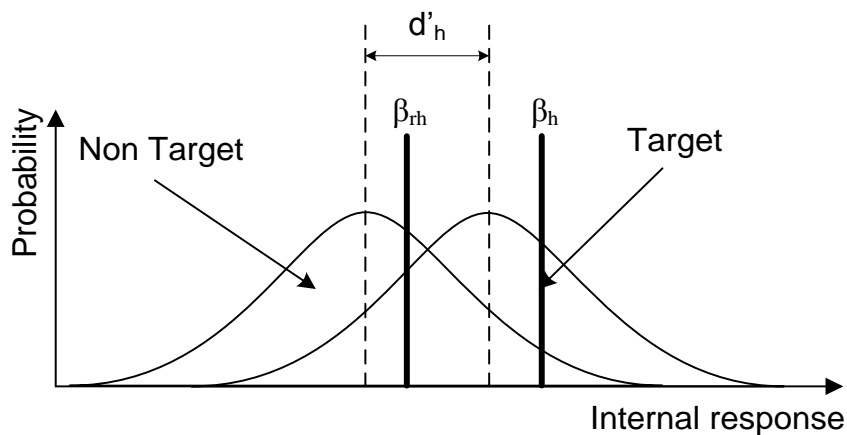


Figure 8: Modified SDT model for both human and robot

The symbols of signal detection theory that will be used in this work:

$\mu_{S,N}$  – signal/noise mean

$\sigma_{S,N}$  – signal/noise standard deviation

$Z_S$  – the distance in standard deviation units between  $x$  and  $\mu_S$  (along coordinate  $Z$ ).  $Z_S$  is positive

where  $x$  is bigger than  $\mu_S$  and negative where  $x$  is smaller than  $\mu_S$ .  $Z_S = \frac{x - \mu_S}{\sigma_S}$

$Z_N$  – the distance in standard deviation units between  $x$  and  $\mu_N$  (along coordinate  $Z$ ).  $Z_N$  is positive

where  $x$  is bigger than  $\mu_N$  and negative where  $x$  is smaller than  $\mu_N$ .  $Z_N = \frac{x - \mu_N}{\sigma_N}$

$P_S$  – probability that an object is a signal (target)

$f_S(Z_S)$  – the signal density function value at  $Z_S$ .  $f_S(Z_S) = \frac{e^{-\frac{Z_S^2}{2}}}{\sqrt{2\pi}}$

$f_N(Z_N)$  – the noise density function value at  $Z_N$ .  $f_N(Z_N) = \frac{e^{-\frac{Z_N^2}{2}}}{\sqrt{2\pi}}$

$d'$  – the distance between  $\mu_S$  and  $\mu_N$  on  $X$  coordinate.  $d' = \mu_S - \mu_N$

$\beta$  - the likelihood ratio of the two distributions at the cutoff point  $x$  (criterion).  $\beta = \frac{f_S(Z_S)}{f_N(Z_N)}$

$\beta^*$  - optimal  $\beta$  for one detector case.  $\beta^* = \frac{1 - P_S}{P_S} \times \frac{V_{CR} - V_{FA}}{V_H - V_M}$  (Swet et al., 2000)

Values extracted from Bechar et al. (2006):

$P_M$  – the probability of a miss.  $P_M(Z_S) = \int_{-\infty}^{Z_S} f_S(Z) dZ = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Z_S} e^{-\frac{Z^2}{2}} dZ$

$P_H$  – the probability of a hit.  $P_H(Z_S) = 1 - \int_{-\infty}^{Z_S} f_S(Z) dZ = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Z_S} e^{-\frac{Z^2}{2}} dZ = 1 - P_M$

$P_{CR}$  – the probability of correct rejection.  $P_{CR}(Z_N) = \int_{-\infty}^{Z_N} f_N(Z) dZ = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Z_N} e^{-\frac{Z^2}{2}} dZ$

$P_{FA}$  – the probability of a false alarm.  $P_{FA}(Z_N) = 1 - \int_{-\infty}^{Z_N} f_N(Z) dZ = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Z_N} e^{-\frac{Z^2}{2}} dZ = 1 - P_{CR}$

$V_{CR}$  – value of each correct rejection, positive values.

$V_{FA}$  – value of each false alarm, negative values.

$V_H$  – value of each hit, positive values.

$V_M$  – value of each miss, negative values.

$V_{AR}$  – payoff ratio.  $V_{AR} = -\frac{V_{FA}}{V_H}$

## 2.4 Filtering and Control

“When designing models for the signal and noise (targets and non targets) processing and applying SDT, it is possible, at least in principle, to design a filter that optimally enhances the signal relative to the noise” (Goodwin et al., 2001). A great deal is known about filter design procedures in simple cases, for example when the signal model is linear (Goodwin et al., 2001). When the signal and noise models are not completely specified, it seems plausible that appropriate models could be estimated by analyzing actual data. This is frequently done in practice by adaptive filters, especially when the models are ill defined or time varying (Goodwin et al., 2001). “Prediction is concerned with the problem of extrapolating a given time series into the future” (Goodwin et al., 2001). There is a vast array of design techniques for generating control strategies when the model of the system is known. When the model is unknown, on-line parameter estimation could be combined with on-line control. This leads to adaptive or self-learning controllers (Goodwin et al., 2001). “Control is concerned with the manipulation of the inputs to a system so that the outputs achieve certain specified objectives” (Goodwin et al., 2001).

According to Dorf (1989) the design and analysis of control systems is based on mathematical models of complex physical systems. The mathematical models, which follow from the physical laws of the process, are generally highly coupled nonlinear differential equations:

- Deterministic control (when there are no disturbances and the system model is known).
- Stochastic control (when there are disturbances and models are available for the system and disturbances).
- Adaptive control (when there may be disturbances and the models are not completely specified).

## 2.5 Previous work on collaboration levels in target recognition tasks

Based on Sheridan's scale of "action selection and automation of decision" (Table 1), Bechar et al. (2006), defined, tested and evaluated four basic levels for Human-Robot collaboration. The collaboration levels were designed specifically for target recognition tasks and adjusted to an extensive range of automation, from manual to fully autonomous. First is H, where the human operator detects and marks the desired target solely. This level is compatible to level 1 on Sheridan's scale. Second is HR, where the human marks targets, aided by recommendations from an automatic detection algorithm, i.e., the targets are automatically marked by a robot detection algorithm, the human acknowledges the robot's correct detections, ignores the false detections and marks the targets missed by the robot. This level is compatible to levels 3-4 on Sheridan's scale. The third level HOR, where targets are identified automatically by the robot's detection algorithm and the human's assignment is to cancel the false detections and mark the targets missed by the robot system, compatible with levels 5-7 on Sheridan's scale. The fourth level is R where the targets are marked automatically by the robot, compatible to level 10 on Sheridan's scale. The system objective function is designed to enable determination of the expected value of task performance, given the parameters of the system, the task, and the environment (Bechar et al., 2006).

### 2.5.1 Summary of Bechar's work (2006)

An objective function for target recognition in human-robot systems was developed to allow computation of the expected value of system performance given the parameters of the human, robot, environment and task. The objective function quantifies the multitude of influencing parameters through a weighted sum of performance measures, and enables the prediction of system performance and the desirable level of collaborations. It includes operational and time costs, both of which are important in the evaluation and optimization of system performance. It can also be applied to help design optimal systems for specific tasks.

The goal is to maximize the objective function. The system objective function in a target detection task ( $V_{Is}$ ) is composed of the four responses of the detection process and the system operational costs and can be defined as:

$$V_{Is} = V_{Hs} + V_{Ms} + V_{FAs} + V_{CRs} + V_{Ts} \quad (1)$$

where  $V_{Hs}$  (equation 2) is the system gain for target detection (hit),  $V_{FAs}$  (equation 4) is the system penalty for false alarms (FA),  $V_{Ms}$  (equation 3) is the system penalty for missing the target (miss),  $V_{CRs}$  (equation 5) is the system gain for correct rejection, and  $V_{Ts}$  (equation

6) is the system operation cost. All gain, penalty and cost values mentioned above have the same units (i.e., a common monetary value such as US dollar) combined in a single objective function. The gain function for detecting the targets is:

$$V_{H_s} = N \times P_s \times p_{H_s} \times V_H \quad (2)$$

where,

$N$  is the number of objects,

$P_s$  is the probability of an object becoming a target,

$V_H$  is the gain from a single hit, where the units of  $V_H$  are 'monetary value'. The value of  $V_H$  is target dependent (e.g., the price of one melon for the farmer).

$p_{H_s}$  is the system probability for a hit, composed of the human probability to confirm a robot hit and the probability to detect a target that the robot did not detect and that neither marked as a false alarm:

$$p_{H_s} = p_{H_r} \times p_{H_{rh}} + (1 - p_{H_r}) \times p_{H_h}$$

$p_{H_r}$  is the robot probability of a hit,

$p_{H_{rh}}$  is the human probability of confirming a robot hit, and

$p_{H_h}$  is the human probability of detecting a target which the robot did not detect.

The penalty of missed targets is shown in equation 3:

$$V_{M_s} = N \times P_s \times p_{M_s} \times V_M = N \times P_s \times (1 - p_{H_s}) \times V_M \quad (3)$$

where,

$V_M$  is the penalty of a single miss where the units of  $V_M$  are 'monetary value'. The value of  $V_M$  is target dependent (e.g., the damage created from not detecting one landmine can be the destruction of one vehicle).

$p_{M_s}$  is the probability of a system miss, composed of the human probability to not confirm a robot hit and the probability to miss a target that the robot did not detect and that neither marked as a FA:

$$p_{M_s} = p_{H_s} \times (1 - p_{H_{rh}}) + (1 - p_{H_r}) \times (1 - p_{H_h})$$

The penalty from false alarms is specified in equation 4:

$$V_{FAs} = F_{FAs} \times V_{FA} \quad (4)$$

where

$V_{FA}$  is the damage from a single false alarm, where the units of  $V_{FA}$  are 'monetary value'. The value of  $V_{FA}$  is system, environment and non-target object dependant (e.g. the

damage created by one non-target object to the machine or system, if the system will detect and pick a rock instead of a melon it could damage the robot or system mechanism).

$F_{FAS}$  is the number of system false alarm objects, composed of the robot's false alarms that the human does not correct and the human false alarm:

$$F_{FAS} = N \times (1 - P_S) \times [p_{FAR} \times p_{FA_{th}} + (1 - p_{FAR}) \times p_{FAh}]$$

$p_{FAR}$  is the robot false alarm probability,

$p_{FA_{th}}$  is the human probability of not correcting the robot false alarm, and

$p_{FAh}$  is the human false alarm probability.

The gain from correct rejection is specified in equation 5:

$$V_{CRs} = F_{CRs} \times V_{CR} \quad (5)$$

where

$V_{CR}$  is the gain from a single correct rejection, where the units of  $V_{CR}$  are 'monetary value'.

$F_{CRs}$  is the correct rejection density function for the system, composed of the robot correct rejections that the human does correct and the human correct rejection marks:

$$F_{CRs} = N \times (1 - P_S) \times [p_{FAR} \times (1 - p_{FA_{th}}) + (1 - p_{FAR}) \times (1 - p_{FAh})]$$

The system operational cost includes both costs of time and operation as illustrated in equation 6:

$$V_{Ts} = t_s \times V_t + (N \times P_S \times p_{Hs} + F_{FAS}) \times V_C \quad (6)$$

where,

$t_s$  is the system time that is required to perform the task,

$V_t$  is the cost of one time unit and its units are 'monetary value/time', and

$V_C$  is the cost of one object recognition operation (hit or false alarm) and its units are 'monetary/operations'. The cost values can be determined according to the time costs of the workers and the system and system operational costs and maintenance. The value of  $V_C$  is equal for hit and false alarms since it required the same treatment and manipulation for both.

Bechar (2006) assumed that the picking times are shorter than the sum of detection times and technical times related to the detection process. Therefore, the time terms in the objective function express only the detection times and do not consider the related operational time (picking times).

The system time consists of the time for the human to confirm the robot hits, the time for the human to hit additional targets, the time for the human to correct the robot false



alarms, the time for the human to mark false alarms, and the robot time to process the images and to perform hits or false alarms. Also included in  $t_s$  is the time it takes the human to decide whether an object has been correctly rejected (CR) or missed (M).

$$\begin{aligned}
t_s = & N \times P_S \times p_{Hr} \times p_{Hrh} \times t_{Hrh} + N \times P_S \times (1 - p_{Hr}) \times p_{Hh} \times t_{Hh} + \\
& + N \times (1 - P_S) \times p_{FAr} \times p_{FArh} \times t_{FArh} + N \times (1 - P_S) \times (1 - p_{FAr}) \times p_{FAh} \times t_{FAh} + \\
& + N \times P_S \times p_{Hr} \times (1 - p_{Hrh}) \times t_{Mrh} + N \times P_S \times (1 - p_{Hr}) \times (1 - p_{Hh}) \times t_{Mh} + \\
& + N \times (1 - P_S) \times p_{FAr} \times (1 - p_{FArh}) \times t_{CRrh} + N \times (1 - P_S) \times (1 - p_{FAr}) \times (1 - p_{FAh}) \times t_{CRh} + t_r
\end{aligned} \tag{7}$$

where,

$t_{Hrh}$  is the human time required to confirm a robot hit,

$t_{Hh}$  is the human time required to hit a target which the robot did not hit,

$t_{FArh}$  is the human time needed to correct a robot false alarm,

$t_{FAh}$  is the human false alarm time,

$t_{Mrh}$  is the human time lost when a robot hit is missed,

$t_{Mh}$  is the human time invested when missing a target which the robot did not hit,

$t_{CRrh}$  is the human time to correctly reject a robot false alarm,

$t_{CRh}$  is the HO correct rejection time, and

$t_r$  is the robot time.

Bechar (2006) assumed that each of the human time variables represents a superposition of a decision time,  $t_D$ , and a motoric time,  $t_M$ , in accordance with the collaboration level.

Explicit operation of the system objective function  $V_{Is}$ , that is suitable for all collaboration levels is described in equation 8:

(8)

$$\begin{aligned}
V_{Is} = & N \times P_S \times \left[ p_{Hr} \times p_{Hrh} \times (V_H + V_C + t_{Hrh} \times V_t) + (1 - p_{Hr}) \times p_{Hh} \times (V_H + V_C + t_{Hh} \times V_t) \right] + \\
& + N \times P_S \times \left[ p_{Hr} \times (1 - p_{Hrh}) \times (V_M + t_{Mrh} \times V_t) + (1 - p_{Hr}) \times (1 - p_{Hh}) \times (V_M + t_{Mh} \times V_t) \right] + \\
& + N \times (1 - P_S) \times \left[ p_{FAr} \times p_{FArh} \times (V_{FA} + V_C + t_{FArh} \times V_t) + (1 - p_{FAr}) \times p_{FAh} \times (V_{FA} + V_C + t_{FAh} \times V_t) \right] \\
& + N \times (1 - P_S) \times \left[ p_{FAr} \times (1 - p_{FArh}) \times (V_{CR} + t_{CRrh} \times V_t) + (1 - p_{FAr}) \times (1 - p_{FAh}) \times (V_{CR} + t_{CRh} \times V_t) \right] + t_r \times V_t
\end{aligned}$$

For the H collaboration level the objective function will be a degenerate form of equation 9, excluding the robot variables and therefore results in:

$$\begin{aligned}
V_{Is} = & N \times P_S \times \left[ p_{Hh} \times (V_H + V_C + t_{Hh} \times V_t) + (1 - p_{Hh}) \times (V_M + t_{Mh} \times V_t) \right] + \\
& + N \times (1 - P_S) \times \left[ p_{FAh} \times (V_{FA} + V_C + t_{FAh} \times V_t) + (1 - p_{FAh}) \times (V_{CR} + t_{CRh} \times V_t) \right]
\end{aligned} \tag{9}$$

In the R collaboration level the system objective function, VIs will be a degenerate form of equation 10 excluding the human variables:

(10)

$$V_{Is} = N \times P_S \times [p_{Hr} \times (V_H + V_C) + (1 - p_{Hr}) \times V_M] + N \times (1 - P_S) \times [p_{FAr} \times (V_{FA} + V_C) + (1 - p_{FAr}) \times V_{CR}] + t_r \times V_t$$

The time parameters for the H, HR, and HOR collaborations are shown in equations 11, 12, and 13, respectively.

$$\begin{aligned} t_{Hh} &= t_D + t_M \\ t_{FAh} &= t_D + t_M \\ t_{Mh} &= t_D \\ t_{CRh} &= t_D \end{aligned} \quad (11)$$

$$\begin{aligned} t_{Hh} &= t_D + t_M \\ t_{FAh} &= t_D + t_M \\ t_{Mh} &= t_D \\ t_{CRh} &= t_D \\ t_{Hrh} &= t_D + t_M \\ t_{FArh} &= t_D + t_M \\ t_{Mrh} &= t_D \\ t_{CRrh} &= t_D \end{aligned} \quad (12)$$

$$\begin{aligned} t_{Hh} &= t_D + t_M \\ t_{FAh} &= t_D + t_M \\ t_{Mh} &= t_D \\ t_{CRh} &= t_D \\ t_{Hrh} &= t_D \\ t_{FArh} &= t_D \\ t_{Mrh} &= t_D + t_M \\ t_{CRrh} &= t_D + t_M \end{aligned} \quad (13)$$

A methodology for determining the best collaboration level based on the human, robot, task, and environmental variables was developed by Bechar (2006). Numerical analysis of the developed objective function combined with signal detection theory was applied for the defined collaboration levels, and a sensitivity analysis of the influencing variables was performed on the optimum values (Bechar, 2006). These developments provide the basis for adjusting the combined human-robot system to each task and environment and aid in effective system design.

The main conclusions from Bechar's (2006) work are:

Numerical analysis results indicated that the best system performance, the optimal performance measures values, and the best collaboration level depend on task, environment, human, and robot parameters as well as the system characteristics. Since the number of independent parameters is vast and, in addition, there are interactions between the parameters, a prediction of system performance and the optimal solution is comprehensive and not obvious. However, it can be determined by investigating the objective function. The findings indicate that for the tested cases H is never the best collaboration level for the optimal solution, probably due to its high operational cost and low hit rate relative to the other collaboration levels. Thus, collaboration of human and robot in target recognition tasks will always improve the optimal performance of a single human detector. In addition, for the optimal solution of the objective function including operational costs, the best collaboration level is R when robot sensitivity is higher than human sensitivity. Moreover, the overall system sensitivity never decreases beneath the robot sensitivity.

The sensitivity analyses illustrated the influence of small variations, in the human and robot optimal values and in the environmental parameters, on the objective function and on the best collaboration level. Results indicated that small changes in the optimal values can cause shifts in the best collaboration levels from one to another but the shift is always to an adjacent level. A sensitivity analysis of the environmental target probability parameter showed that small changes in the optimal value can shift the best collaboration level from one to another and in some cases that shift leads directly to H. This finding can be exploited for the design and operation of integrated human-robot systems under dynamic and realistic conditions where the true value of the parameters is unknown and the resolution and accuracy are low, or in cases where the parameters are dynamic and drifting around their expected values.

Experimental results indicated that although the participants' performances were not optimal, they significantly reacted to the different robot, task and environmental parameters, and their results are consistent with the results of the numerical analysis of the objective function excluding the operational cost. Due to the unknown number of total objects, targets and non-targets, only part of the human and robot performance measures and the environmental parameters could be evaluated and compared to the numerical analysis.

### 2.5.2 Summary of Oren's work (2007)

A numerical analysis of the global system objective function with all its parts was conducted in the work of Oren (2007). The analysis determined the optimal collaboration level and the objective function score for a given human sensitivity, robot sensitivity, target probability and different ratios between the objective function weights. All analyses were performed for systems which work with optimal criterions.

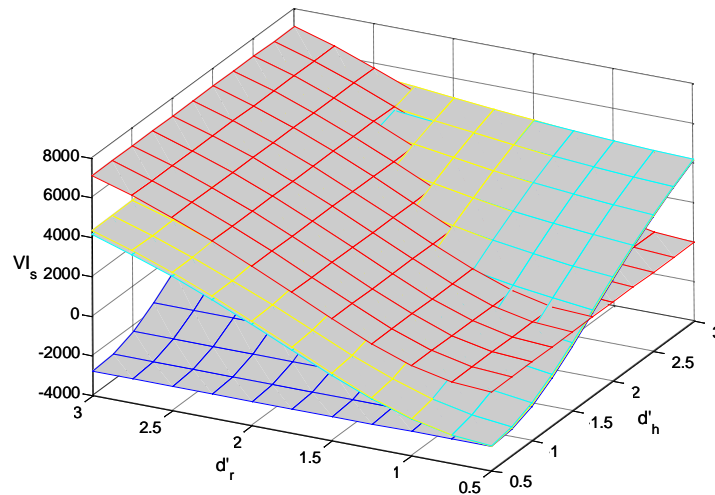


Figure 9: Objective function score for different human and robot sensitivities of the four collaboration levels for  $P_s=0.2$ . H – blue, HR – cyan, HOR yellow and R – red.

The Figure below presents the collaboration level required to achieve the best system performance.

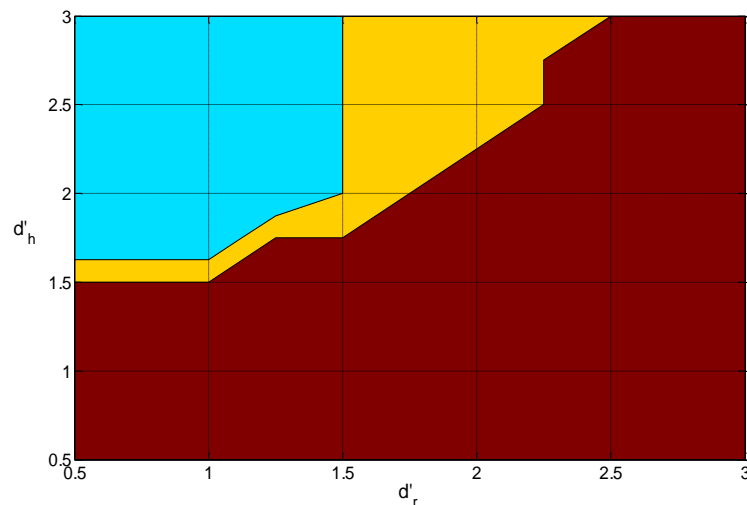


Figure 10: Best collaboration level map for different human and robot sensitivities. The colors represent different collaboration levels: HR – cyan, HOR - yellow and R – red.

The main conclusions from Oren's (2007) work are:

Numerical analysis exposed two different behavior types; each type consisting of different systems. For all of the systems examined, an increase in the human and/or robot's sensitivity led to an increase in the objective function score. This results from the fact that better sensitivity means better discrimination ability between target and noise (no target); despite the different weights given to the target function. Better sensitivity leads to more hits and less false alarms and thus regardless of the system's type or method of reward or penalty the objective function score rises. In addition, level H, the human operating alone, was not found to be preferable in any of the systems examined; this may be the result of high operational costs and a relatively low detection rate. The meaning of this finding is that collaboration between human and robot in target recognition tasks will always improve the system's performance. It appears that the improvement in detection rate and hence rise in profits gained by this collaboration outweigh the rise in operational cost attributable to adding the robot to the system.

The results showed opposite tendencies between the two types of systems found. In type one system as target probability increased R level was preferable in more cases, and as a result collaboration levels were less preferable. In type two systems the trend was reversed: as target probability increased collaboration levels were preferable in more cases. Type one systems greatly value not committing errors; that is to say, they place high importance on results in situations where no target is present, or target probability is low. In turn, type two systems greatly value results in which a target is present. Even though very different tendencies were discovered by the function analysis, several important similarities found between them should be pointed out: First, in both systems as the probability of the prominent object (non target in system one and target in system two) increases, R level will be preferable in more cases. In turn, as the probability of the prominent object decreases, collaboration between human and robot is preferable. It can be assumed that this trend stems from the reciprocation between operational costs and recognition profits.

Sensitivity analysis of the betas indicated that while R level was only found to be affected by the position of  $\beta_r$ , the two collaboration levels, HR and HOR, were found to be affected by all the three betas. This analysis has revealed that in many cases, a small deviation from the optimal value required the system to switch to another operational level in order to stay at an optimum.

## **3 Methodology**

### **3.1 Problem objective**

This research aims to develop a methodology for dynamic switching between collaboration levels of human-robot systems to maintain maximum system performance despite deviations in the parameter values during task performance.

### **3.2 Overview**

This thesis includes the following developments:

- A decision model for the dynamic switching execution.
- A closed-loop paradigm of the human – robot system based on the decision model.
- A rule-based algorithm to switch between collaboration levels based on human, robot, task, and environmental parameters for
  - Immediate response time.
  - Unknown response time of the system.
  - Restriction of minimal time for switching.
- A tool to implement the rule-based algorithm and model and simulate its performance.
- A comprehensive numerical analysis of the results.

In addition this research will answer some major questions concerning shifting between collaboration levels in dynamic human-robot systems:

1. Who decides on the shift, human, robot, or is it a collaborative decision?
2. How many levels can be switched at one time?
3. What parameters should be considered in the switch?
4. What should the frequencies of the switches be?

### **3.3 Human-robot system definition**

In this research we investigate an integrated human-robot system for target recognition tasks based on work developed by Bechar (2006). Definitions are according to Bechar (2006). The term 'system' is defined to represent both the 'human operator', 'robot' and 'controller' subsystems and includes their overall combined performances and parameters. The 'human operator' subsystem is defined by the displays at which the human perceives

information and by control devices which are used for manual operations; the ‘robot’ subsystem comprises the autonomous operations defined by automatic programs residing in the robot controller (Figure 12). The phrases 'human operator' and 'robot' refer to the subsystems and to their specific performances and parameters. The phrase 'environment' refers to the surrounding conditions the 'system' operates in. It includes parameters such as target probability, number of objects, and other parameters that are not related to the 'system'. The system inputs are the parameters  $d'$ ,  $\beta$  and  $P_s$ . These parameters can vary over time, thus affecting the system performance dynamically. The criterion,  $\beta$ , is assigned to an optimal value during system performance by comparing the different likelihood ratios. The parameters can be classified into three subgroups that include the human operator, robot and environment (Table 4).

**Table 4: Performance parameters (Bechar, 2006)**

Human performance		robot performance		environment performance
$p_{Hh}$	$P_s, d'_r, d'_h$	$p_{Hr}$	$P_s, d'_r$	$P_s, N$
$P_{Hrh}$				
$P_{FAh}$		$P_{FAr}$		
$P_{FArh}$				
$t_{Hh}$		$t_r$		
$t_{Hrh}$				
$t_{FAh}$				
$t_{FArh}$				
$t_{Mh}$				
$t_{Mrh}$				
$t_{CRh}$				
$t_{CRrh}$				

The system is defined as serial; each object is at first analyzed by the robot and then by the human operator. Nevertheless, the robot analysis is exposed to the human operator. In some cases the human response and the system outcome, or the system outcome by itself, can influence the robot threshold (Bechar et al., 2006).

The switching system was designed in this work so the robot can switch between collaboration levels, if it decides that the state is not optimal at some level. The human operator can view the displays (represented by “Displays” block in Figure 12) showing the system’s performance, and according to the current system’s collaboration level he/she can participate in target classification through control devices. The human operator can also decide whether to intervene to the current work state in any collaboration level if he/she thinks that the system’s state is not satisfactory.

The system is described as a closed-loop system (which is characterized by presence of feedback) because it must receive feedback of the current state of collaboration level which may vary over time.

### 3.4 Controller design

The purpose of the controller is to automatically switch between collaboration levels of the system to maintain high system performance during the target recognition task. The design of this controller is based on classical control methods for dynamic systems. The system should accumulate an optimal objective function score over time. Control theory was employed for that purpose. A controller manipulates the inputs to the system to obtain the desired effect on the output of the system. The controller manipulates a collaboration level (i.e., performs switching) to allow the process to obtain the desired reference.

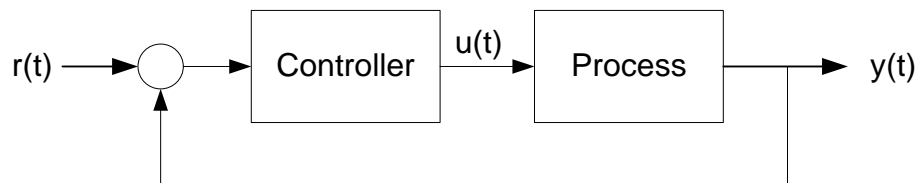


Figure 11: Block diagram of a closed loop control method.

Where  $r(t)$  is the input, i.e., desired reference (optimal objective function score),  $y(t)$  is the output, i.e., reference (current objective function score),  $u(t)$  is the input to the process which is optimal collaboration level OCL (output from the controller). The controller decides on the collaboration levels and makes a collaboration level switch if necessary, to provide the optimal collaboration level as a reference to the process. Therefore,  $u(t)$  provides the optimal collaboration level (as calculated by the controller) to the robot controller. The process stands for a target recognition process at a collaboration level provided by the controller.

The system block diagram is presented in Figure 12 including the controller and the robot with the human operator defined as a process of the target recognition system where each object is first analyzed by the robot and then by the human operator.

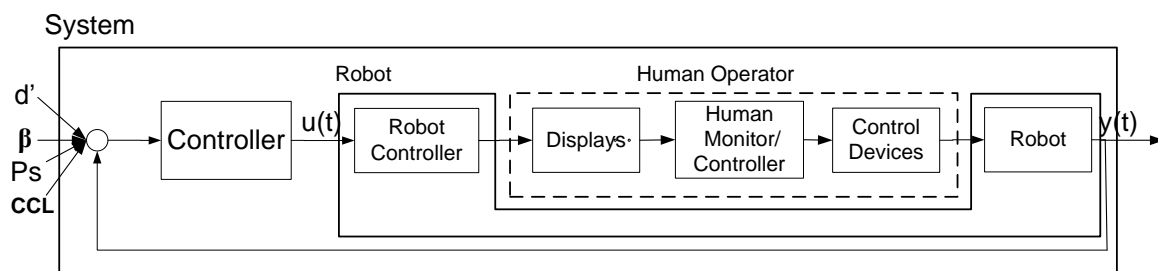


Figure 12: System's block diagram.



The controller is designed as a logical controller which receives four inputs:  $d'$ ,  $\beta$ ,  $P_s$  and the current collaboration level (CCL). Different algorithms were developed for the operation of this controller as detailed in section 5.2.

The basic logic for the control process is as follows: the controller obtains the optimal collaboration level from the inputs (which is the reference input  $r(t)$ ) by calculating the objective function (developed by Bechar, 2006) score for each collaboration level and compares it to the current collaboration level of the system. If the score of the optimal collaboration level (OCL) is not equal to the score of the current collaboration level (CCL) the controller makes a switch to the optimal collaboration level and provides a manipulated input to the process –  $u(t)$ . The system has four possible collaboration levels (H, HR, HOR and R), while each collaboration level has system objective function score.

### **3.5 Dynamic switching methodology**

The decision concerning the shift could be done by the human operator, the robot and the controller. Table 5 illustrates the switching decisions of the human operator and the robot. By default the human operator is given the higher priority for the switch. This will allow him to make corrections in case of some failures in the system's performance or in the robot or if he decides that the current performance is not satisfactory. In such a case the human operator can switch the collaboration level. The robot can decide to make a switch in collaboration level when it senses that system's state is unacceptable (performance too low for a period of time, some sensors suffer from malfunctioning, failure in the controller, etc.). The controller makes dynamic switching of the collaboration levels according to the switching algorithms in order to maintain adequate system performance. Thus, the switching operation is classified in three layers: the control layer, the robot layer and the operator layer. The upper layer is given an ability to by-pass the layer/layers beneath it (Figure 13). This grants also the ability to avoid system failure due to malfunctioning in one of the layers (robot or operator). The summary of the priorities and the decisions regarding the switching is presented in Table 5.

Switching to any level from any level will be enabled in the designed controller in order not to constrain the system, and make it more flexible to deal with changes of the environment and the parameters. Each change has an associated cost, depending on the jump in collaboration level, due to the response time required to perform this operation.

Table 5: Switching decisions.

	Priority	Decision type
<b>Human operator</b>	1	Human decides the system state is not satisfactory
<b>Robot</b>	2	Robot decides the state is sufficiently abnormal
<b>Controller</b>	3	According to switching algorithm

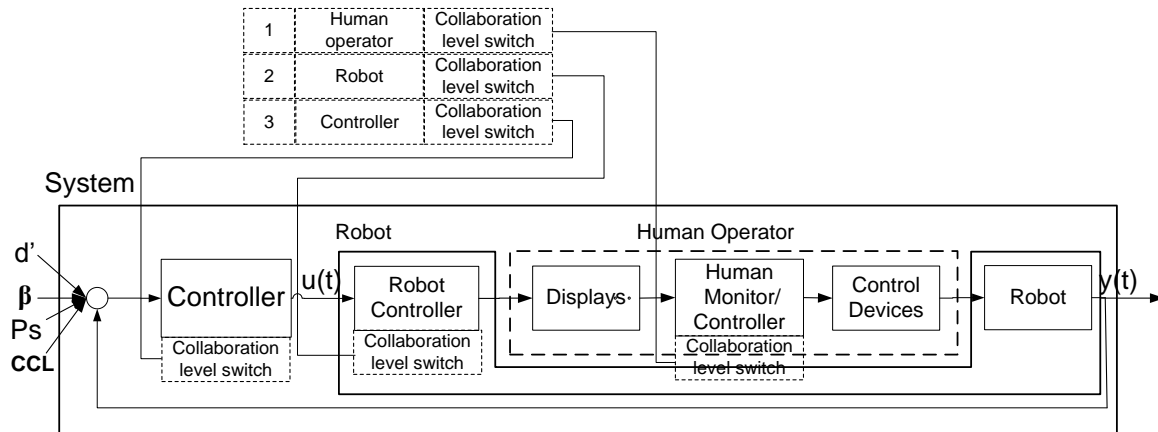


Figure 13: System's block diagram with collaboration switching priorities. In the upper part presented taxonomy of the switching layers and below illustration of blocks of collaboration level switch for each layer.

### 3.6 Assumptions

- Human performance has no influence on robot performance (Bechar, 2006).
- The human, robot, and system performances do not influence the appearance of target and non-target objects (Bechar, 2006).
- Target marking is done after each image sample without the ability to return to previous samples.
- A new image is sampled after a classification of a previous one (real-time work assumption).
- Noise and signal have the same distributions.
- System inputs can be obtained and given to the system prior to each image sample.

### 3.7 Switching objective function

The switching objective function is designed to enable determination of the expected value of the switching operation done by the controller, given the parameters of the system, the task, and the environment. The switching objective function parameters can be divided into three major categories – gain from system objective function, system's response time and switching frequency.

### 3.8 Switching algorithms

The dynamic switching algorithms are designed to switch the system to the best collaboration level by calculating the best objective function score with the current human, robot, environment and task parameters. Four different algorithms were developed for this purpose. The algorithms took into consideration the systems' response time, the changes of the system parameters due to this response time and limitation on the switching frequency.

### 3.9 Numerical Analysis

Numerical computations were performed on a PC with Matlab 7<sup>TM</sup> (The Mathworks, 2006) to simulate and evaluate the developed switching algorithms. The PC platform was Pentium Centrino 1.6GHz with 512mb of RAM.

The numerical computations were executed for several target probability distributions,  $P_s$ :

1. Linear distribution.
2. Uniformly distributed random values.
3. Normalized distribution of random values.
4. Normalized distributed of random values sorted by ascending order.

For each distribution, all the algorithms were simulated and compared by their results of increasing the objective function score. The simulation was performed for four different starting collaboration levels.

The values of the different parameters of the simulation were extracted from a preliminary experiment performed by Bechar et al. (2006). The value of hit weight,  $V_H$ , was set to 10, the cost of a single miss,  $V_M$ , was set to 5, the benefit of correct rejection,  $V_{CR}$ , was set to 3 and the damage from false alarm,  $V_{FA}$ , was set to -50. The probability for target,  $P_s$ , ranged from 0.1 to 0.9. The human sensitivity,  $d'_h$ , and the robot sensitivity,  $d'_r$ , were set arbitrary to examine the influence of different  $P_s$  distributions. The operational cost weights were constant where the cost for one system action was set to  $V_C=-2$  and the cost for one time unit was set to  $V_t=-2000 \text{ hr}^{-1}$ . The number of objects in each image was set to  $N= 1000$ . The decision time for all human time parameters was set to  $t_D=5 \text{ sec/object}$ , and the human motoric time was set to  $t_M=2 \text{ sec}/(\text{detected object})$ . The robot time was set to  $t_r=0.01 \text{ sec/object}$  on average. The system's response time and the frequency were set to  $t_{\text{response}}=5 \text{ sec}$  and  $\Psi=1 \cdot 10^{-6} \text{ sec}$ , to comply with the CPU frequency during the simulation (in order to simulate the real image sampling process). The  $V_p$  value was set to  $-5 \cdot 10^9$  to constrain the switching operation. The simulations were also executed for different frequencies and provided different results; this is discussed further in this work.

## 4 Switching objective function

This chapter deals with formulation of a switching objective function of a human-robot system based on the system objective function and determination of the gain in system performance value due to the switches.

### 4.1 Switching objective function formulation

The switching objective function is formulated to quantify the switching operations and to enable determination of the expected value of the switching operation done by the controller, given the parameters of the system, the task, and the environment.

For presentation of the terms used in this work for formulation of a dynamic switching methodology, the following symbols are introduced here:

The switching points are marked by  $\omega_i$ .  $i$  – is the index of the switching points.

The image sampling points are marked by  $\zeta_i$ .  $i$  – is the index of the sampling points.

The switching objective function score is marked by  $V_{ISwitch}$ .

The switching objective function ( $V_{ISwitch}$ ) evaluates the total gain from switching the system from one collaboration level to another. This gain is calculated as the difference between the system objective function<sup>1</sup> score in optimal collaboration ( $V_{ISOptimal}$ ) level and the objective function score in current collaboration level ( $V_{IScurrent}$ ).

$$V_{ISwitch} = (V_{ISOptimal} - V_{IScurrent}) \quad (14)$$

To find the optimal collaboration level, the total objective function score was calculated for all collaboration levels. The optimal collaboration level is the level that achieves a maximal score in the objective function (equation 15).

$$OCL = \begin{cases} H & \text{if } V_{IS}(H) = \text{Max} \{V_{IS}(H), V_{IS}(HOR), V_{IS}(HR), V_{IS}(R)\} \\ HOR & \text{if } V_{IS}(HOR) = \text{Max} \{V_{IS}(H), V_{IS}(HOR), V_{IS}(HR), V_{IS}(R)\} \\ HR & \text{if } V_{IS}(HR) = \text{Max} \{V_{IS}(H), V_{IS}(HOR), V_{IS}(HR), V_{IS}(R)\} \\ R & \text{if } V_{IS}(R) = \text{Max} \{V_{IS}(H), V_{IS}(HOR), V_{IS}(HR), V_{IS}(R)\} \end{cases} \quad (15)$$

$$V_{ISOptimal} = V_{IS}(OCL) \quad (16)$$

---

<sup>1</sup> The system objective function (function 1) was developed by Bechar (2006).

The current collaboration level is described as the level that the system is currently working at (i.e., the level at which the system was before switching). CCL can be determined by sampling the system at a desired time -  $t_{now}$  (equation 17).

$$CCL = \{H, HOR, HR, R\}_{t_{now}} \quad (17)$$

$$V_{IS_{current}} = V_{IS}(CCL) \quad (18)$$

where (from Bechar et al., 2006),

- a) H: The H detects and marks the desired target solely.
- b) HR: The H marks targets, aided by recommendations from an automatic detection algorithm, i.e., the targets are automatically marked by a robot detection algorithm, the human acknowledges the robot's correct detections, ignores false detections and marks targets missed by the robot.
- c) HOR: targets are identified automatically by the robot's detection algorithm; the human's assignment is to unmark false detections and to mark the targets missed by the robot system.
- d) R: the targets are marked automatically by the system.

At the switching point, the penalty for the response time of the system (both human and robot) for making the switch of collaboration level must be considered. This indicates whether the penalty for the system's response time is not greater than the benefit from switching operation. Therefore, a correction to the switching objective function was made in equation 19.

$$V_{I_{Switch}} = (V_{IS_{optimal}} - V_{IS_{current}}) + t_{response} \times V_t \quad (19)$$

$V_t$  is the cost of one time unit and its units are 'monetary value/time' (extracted from objective function).

The response time of the system consists of human operator response time and robot's response time (equation 20).

$$t_{response} = t_{h_{response}} + t_{r_{response}} \quad (20)$$

Where,

- a)  $t_{r_{response}}$  - Time required for the robot to switch between collaboration levels.
- b)  $t_{h_{response}}$  - Time required for the human to identify and adapt to switching.

The human operator consumes a different amount of time adapting to switching of different number of collaboration levels, therefore:

$$t_{h\_response} = t_{h\_x} \Big|_{x=1,2,3} \quad (21)$$

Where,

- a)  $t_{h\_1}$  - Time required for the human to adapt to switching to a nearest collaboration level.
- b)  $t_{h\_2}$  - Time required for the human to adapt to jumping of two collaboration levels.
- c)  $t_{h\_3}$  - Time required for the human to adapt to jumping of three collaboration levels.

This was introduced in the literature review, which indicated that humans have built-in time lags and limited bandwidths and are poor at anticipating future system states; in addition, control order, preview, and time lags influence tracking performance (Wickens, 1984).

A limit on the switching frequency is introduced to enable proper functionality of the human operator in case of constant switching. Elkind and Sprague (1960) define this frequency as a bandwidth with which corrective decisions can be made by human operator with values between 0.5 and 1.0 Hz. This leads to formulation of switching points (equation 22):

$$t(\omega_{k+1}) = t(\omega_k) + \tau \quad (22)$$

Where,

- a)  $\omega_k$  - Switch point indexed k.
- b)  $\tau$  - Time between switches.

The nominal time between switches defined by the switching frequency is marked by  $\psi$ .

The updated switching objective function that includes switching frequency (equation 23) is:

$$V_{I\_Switch} = (V_{IS\_optimal} - V_{IS\_current}) + t_{response} \times V_t + \Psi \times V_p \quad (23)$$

$$\Psi = \begin{cases} \psi - \tau & \text{if } \tau < \psi \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

Where,

- a)  $V_p$  - Penalty for switching earlier than the nominal value of switching frequency.
- b)  $\psi$  - The nominal time between switching operations.
- c)  $\Psi$  - The deviation value from switching frequency.

Formulation of the gain from switching is described as follow:

$$\tilde{\sigma} = \sum_{i=0}^n V_{I_{\text{Switch}}}(\omega_i), \quad 0 < n < m \quad (25)$$

$$\begin{aligned} \bar{\sigma}(\zeta_k) &= V_{I_{\text{current}}}(\zeta_k) - V_{I_{\text{reference}}}(\zeta_k) + V_{I_{\text{Switch}}}(\zeta_k) \\ V_{I_{\text{Switch}}}(\zeta_k) &= \begin{cases} V_{I_{\text{Switch}}}(\omega_k) & \text{if } k \text{ is switch point} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (26)$$

$$f = \sum_{k=1}^m \bar{\sigma}(\zeta_k) \quad (27)$$

$$F(s) = \text{Max}_{\{\bar{\sigma}\}} \left\{ \sum_{k=1}^m \bar{\sigma}(\zeta_k) \right\}, \quad s \in S \quad (28)$$

Where,

- a)  $\tilde{\sigma}$  - Sum of switching objective functions at switching points - gross switching gain.
- b)  $n$  - Total number of switches.
- c)  $\bar{\sigma}$  - Gain in the objective function score achieved by each system's operation (image sampling) compared to a particular collaboration level that used as a reference.
- d)  $m$  - Total number of image samples.
- e)  $\zeta_k$  - Image sample indexed  $k$ .
- f)  $V_{I_{\text{reference}}}$  - Objective function score of particular collaboration level that used as a reference.
- g)  $f$  - Cumulative gain of the entire operation of the system - net switching gain.
- h)  $F(s)$  - Fitness function of the model.
- i)  $S$  - Solution space for  $F$ .
- j)  $s$  - Possible solution for  $F$ .

## 5 Switching algorithms

This chapter presents the design of a controller for the human-robot system and the four switching algorithms developed for the controlled dynamic switching operation.

### 5.1 Basic switching algorithm for immediate response time of the system

The controller obtains the optimal collaboration level by calculating and comparing objective function scores for all collaboration levels. The controller compares the optimal collaboration level to the current collaboration level of the system. If the optimal collaboration level (OCL) is not equal to the current collaboration level (CCL) then the controller makes a switch to the optimal collaboration level. The algorithm is described in pseudocode in Figure 14 and in Table A-7 in Appendix III.

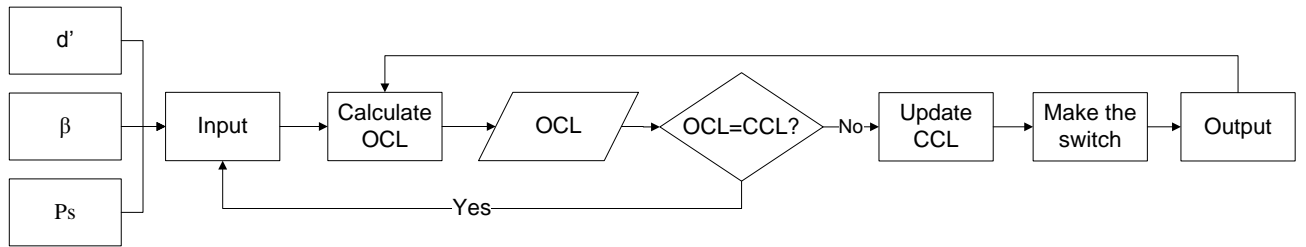


Figure 14: Basic switching algorithm for immediate response time of the system

### 5.2 Switching algorithms formulation

The following four algorithms were developed for the control layer:

“RLSA” - regular switch algorithm, “CTSA” - constrained switch algorithm, “CFSA” - constrained without frequency limitation switch algorithm and “PRSA” - predictive switch algorithm.

These algorithms perform under the limitation of the switching execution frequency, which does not allow making switches too frequently. The four algorithms differ by their set of conditions for making the switch of the collaboration level. Algorithm “RLSA” makes a switch whenever the switching objective function gain ( $V_{\text{Switch}}$ ) as formulated in equation 19 is positive. Algorithm “CTSA” makes switches whenever  $V_{\text{Switch}}$  value is higher than a desired threshold. Algorithm “CFSA” also makes switches whenever  $V_{\text{Switch}}$  value is higher than a desired threshold like “CTSA” algorithm but with no limitation of the switching execution frequency, which became possible due to definition of a penalty for breaking the frequency (equation 23). Algorithm “PRSA” makes switches by including predictions from



past data to operation of “CFSA” algorithm. It switches the system to the optimal collaboration level only when this collaboration level was optimal in most image samples analyzed by the system.

A logic model block diagram was built to represent the switching methodologies for all the algorithms (Figure 15). For example, if the current collaboration level is not optimal and the gain from the switch would be above desired threshold then “CTSA” algorithm should make a switch. The values in brackets are shortening names for each block.

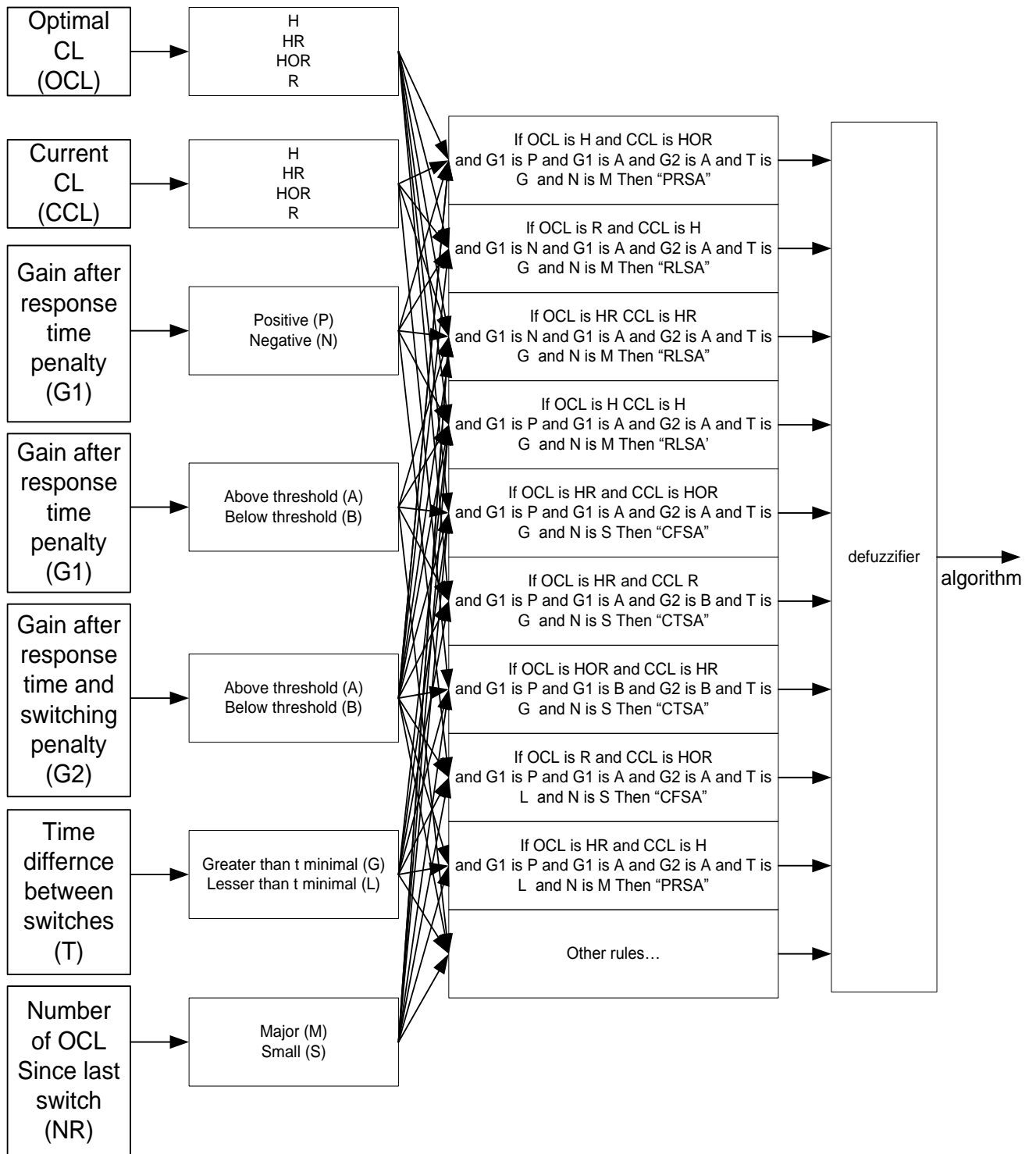


Figure 15: Block diagram of the control system algorithms

### 5.3 “RLSA” algorithm

The controller receives four inputs:  $d'$ ,  $\beta$ ,  $P_s$  and the current collaboration level. The controller obtains the optimal collaboration level by calculating and comparing objective function scores for all collaboration levels. It compares the obtained level to the current collaboration level of the system. Switching to the optimal collaboration level is conducted based on three conditions:

- If the optimal collaboration level (OCL) is not equal to the current collaboration level (CCL) and,
- If the gain achieved by switching as formulated in equation 19 is positive and,
- If the time difference between switches ( $\tau$ ) is greater than the nominal time between switches ( $\Psi$ )

The time difference between switches ( $\tau$ ) is updated in order to calculate the new time difference for the following switches. If the controller's conditions are not met then an output stating “Collaboration level not optimal” is placed on the displays (illustrated in Appendix I) and the human operator has an ability to intervene and change the collaboration level manually if he/she decides so. The algorithm is described in pseudocode in Figure 16 and in Table A-5 in Appendix III.

### 5.4 “CTSA” algorithm

The controller receives four inputs:  $d'$ ,  $\beta$ ,  $P_s$  and the current collaboration level. The controller obtains the optimal collaboration level by calculating and comparing objective function scores for all collaboration levels. It compares the obtained level to the current collaboration level of the system. Switching to the optimal collaboration level is conducted based on three conditions:

- If the optimal collaboration level (OCL) is not equal to the current collaboration level (CCL) and,
- If the gain achieved by switching as formulated in equation 19 is above desired threshold and,
- If the time difference between switches ( $\tau$ ) is greater than the nominal time between switches ( $\Psi$ )

The time difference between switches ( $\tau$ ) is updated in order to calculate the new time difference for the following switches. If the controller's conditions are not met then an output stating “Collaboration level not optimal” is placed on the displays (illustrated in

Appendix I) and the human operator has an ability to intervene and change the collaboration level manually if he/she decides so. The algorithm is described in pseudocode in Figure 17 and in Table A-6 in Appendix III.

### **5.5 “CFSA” algorithm**

The controller receives four inputs:  $d'$ ,  $\beta$ ,  $P_s$  and the current collaboration level. The controller obtains the optimal collaboration level by calculating and comparing objective function scores for all collaboration levels. It compares the obtained level to the current collaboration level of the system. Switching to the optimal collaboration level is conducted based on three conditions:

- If the optimal collaboration level (OCL) is not equal to the current collaboration level (CCL) and,
- If the gain achieved by switching as formulated in equation 23 is above desired threshold

The time difference between switches ( $\tau$ ) is updated in order to calculate the new time difference for the following switches. If the controller's conditions are not met then an output stating “Collaboration level not optimal” is placed on the displays (illustrated in Appendix I) and the human operator has an ability to intervene and change the collaboration level manually if he/she decides so. The algorithm is described in pseudocode in Figure 18 and in Table A-7 in Appendix III.

### **5.6 “PRSA” algorithm**

The controller receives four inputs:  $d'$ ,  $\beta$ ,  $P_s$  and the current collaboration level. The controller obtains the optimal collaboration level by calculating and comparing objective function scores for all collaboration levels. It compares the obtained level to the current collaboration level of the system. Switching to the optimal collaboration level is conducted based on three conditions:

- If the optimal collaboration level (OCL) is not equal to the current collaboration level (CCL) and,
- If the gain achieved by switching as formulated in equation 23 is positive or time difference between switches ( $\tau$ ) is greater than the nominal time between switches ( $\Psi$ ) and,

- If there is a majority of optimal collaboration levels of the same type since the last switch

The time difference between switches ( $\tau$ ) is updated in order to calculate the new time difference for the following switches. If the controller's conditions are not met then an output stating "Collaboration level not optimal" is placed on the displays (illustrated in Appendix I) and the human operator has an ability to intervene and change the collaboration level manually if he/she decides so. The algorithm is described in pseudocode in Figure 19 and in Table A-8 in Appendix III.

Every algorithm calculates the gains that were achieved by switching according to equations 26 and 27.

**Table 6: Summary of the terms**

<b>Term</b>	<b>Meaning</b>
OCL	Optimal collaboration level
CCL	Current collaboration level
G1	Gain after response time penalty
G2	Gain after response time and switch frequency penalty
T	Time difference between switches
NR	Number of OCLs since last switch
P	Positive
N	Negative
A	Above threshold
B	Below threshold
G	Greater than $\Psi$
L	Lesser than $\Psi$
M	Major
S	Small

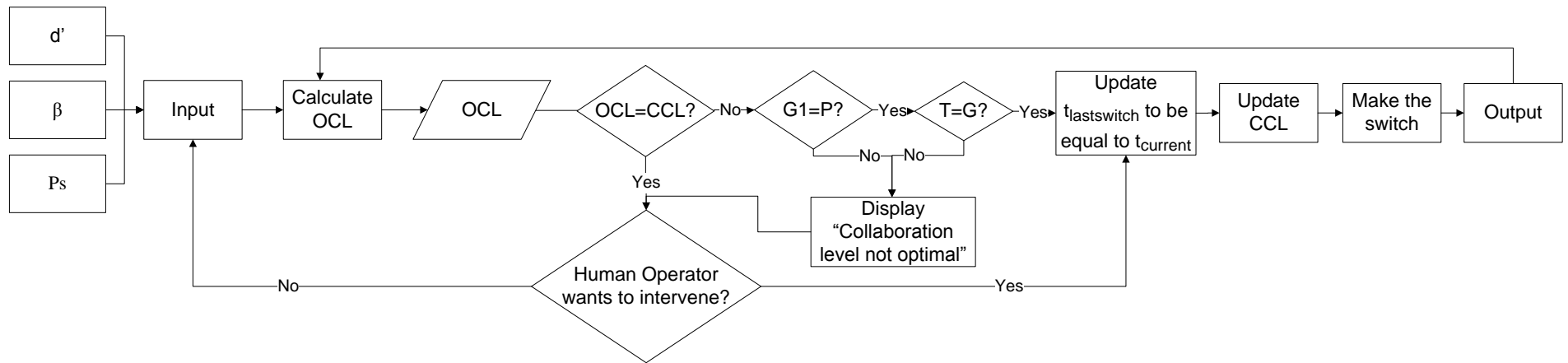


Figure 16: Block diagram of “RLSA” algorithm

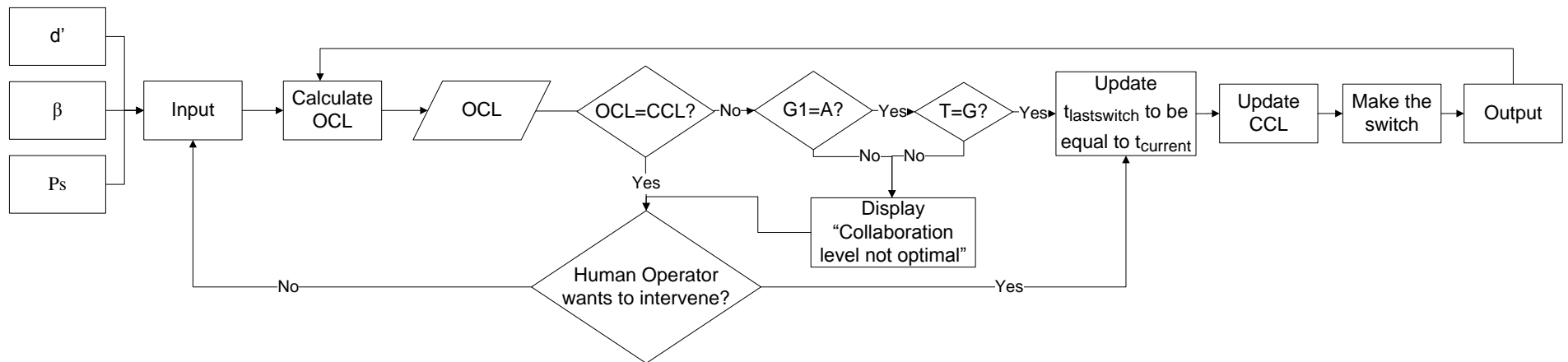


Figure 17: Block diagram of “CTSA” algorithm

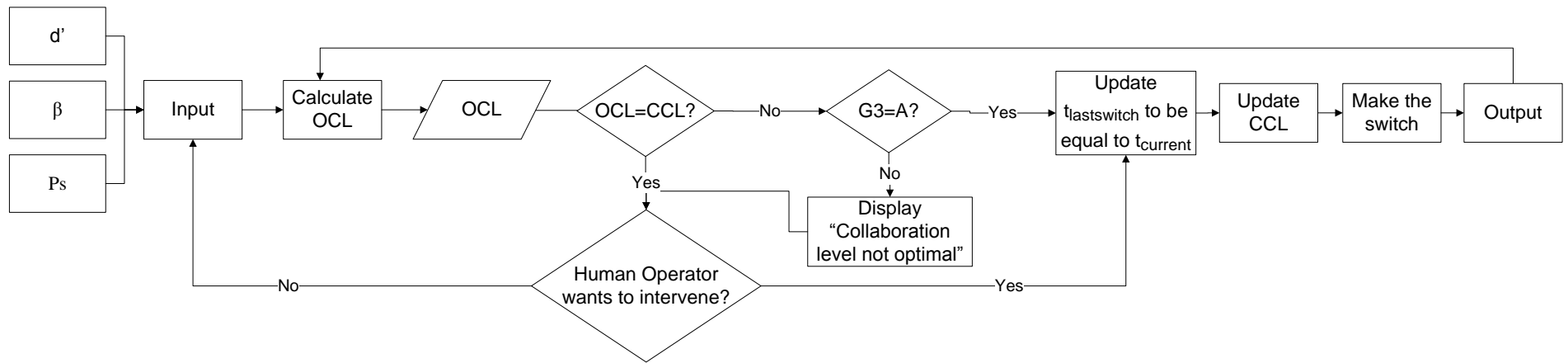


Figure 18: Block diagram of “CFSA” algorithm

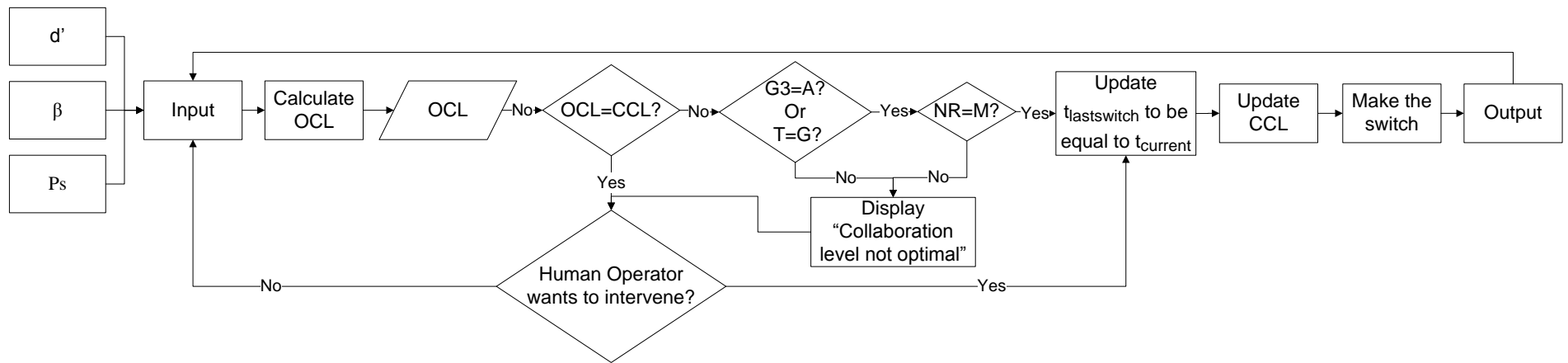


Figure 19: Block diagram of “PRSA” algorithm

## 5.7 Parameters classification

There are several groups of parameters used in the algorithms mentioned above. The first group is the algorithm's calculated variables, which are calculated in the computing process during the task. The second is user defined parameters. These parameters should be defined by the operator of the system to allow him to customize its performance according to his needs. The third group is system oriented parameters, which depend on a system, like system's response time. These parameters should be determined a priori from the desired system.

The classification of the parameters:

Algorithm calculated variables:  $V_{IS_{reference}}$ ,  $\tilde{\sigma}$ ,  $\tau$ ,  $\omega_k$ ,  $V_{IS_{optimal}}$ ,  $V_{IS_{current}}$ ,  $\bar{\sigma}$ ,  $n$ ,  $m$ ,  $\zeta_k$ ,  $f$ ,  $F$

User defined parameters:  $\psi$ ,  $B$ ,  $m$ ,  $V_p$

System dependable parameters (system oriented):  $t_{r\_response}$ ,  $t_{h\_response}$ ,  $V_t$



## 6 Numerical analysis

This chapter presents numerical analysis of the switching algorithms and their operation, and the analysis of switching points.

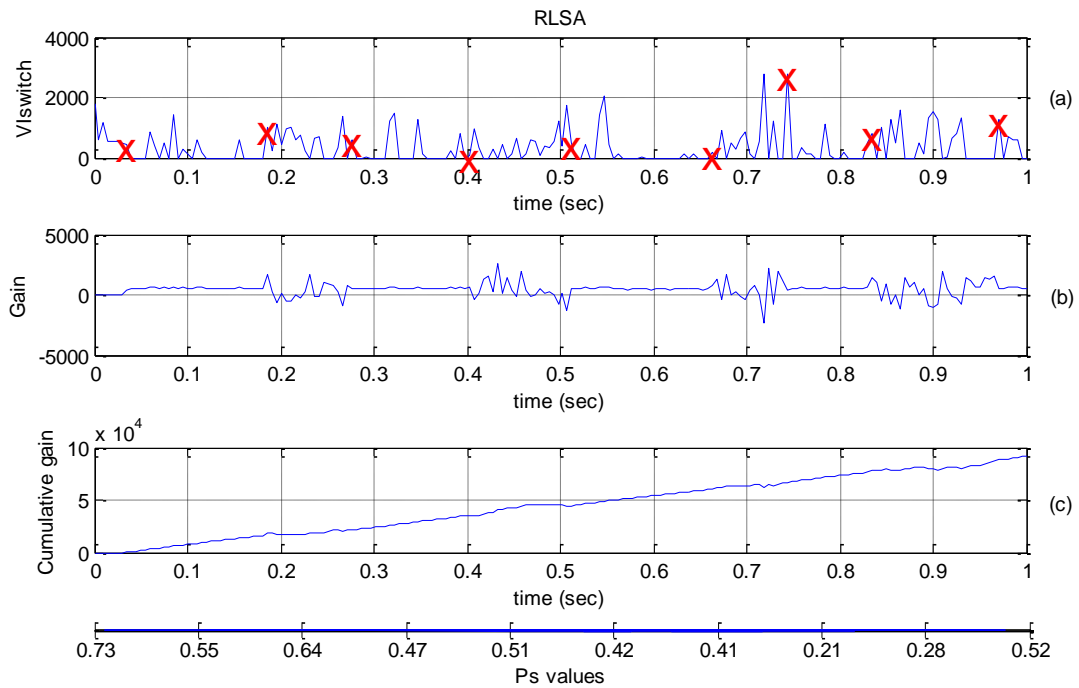
### 6.1 Numerical simulation of switching algorithms

The main goal of these simulations is to check the feasibility of performing automatic switches between collaboration levels in a dynamic human – robot system. The second aim is to check the benefit of automatic switching – does switching cause an increase in system performance and if so, how much does it influence performance?

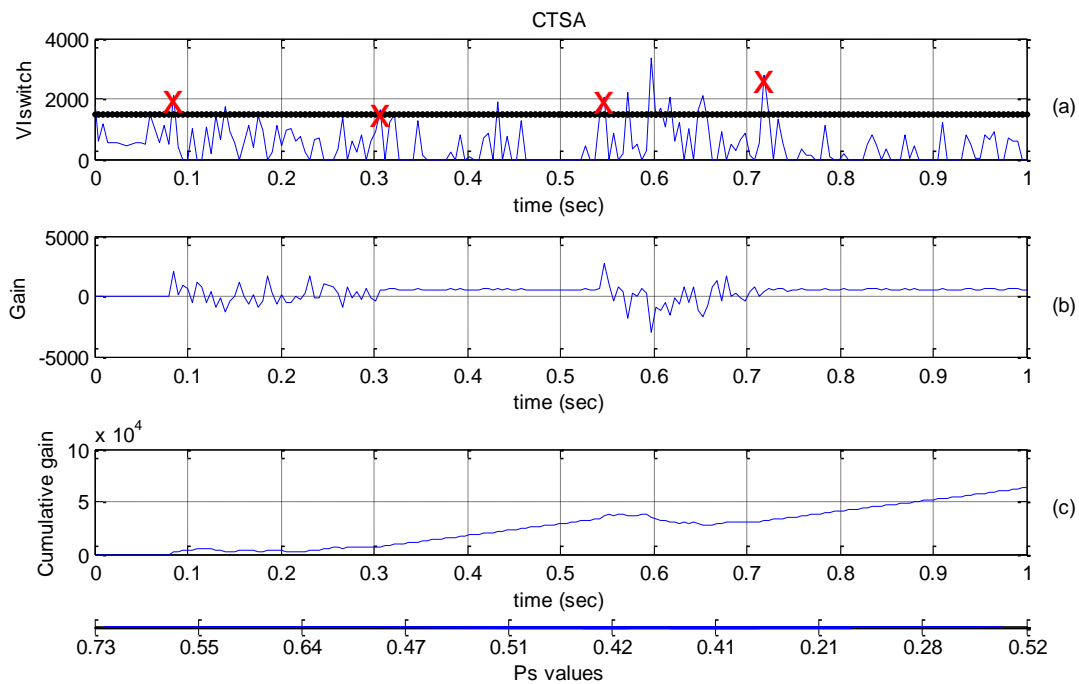
The designed numerical program simulates the operation of the combined human-robot system for target recognition tasks and calculates the objective function values for each operation made by the system. It implements the proposed switching algorithms and evaluates the  $V_{\text{Iswitch}}$  value for each image sample and the gain achieved according to the reference as described by each pseudocode. Representative graphs of the simulated results are presented in Figures 20, 21, 22 and 23 for normally distributed random values of target probability  $P_s$  from 0.1 to 0.9. The figures are divided into three graphs; (a), (b), (c). The  $V_{\text{Iswitch}}$  axis (y axis) in (a) corresponds to the aforementioned developed switching objective function score which evaluates the gain from switching the system to an optimal collaboration level according to equation 23. The  $V_{\text{Iswitch}}$  values on the graph were updated with each image sample. Figure (b) represents the gain in the objective function score ( $V_{\text{IS}}$ ) achieved by each system's operation (image sampling) compared to a particular collaboration level that was used as a reference according to equation 26. Figure (c) represents the cumulative gain for the entire operation of the system which is the sum of the gains in the second graph according to equation 27. The switch points are marked with 'X' on the upper graph. The switch points represent a collaboration level switch at a particular image sample. An axis with  $P_s$  values is illustrated at the bottom of the figures.

The following figures describe the operation of the switching algorithms in a representative case starting at the HR collaboration level. The starting collaboration level is used as a reference for calculating the gain achieved by the algorithms. The system objective function score of the reference is calculated as  $V_{\text{IS}_{\text{reference}}}$ . The gains achieved by the different algorithms are calculated relative to objective function score of a system that worked solely with HR collaboration level in this case. The figures present the dynamic switching operation

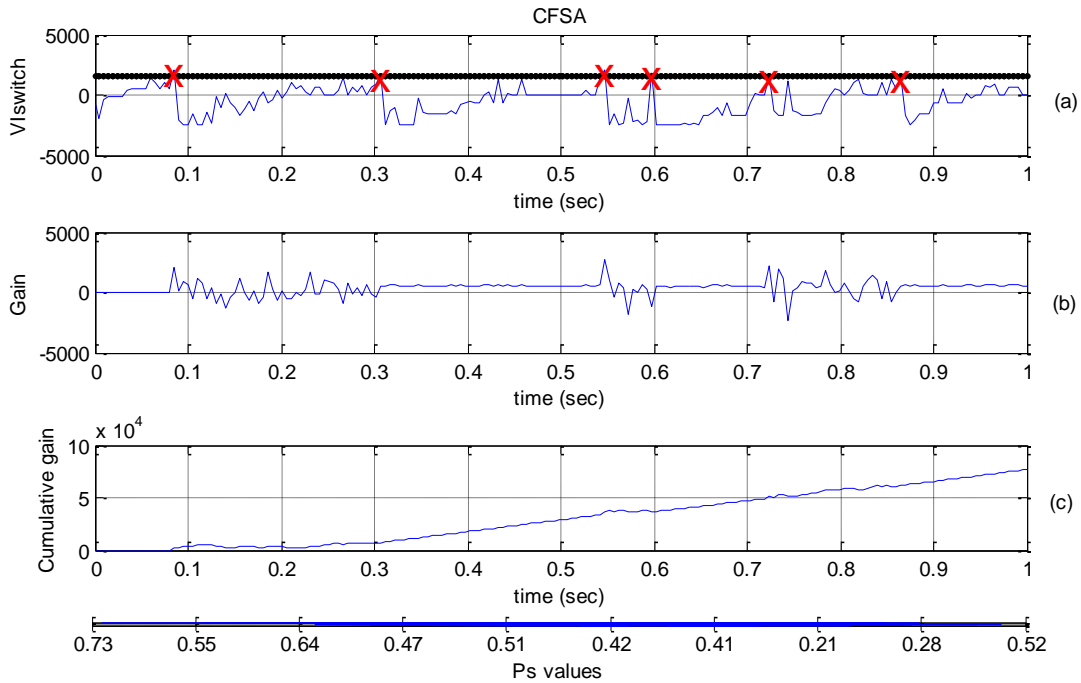
of each algorithm and the gain in objective function score which has been achieved by the algorithms during the simulation of the target recognition task of the system. The sequence and the values of  $P_s$  for all the algorithms were identical in this simulation execution.



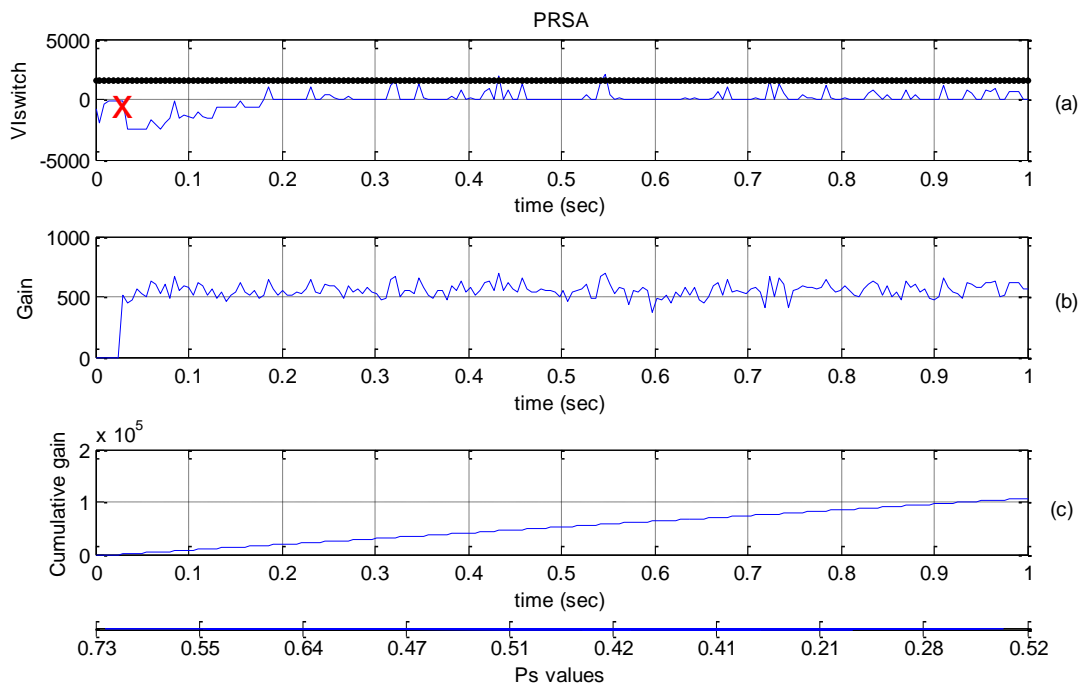
**Figure 20: The cost for non optimal work with switching in normally distributed random values of  $P_s$  – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure 21: The cost for non optimal work with switching in normally distributed random values of  $P_s$  – algorithm “CTSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure 22: The cost for non optimal work with switching in normally distributed random values of Ps – algorithm “CFSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure 23: The cost for non optimal work with switching in normally distributed random values of Ps – algorithm “PRSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

Figure 20 presents the results of a representative simulation of the “RLSA” algorithm operation. In this example the system made nine switches in deviations dictated by the switching frequency which represents the nominal time between switching operations,  $\Psi$ . Due to this limitation we can assume that the image sample rate is much higher than the switching rate. We can clearly see that many switches were performed for low  $V_{\text{Iswitch}}$  values and therefore had a little impact on the systems’ gain. The switching operation of this algorithm is purely constrained by the switching frequency, which makes it not so effective for high changing rates of the target probability (in this example). This phenomenon is illustrated in Figure 20 (b) which often receives negative values, which means that in some image samples the switching operation decreased the systems’ score. Nevertheless, in this example the algorithm resulted in improved overall system performance as indicated in Figure 20 (c). Figure 21 presents the results of a representative simulation of the “CTSA” algorithm operation. In this case another constrain is introduced by the threshold. The threshold value was set to 1500 units in this example. This algorithm switches the collaboration levels only when the  $V_{\text{Iswitch}}$  value is above the threshold. This results in fewer switches but on relatively high  $V_{\text{Iswitch}}$  values. Nevertheless this algorithm misses some high valued points for switching due to its constraint by the switching execution frequency. In Figure 21 (b) we can see fewer negative values and improved overall system performance compared to “RLSA” algorithm. Figure 22 presents the results of a representative simulation of the “CFSA” algorithm. This algorithm makes switches whenever  $V_{\text{Iswitch}}$  value is higher than a desired threshold with no switching frequency limitation except a penalty for breaking the frequency, according to equation 23. For example, after the ‘switch point’ in the current Figure there is a drop in  $V_{\text{Iswitch}}$  value below zero, which indicates the penalty for switching the system before the amount of time allowed by the switching frequency. When the switching objective function score becomes positive again then the time window of switching frequency is over and the controller can make a switch without any penalty. This strategy ensures switching the system in all high valued points as illustrated in Figure 22 (a), unlike in the previous algorithms, which eventually results in improved overall system performance. Figure 23 presents the results of a representative simulation of the “PRSA” algorithm. This algorithm makes switches by including predictions from past data. The controller switches the collaboration level to a level that was detected as the best collaboration level in the time window between the previous switch until the decision making. Therefore this algorithm is searching for consistency in the optimal collaboration level before making a switch. This algorithm also uses the strategy of “CFSA” algorithm of breaking the switching frequency by

paying a sufficient penalty. In Figure 23 (a) we see that only one switch operations were made, which is the result of the high inconsistency of the optimal collaboration levels in the time series due high deviations of the  $P_s$  values in the presented example. This algorithm has no negative values on the gain graph and received the best overall system performance score.

The numerical analyses of the proposed algorithms for other target probability distribution values are presented in Appendix II. Analysis for changes both in target probability and human sensitivity values and different frequency and  $P_s$  change values are presented in the Appendix II.

### 6.1.1 Summary of the results

Table 7 presents summary of the results of the numerical analysis for 200 independent simulations of each algorithm. The distributions of the target probability,  $P_s$ , simulate changes in the environmental conditions during system's work. The first column in Table 7 represents the algorithms, the second column represents starting collaboration level of each algorithm and the last four columns represent different target probability distributions. The first sub column of each distribution represents the gain in percentage for each algorithm relative to performance of a system without switching methodology, the second sub column represents the standard deviation of the results from 200 simulations and the third sub column represents the average number of switches performed in each case. The second and third columns in Table 7 represent random changes in  $P_s$  values and therefore, simulate drastic environmental changes in the system's working process. The uniform distribution of these random values presents an unpredictable and most radical target probability changes, and "PRSA" algorithm which is based partially on prediction, didn't performed with best results. The ascending and linear distributions represent gradually growing  $P_s$  values over time; consequently the simplest switching algorithm is sufficient to greatly improve the system performance by simply switch the collaboration level to optimal. In this case the more sophisticated algorithms do not perform well because of their additional set of conditions for switching, which makes it difficult for the system to pass them in such stable environment parameters.

The simpler "RLSA" algorithm is preferable in case of non radical changes of the parameters, because of its great results in improving the system performance in these cases and its relative low cost for implementation to the system due to its simplicity which requires less computational resources.

The “CFSA” and “PRSA” algorithms are preferable in cases that the system parameters change frequently and/or with high deviations. These algorithms were shown to give better results in improving system performance in these cases.

Algorithm “CTSA” should be considered for use when the system designer wants to implement an algorithm which allows customizations (threshold for switching) with relatively low computational cost.

**Table 7: Summary of the results for all distributions with 200 independent simulations with different starting conditions.**

Algorithm	Starting collaboration level	Linear distribution			Uniformly distributed random values			Normally distributed random values			Normally distributed random values sorted by ascending order		
		% Gain	Standard deviation	Number of switches	% Gain	Standard deviation	Number of switches	% Gain	Standard deviation	Number of switches	% Gain	Standard deviation	Number of switches
“RLSA”	<b>HOR</b>	99.99	0.01	2	2.37	13.27	18	3.42	25.32	19	99.99	0.01	2
	<b>HR</b>	98.21	1.32	3	43.39	6.56	18	53.87	6.54	18	96.51	1.01	3
	<b>R</b>	84.72	13.48	3	28.31	8.38	17	43.33	8.66	15	80.75	6.96	3
	<b>H</b>	93.61	6.05	3	79.81	2.58	16	86.75	3.03	21	92.81	2.87	3
“CTSA”	<b>HOR</b>	53.74	0.01	2	3.18	14.33	16	4.21	18.54	7	28.53	11.38	2
	<b>HR</b>	56.31	0.01	2	44.66	6.15	16	56.02	6.59	7	30.58	7.79	2
	<b>R</b>	57.69	14.16	3	31.52	8.47	17	49.29	10.14	6	52.19	10.11	3
	<b>H</b>	88.65	6.27	3	80.26	2.71	15	88.01	3.35	9	88.47	3.13	3
“CFSA”	<b>HOR</b>	53.74	0.01	2	7.82	11.84	23	4.56	16.93	7	28.53	11.38	2
	<b>HR</b>	56.31	0.01	2	49.41	6.11	22	57.72	6.21	9	30.58	7.79	2
	<b>R</b>	62.23	8.63	3	36.33	8.55	21	51.26	9.57	6	56.88	11.36	3
	<b>H</b>	94.39	0.01	3	85.83	1.61	20	91.95	1.49	8	95.04	0.77	3
“PRSA”	<b>HOR</b>	0.0 <sup>2</sup>	0	1	6.98	7.25	2	5.02	7.83	4	0.0 <sup>2</sup>	0	1
	<b>HR</b>	51.96	1.29	2	42.84	4.35	2	63.72	5.29	3	64.43	4.67	2
	<b>R</b>	29.79	8.64	2	26.81	11.92	2	54.28	9.77	4	44.82	13.44	2
	<b>H</b>	88.13	0.01	2	84.03	1.92	3	92.38	1.58	5	93.02	1.48	2

Each distribution results are further illustrated in appendix II.

<sup>2</sup> The “PRSA” algorithm didn’t make switches in these distributions of the target probability.

## 6.2 Analysis of the switching points

In this analysis the system was simulated to make switches in collaboration levels for normally distributed random values of target probability  $P_s$ . The simulation was conducted on the basis of the “RLSA” algorithm but without the constraint of the frequency of switching. The goal of this analysis is to learn about the amount of contribution of the discrete switch points to the system overall performance. Results are presented in Figure 24. The graphs present the number of switches that were made during the simulation vs. their contribution to the system performance in terms of  $V_{Iswitch}$  values. The figure illustrates that the majority of the switch points have poor potential to increase system performance and only few switch points have a great value on improving it<sup>3</sup>. Therefore, it is worthy to make switches in this few points that have high potential to increase system performance especially in a case of switching frequency limitation. This way better results could be achieved by switching the system less times. This strategy was implemented in “CTSA”, “CFSA” and “PRSA” algorithms by defining a certain threshold for switching. Note that algorithm “RLSA” does not have a constraint of a threshold for switching as was described earlier in order to maintain minimum limitations as possible.

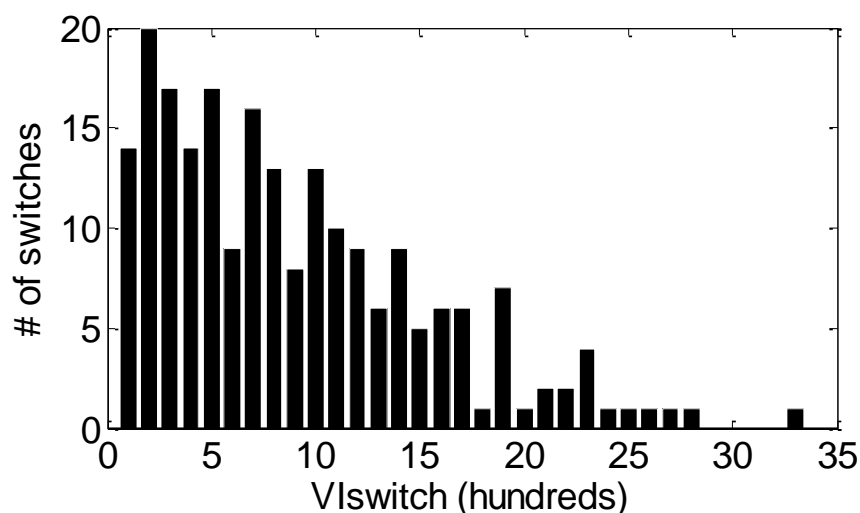


Figure 24: Graphical presentation of number of switches vs. switching gain in random distributed  $P_s$ .

<sup>3</sup> This finding doesn't mean that the proposed algorithm result in poor performance.

# 7 Conclusions and future research

## 7.1 Conclusions

A comprehensive process was designed and undertaken to develop and evaluate switching algorithms between different collaboration levels and the impact on the performance of an integrated human-robot system for target recognition tasks in different cases. It included the development of a switching objective function and four algorithms for switching. The algorithms were evaluated using numerical analyses. A logical controller with a dynamic control mechanism for an uncontrolled human-robot system that includes different collaboration levels for target recognition tasks in unstructured environments was designed. This controller allows to maintain maximum system performance despite possible deviations in the parameter values during task performance. Real-time dynamic switching of the collaboration levels was achieved by the algorithms implementation in a closed loop control process. The performance of the introduced switching methodology was simulated numerically for different parameters and conditions. The four algorithms present different methods for dynamic switching between collaboration levels. All the algorithms were proven by numerical simulations to increase system performance. Each algorithm is best suited for a different task performance and operating scenario as described in 6.1.1. System designers can choose what algorithm to implement for their system according to their specific environment and task, by the recommendations presented in this work.

These developments enable smooth real-time adaptation of the combined human-robot system to many possible changes of the conditions and parameters during system's task performance, like changes in the environment, human operator performance and robot performance. Additionally, it was shown to improve the overall system performance by the dynamic switching mechanism.

The main conclusions from this research were:

1. It is possible to automatically switch between different collaboration levels in a dynamic human – robot system.
2. In some cases dynamic switching of collaboration levels in a human – robot system increases the overall profit and system performance dramatically.
3. At non-controlled or non-automatic switching of the system, it is possible to have a decrease in the system's overall performance by switching it to 'bad' points by the



human operator. This will result in the trial to achieve local optimum and miss the global optimum as was found in the numerical analysis and presented in Appendix II.

4. Switching execution frequency has a great impact on dynamic switching performance, which can cause a decrease in the overall system performance in some cases.
5. Increase in the switching execution frequency value greatly improves the score achieved by the switching algorithms and increases system performance.
6. When a threshold for switching is applied, the limitation on frequency of the switches decreases the score achieved by the algorithm by not allowing it to switch at major points.
7. The prediction algorithm, “PRSA”, behaves well under most probabilities and greatly increases the system performance in most of the simulated scenarios.
8. All algorithms resulted in increased objective function score under radical cases which include uniformly and normally distributed random values of the system parameters.
9. The algorithms were shown to increase system’s overall performance by more than 90% in some cases.
10. It is worthwhile to have a dynamic switching frequency so as to enable adaptation to system parameter changes. This way the switching frequency will be equal or higher than the changes of the system parameters and the system’s performance could be improved by 100% as illustrated in Appendix II.

## **7.2 Future research**

Suggestions for future research:

1. Design a methodology for automatic selection of a best algorithm for a specific task. This may increase the performance of the controlled system and make it robust and suitable for many operating scenarios.
2. Design of another switching algorithm based on the “PRSA” algorithm that will use fuzzy logic theory for the switching operation. It should make a decision regarding the majority of optimal collaboration levels of the same type not by a crisp value as suggested in this work but by a fuzzy set. This might increase performance due to its flexibility.

3. Introduce the concept of reinforcement learning in order to train the controller and adapt its actions to the specific environment.
4. Formulate a presentation of the image sampling process as a stochastic Markov process with the parameters distribution as analyzed in this work. This will allow to calculate probabilities of the future states and to predict changes in the environment. Therefore, it may lead to a more efficient representation of the switching algorithms.
5. Find a correlation between the two systems constraints, the threshold and the switching frequency, and determine the influence of different threshold values on system performance. It is clear that the threshold value strongly depends on the switching frequency value and at high switching frequency values the threshold should be lower and vice versa.
6. Conduct an experiment for actual switching in a real system. This experiment should include a platform with robot or simulation of a robot with human participants. In the experiment all the proposed algorithms should be tested for switching the system. The designed experiment should validate the findings of the numerical simulation conducted in this thesis and provide experimental proof for the possibility of dynamic switching of a human-robot system in target recognition tasks. Based on the experiments values of parameters such as switching frequency, response times and the switching penalty can be determined.

## 8 Bibliography

1. Al-Jumaily, A.A.S. and Amin, S.H.M. 2000. Blending multi-behaviors of intelligent reactive navigation for legged walking robot in unstructured environment. Proceedings of TENCON 2: 297-302.
2. Ayanna M. Howard 2006. Role Allocation in Human-Robot Interaction Schemes for Mission Scenario Execution. Proceedings of the IEEE Int. Conf. on Robot and Automation Orlando, Florida. 3588-3594.
3. Bechar A. 2006. "Human – Robot Collaboration Methods for Target Recognition in Unstructured Environments", Ph.D. thesis, Ben-Gurion University of the Negev, Beer-Sheva, Israel.
4. Bechar A. and Edan Y. 2003. Human-robot collaboration for improved target recognition of agricultural robots, *Industrial Robot*, 30(5): 432-436.
5. Bechar A., Edan Y. and Meyer J. 2006. Optimal collaboration in human-robot target recognition systems. *IEEE Transactions on Systems, Man and Cybernetics*, 5: 4243-4248.
6. Bicho E., Mallet P. and Schoner G. 2000. Target representation on an autonomous vehicle with low-level sensors. *Int. Journal of Robotic Research* 19(5): 424-447.
7. Brown C.D. & Davis H.T. (2006). Receiver Operating Characteristics Curves and Related Decision Measures: A Tutorial. *Chemometrics and Intelligent Laboratory Systems* 80(1): 24-38.
8. Chang Y.C., Song G.S. and Hsu S.K. 1998. Automatic extraction of ridge and valley axes using the profile recognition and polygon-breaking algorithm. *Computers & Geosciences* 24(1): 83-93.
9. Dorf R. 1989. *Modern control systems*. Addison-Wesley, Reading, Mass., 5<sup>th</sup> edition.
10. Elkind, J.I., Sprague, L.T. 1960. Transmission of Information in Simple Manual Control Systems. *IRE Transactions on Human Factors in Electronics*, HFE-2(1): 58-60.
11. Everett S. and Dubey R. 1998. Human-Machine Cooperative Telerobotics Using Uncertain Sensor and Model Data. Proceedings of Intl. Conf. on Robotics and Automation. Leuven, Belgium. 1615-1622.

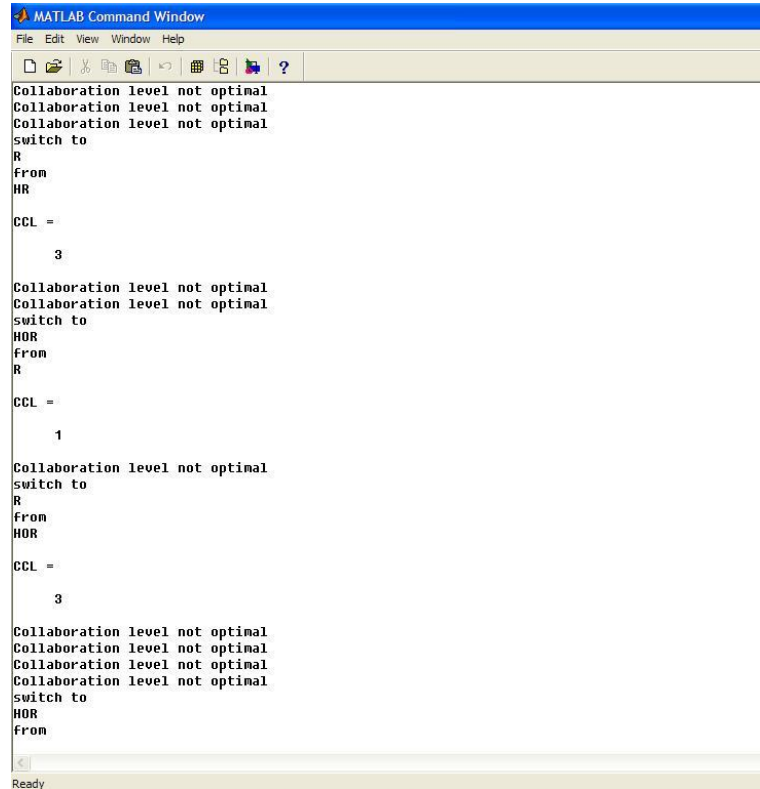
12. Fletcher L., Loy G., Barnes N. and Zelinsky A. 2005. Correlating driver gaze with the road scene for driver assistance systems. *Robotics and Autonomous Systems* 52: 71–84.
13. Green D.M., Swets, J.A. 1966. *Signal detection theory and psychophysics*. New York Wiley.
14. Goodwin G.C. 1984. *Adaptive filtering prediction and control*. Prentice-Hall, Inc. 1-5.
15. Goodwin G.C., Stefan F.G. and Mario E.S. 2001. *Control system design*. Prentice-Hall, Inc. 38-46.
16. Heeger D. 1997. *Signal detection theory*.  
<http://www.cns.nyu.edu/~david/handouts/sdt/sdt.html>.
17. Ivanisevic I. and Lumelsky V. 1997. *A Human - Machine Interface for Teleoperation of Arm Manipulators in a Complex Environment*. Tech. Report RL-97006. Robotics Lab, University of Wisconsin-Madison, Madison, Wisconsin, USA.
18. McRuer, D.T. and Jex, H.R., 1967. A review of quasi-linear pilot models. *IEEE Transactions on Human Factors in Electronics* HFE-8(3) 231–249.
19. Murphy R.R. 2000. *Introduction to AI robotics*. The MIT press.
20. Ng K.C. and Trivedi M.M. 1998. A Neuro-Fuzzy Controller for Mobile Robot Navigation and Multirobot Convoying. *IEEE Trans. Systems, Man and Cybernetics* 28(6): 829-840.
21. Oren Y. (2007). “Performance Analysis of Human-Robot Collaboration in Target Recognition Tasks”. B.Sc dissertation, Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel.
22. Parasuraman R., Sheridan T.B. and Wickens C.D. 2000. A Model for Types and Levels of Human Interaction with Automation. *IEEE Trans. Systems, Man and Cybernetics* 30(3): 286-197.
23. Penin L.F., Aracil R., Ferre M., Pinto E., Hernando M. and Barrientos A. 1998. Telerobotic system for live power lines maintenance: ROBTET. *Proceedings of Intl. Conf. on Robotics and Automation*. Leuven, Belgium. 2210-2115.
24. Pook P.K. and Ballard D.H. 1996. Deictic human/robot interaction. *Robotics and Autonomous Systems* 18(1-2): 259-269.
25. Poulton, E.C. (1974) *Tracking skill and manual control*. New York: Academic. 166.
26. Rodriguez G. and Weisbin C.R. 2003. A New Method to Evaluate Human-Robot System Performance. *Autonomous Robots* 14: 165–178.

27. Rucci M., Edelman G.M. and Wray J. 1999. Adaptation of Orienting Behavior: From the Barn Owl to a Robotic System. *IEEE Trans. Robot and Automatomation* 15(1): 96–110.
28. Salter T. Dautenhahn K. and Boekhorst R. 2006. Learning about natural human–robot interaction styles. *Robotics and Autonomous Systems* 54: 127–134.
29. Sanders M.S. and McCormick E.J. 1993. *Human factors in engineering and design*. McGraw Hill, Inc.
30. Scholtz J. 2002, *Theory and Evaluation of human robot interaction*. National institute of standards and technology. Washington DC, USA.
31. Sheridan T.B. 1976. *Monitoring Behavior and Supervisory Control*. New York. Plenum.
32. Sheridan T.B. 1992. *Telerobotics, Automation, and Supervisory Control*. Cambridge, MA. MIT Press.
33. Steinfeld A. 2004. Interface lessons for fully and semi-autonomous mobile robots. *Proceedings of Intl. Conf. on Robotics and Automation*. 3: 2752 – 2757.
34. Swets J.A. Dawes R.M. and Monahan J. 2000. Better Decisions through Science. *Scientific American* :82-87
35. Van Erp J.B.F. and Van Veen H.A.H.C. 2004. Vibrotactile in-vehicle navigation system. *Transportation Research Part F: Traffic Psychology and Behavior* 7(4-5): 247-256.
36. Uttal, William R. 1989. Teleoperators. *Scientific American*. 124-129.
37. Wampler C. 1990. Teleoperators, supervisory control. *Concise international encyclopedia of robotics: applications and automation*. R. C. Dorf, editor in chief. John Wiley and Sons, Inc. 997.
38. Wickens T.D. 2002. *Elementary signal detection theory*. Oxford University Press.
39. Ye Y.M. and Tsotsos J.K. 1999. Sensor planning for 3D object search. *Computer Vision and Image Understanding*. 73(2): 145-168.

## **9 Appendixes**

## Appendix I: The simulation program

The simulation uses Matlab user interface to display the controller messages and operations. It includes three messages: the first is “collaboration level not optimal” in case of the current collaboration level is not optimal and the controller cannot switch it due to switching frequency limitation, the second is “switch to X from Y” when switching of the collaboration level is performed by the controller; where X is the collaboration level from which the controller switches the system and Y is the collaboration level the controller switching to. The third message is a numerical representation of the current collaboration level. This message displayed only when collaboration level switch is performed.



```
MATLAB Command Window
File Edit View Window Help
Collaboration level not optimal
Collaboration level not optimal
Collaboration level not optimal
switch to
R
from
HR
CCL =
    3
Collaboration level not optimal
Collaboration level not optimal
switch to
HOR
from
R
CCL =
    1
Collaboration level not optimal
switch to
R
from
HOR
CCL =
    3
Collaboration level not optimal
Collaboration level not optimal
Collaboration level not optimal
Collaboration level not optimal
switch to
HOR
from
Ready
```

Figure A-1: Matlab simulation of the control system

## Appendix II: Additional numerical analysis results

### Linear distribution Ps from 0.1 to 1

Here represented the graphs for the simulation result of linear distribution of target probability Ps. For each of four different starting collaboration levels the objective function score is presented and the best collaboration level determined and marked for each Ps value. In Figure A-2 the initial best collaboration level is HOR and then changes to R (marked in black). In Figure A-3 we will mark the switch points where the best collaboration level changes. In this case the starting collaboration level was HR, therefore at the beginning of the simulation the controller switches the best CL to HOR and then to R (marked with red arrow and 'switch point'). This is done by the algorithm "RLSA" methodology. In Figure A-4 presented the switching objective function gain for each starting collaboration level. In this Figure we see that H and HR collaboration levels are never optimal and that HOR is optimal at first and then R level becomes optimal. The  $V_{Iswitch}$  axis indicates the gain for each moment if the controller made a switch to an optimal collaboration level. For example, if the starting collaboration level was R than at Ps=0.1 the switching objective function would receive 2850 units for switching it to the best collaboration level (HOR in this case). At Ps=0.8 the switching objective function will receive no gain for switching the R level because it is already optimal. In other words Figure A-4 describes algorithm which makes no switching and therefore illustrates the loose in the global objective function that can be described as the cost for non optimal work for each collaboration level. The larger the gain of switching objective function the worthier the switching operation of the controller and therefore the cost for non optimal work is larger.

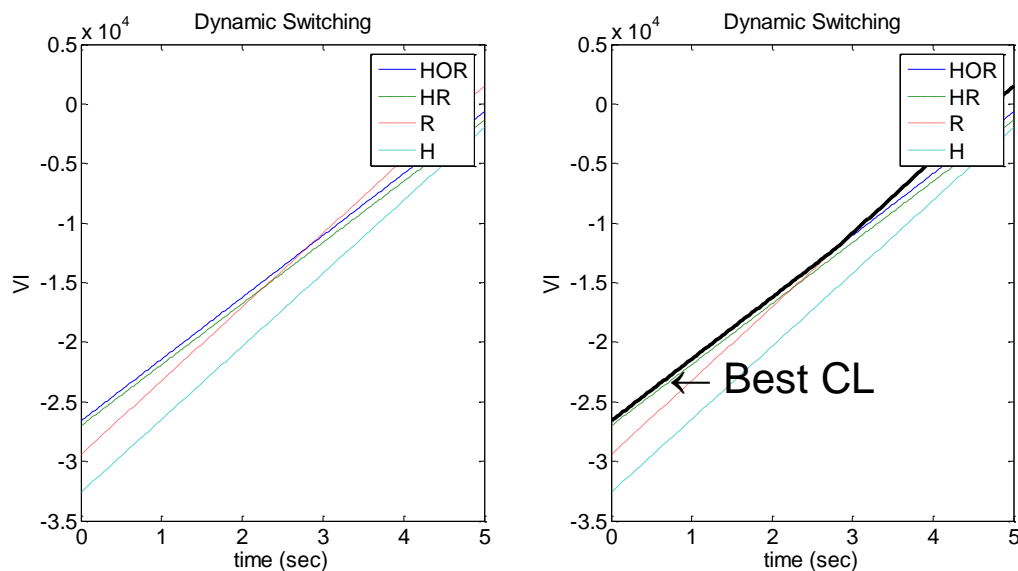
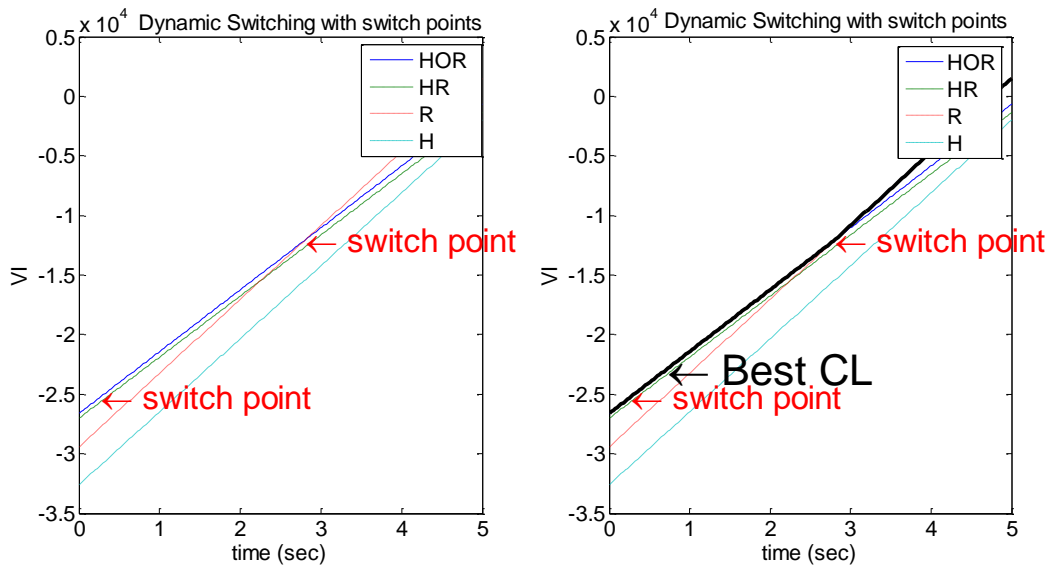
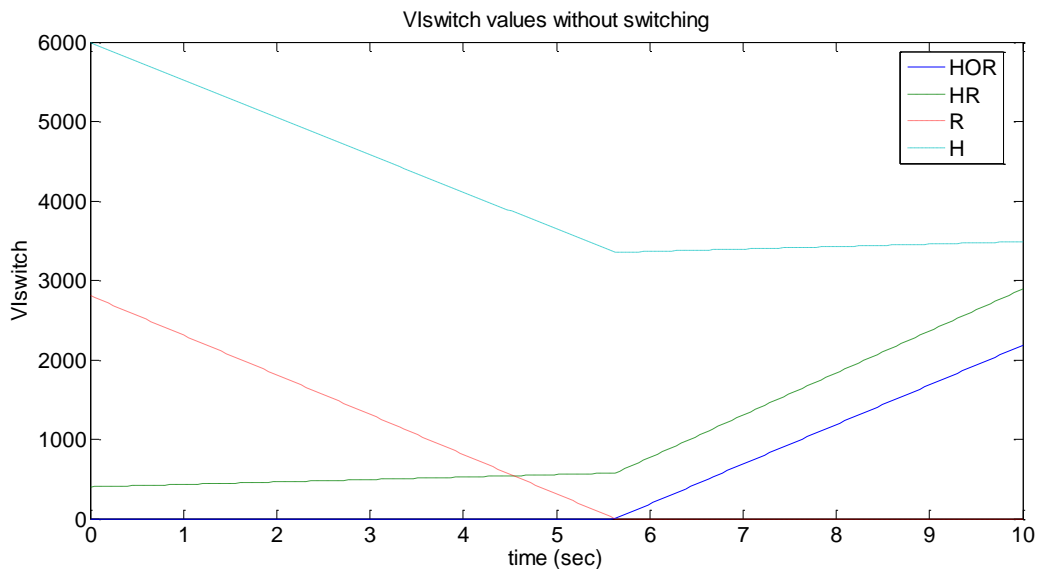


Figure A-2: Objective function scores for different collaboration levels in linear Ps distribution. The left graph represents the VI score for all the collaboration system operation. The right graph represents the best collaboration levels (marked in black).





**Figure A-3: Objective function scores for different collaboration levels in linear Ps distribution with switch points marked. The left graph represents the VI score for all the collaboration levels during system operation. The right graph represents the best collaboration levels (marked in black).**

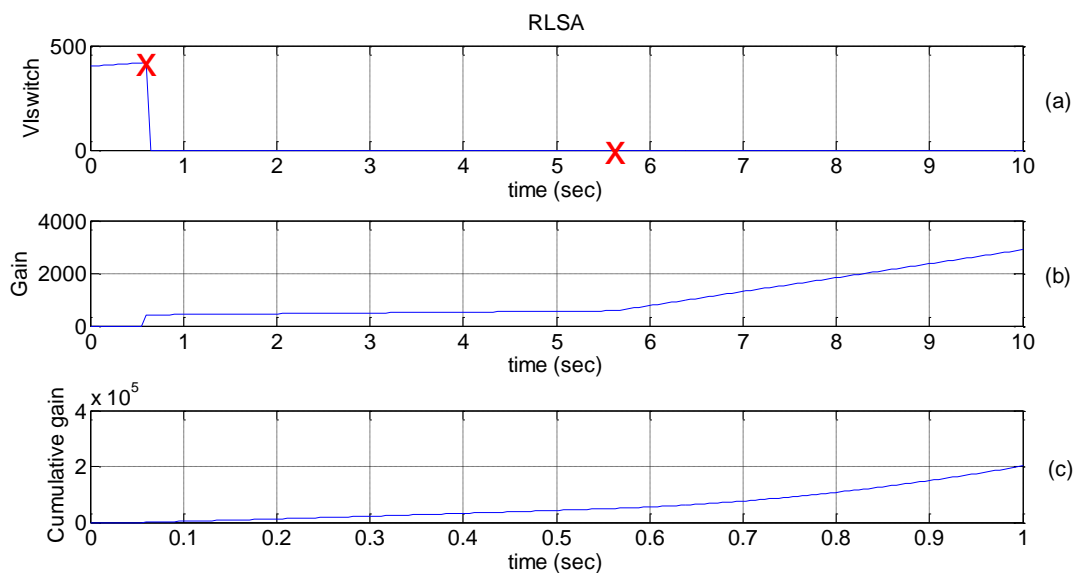


**Figure A-4: The cost for non optimal work (without switching) in linear Ps distribution, for all collaboration levels.**

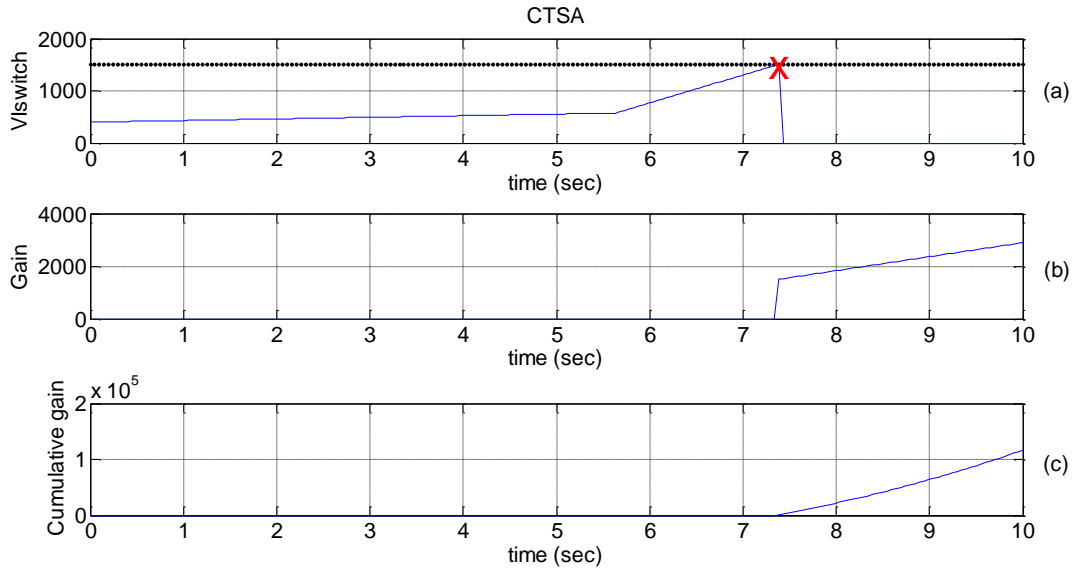
The following Figures describe the operation of the switching algorithms in a representative case of HR starting collaboration level. The gains achieved by the different algorithms are calculated relative to that starting collaboration level as if the system performs no switching at all during its work.

Figure A-5 illustrates the switching operation of “RLSA” algorithm. This algorithm makes switches whenever the switching objective function gain is positive or accordingly whenever the global objective function will make any benefit from the switch with switching frequency. The switch points are marked with X on the upper graph. The mid graph shows the gain in the global objective function score from each frame analyzed relative to score achieved by initial collaboration level. The lower graph shows the cumulative gain, which is the sum of the gains in the mid graph. Figure A-6 illustrates the switching operation of “CTSA” algorithm. This algorithm makes switches whenever the switching objective function gain is

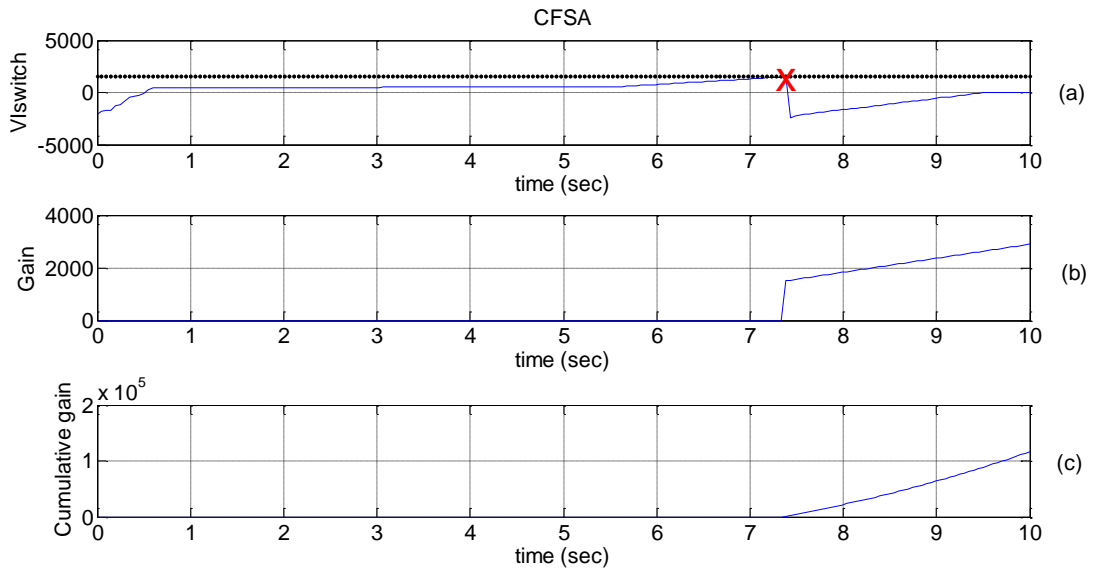
greater than a desired threshold (1500 in this example). Figure A-7 illustrates the switching operation of “CFSA” algorithm. This algorithm makes switches whenever the switching objective function gain is greater than a desired threshold with no switching frequency limitation except penalty for breaking the frequency. For example: after the ‘switch point’ in current Figure there is a drop in switching objective function score below zero, which indicates the penalty for switching before the amount of time allowed by the switching frequency. When the switching objective function score becomes positive again then the time window of switching frequency is over and the controller can make a switch without any penalty. Figure A-8 illustrates the switching operation of “PRSA” algorithm. This algorithm makes switches by prediction from past data. The controller switches the collaboration level to a new one when it was the best collaboration level most of the time in the time window between previous switch to present time. This algorithm also uses the strategy of “CFSA” algorithm of breaking the switching frequency by paying a sufficient cost.



**Figure A-5: The cost for non optimal work with switching in linear  $P_s$  distribution – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-6: The cost for non optimal work with switching in linear Ps distribution – algorithm “CTSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-7: The cost for non optimal work with switching in linear Ps distribution – algorithm “CFSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

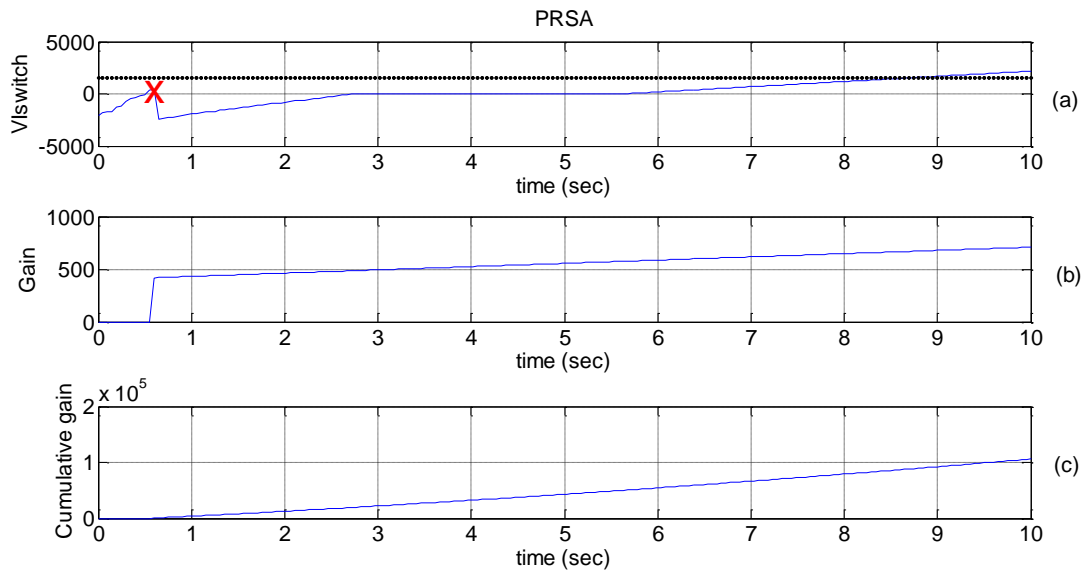


Figure A-8: The cost for non optimal work with switching in linear  $P_s$  distribution – algorithm “PRSA”. (a) represents the dynamic changes in  $V_{switch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.

Figure A-9 summaries all the gains for different starting collaboration levels received by the switching objective function score gained from the switching operation for different starting collaboration levels. The gains are in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.

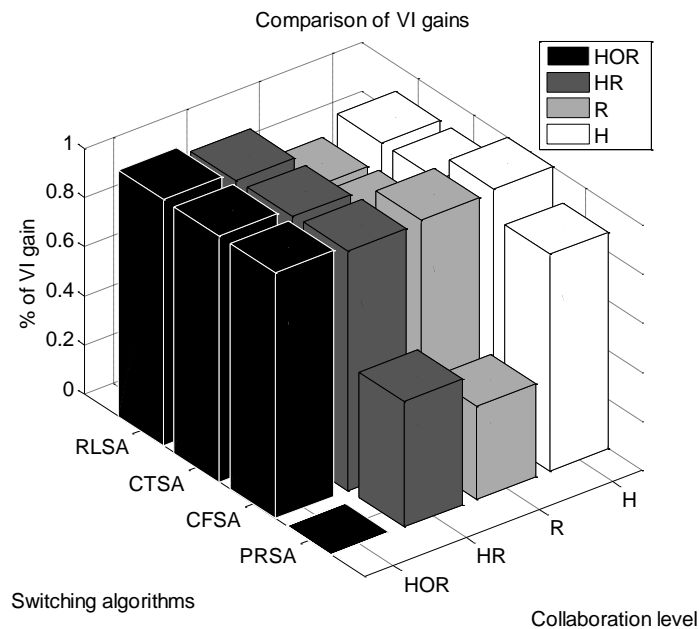


Figure A-9: Summary of all algorithm gains for different starting collaboration levels in linear  $P_s$  distribution in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.

### Uniformly distributed random values of Ps from 0.1 to 1

Here represented the graphs for the simulation result of uniformly random distribution of target probability Ps. Figure A-10 represents values of  $V_{Iswitch}$  for each of four different starting collaboration levels.

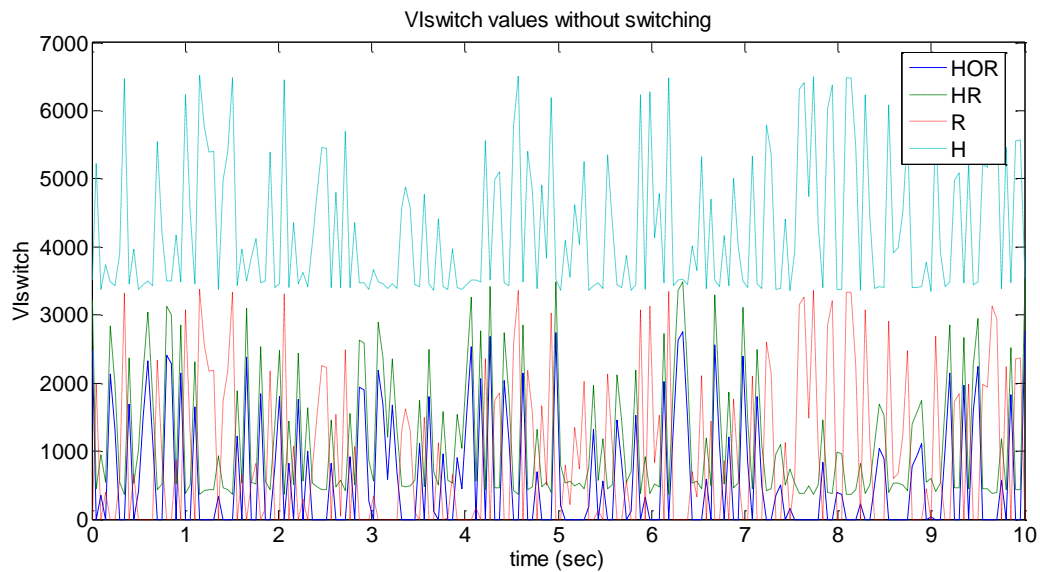


Figure A-10: The cost for non optimal work (without switching) in uniformly random Ps distribution.

Graphical representation of “RLSA”, “CTSA”, “CFSA” and “PRSA” algorithms performance is shown below. The graphs present the dynamic switching operation of each algorithm and the gain in global objective function score which has been achieved by the algorithms during the simulation of the target recognition task of the system.

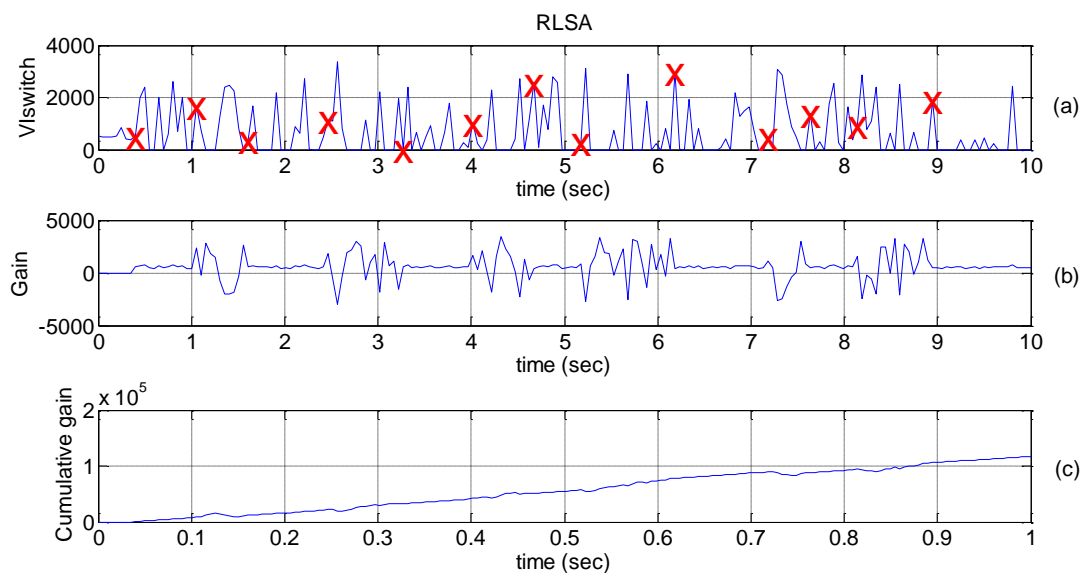
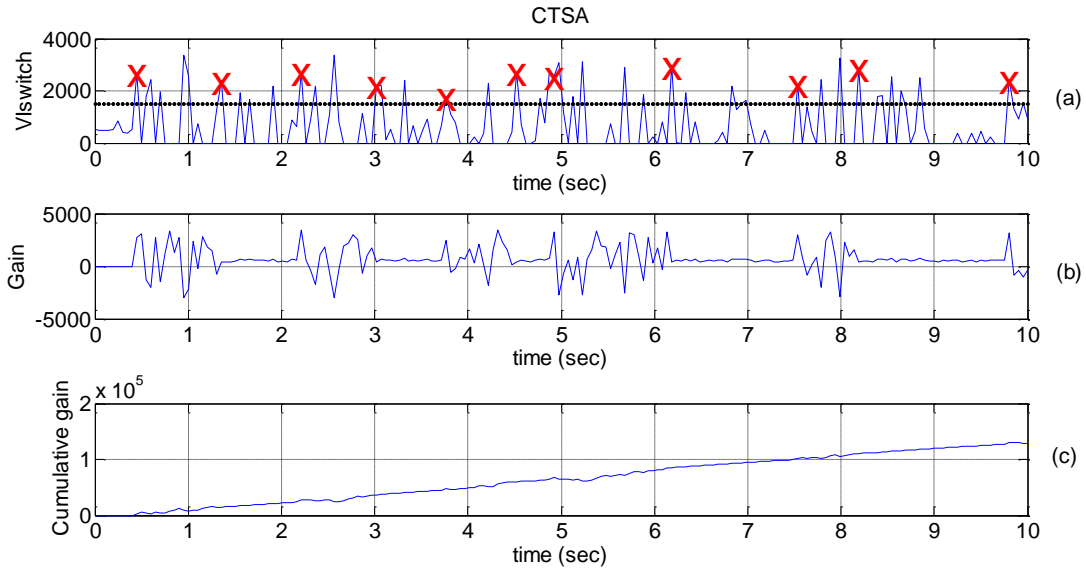
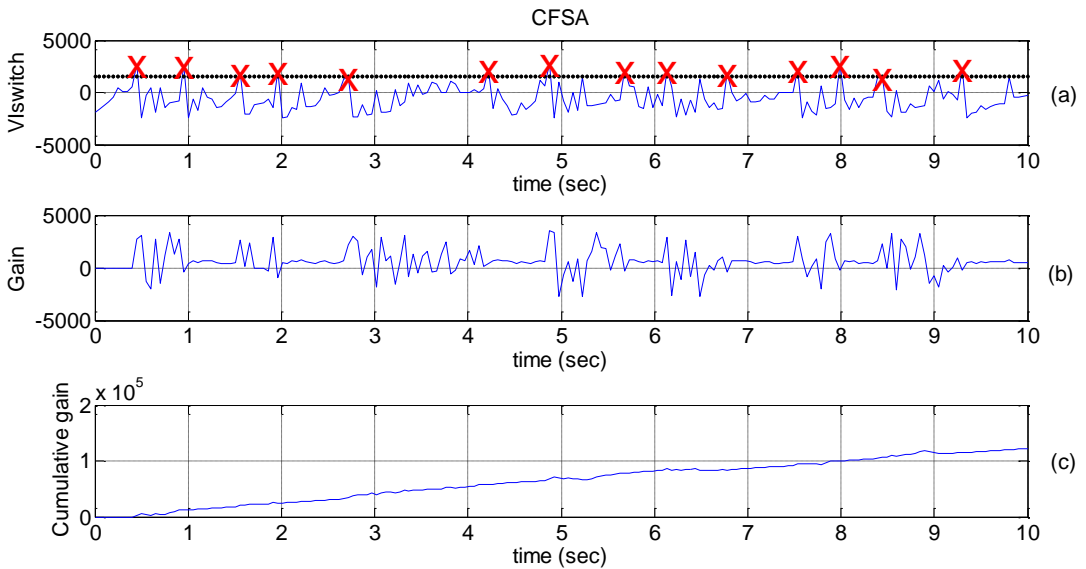


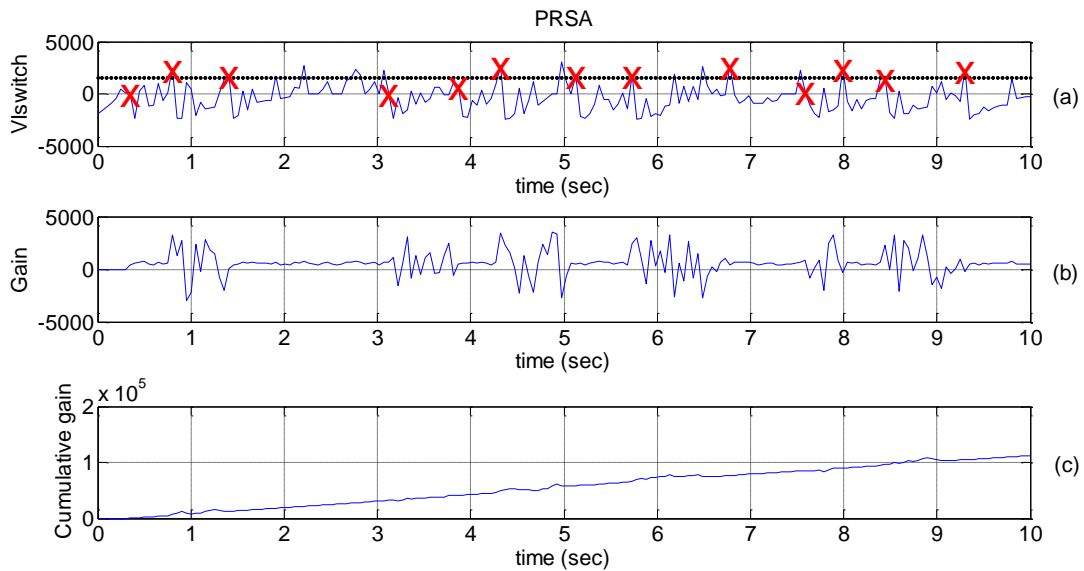
Figure A-11: The cost for non optimal work with switching in uniformly random Ps distribution – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) graph represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.



**Figure A-12: The cost for non optimal work with switching in uniformly random  $P_s$  distribution – algorithm “CTSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

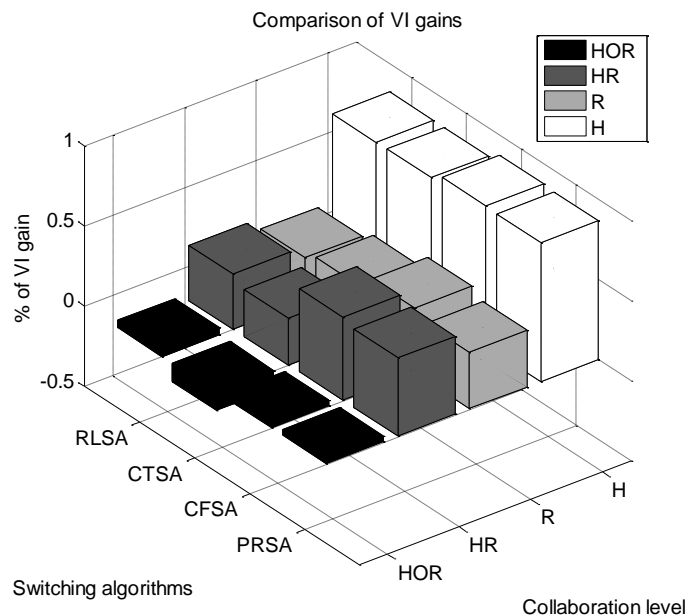


**Figure A-13: The cost for non optimal work with switching in uniformly random  $P_s$  distribution – algorithm “CFSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-14: The cost for non optimal work with switching in uniformly random  $P_s$  distribution – algorithm “PRSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

Figure A-15 presents a comparison of the gains in global objective function score for each algorithm for different starting collaboration levels. The gains are in percentage in scale of 0 to 1 (100%) relative to case with no switching at all. The negative values indicate negative effect of the dynamic switching operation to the system that results in decreasing the objective function score.



**Figure A-15: Summary of all algorithm gains for different starting collaboration levels in uniformly random  $P_s$  distribution in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.**

## Normally distributed random values of Ps from 0.1 to 1

Here represented the graphs for the simulation result of random normal distribution of target probability Ps. Ps values are normally distributed with mean 0.5 and standard deviation 0.5. Figure A-16 represents values of  $V_{Iswitch}$  for each of four different starting collaboration levels.

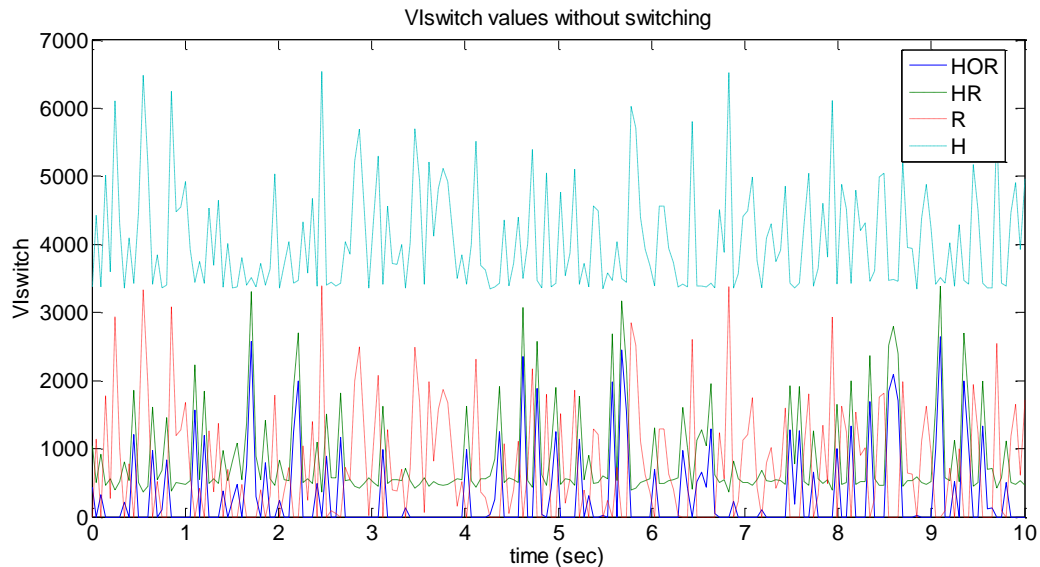


Figure A-16: The cost for non optimal work (without switching) in random normal Ps distribution.

Graphical representation of “RLSA”, “CTSA”, “CFSA” and “PRSA” algorithms performance is shown below. The graphs present the dynamic switching operation of each algorithm and the gain in global objective function score which has been achieved by the algorithms during the simulation of the target recognition task of the system.

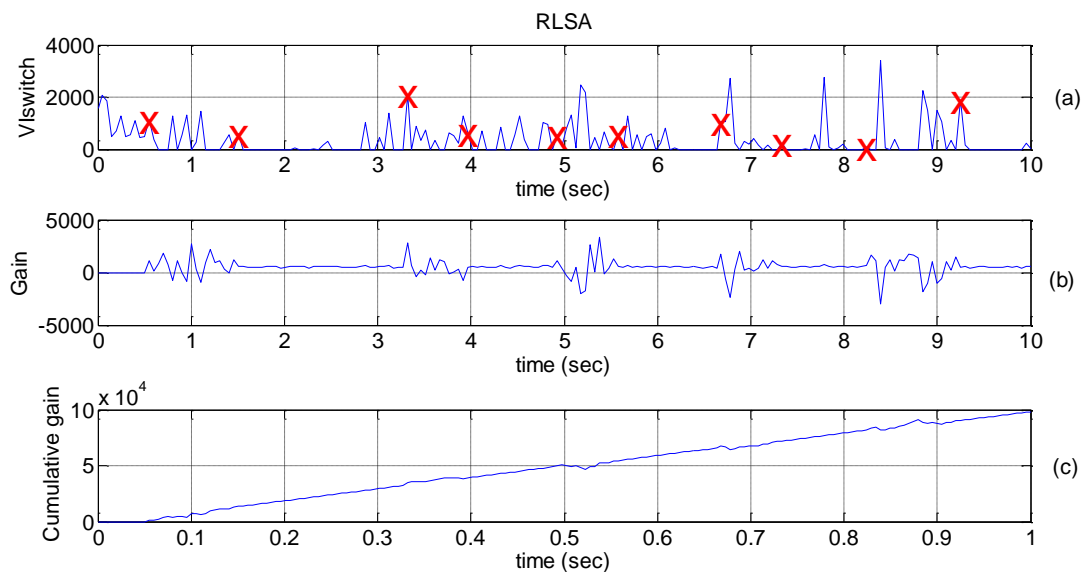
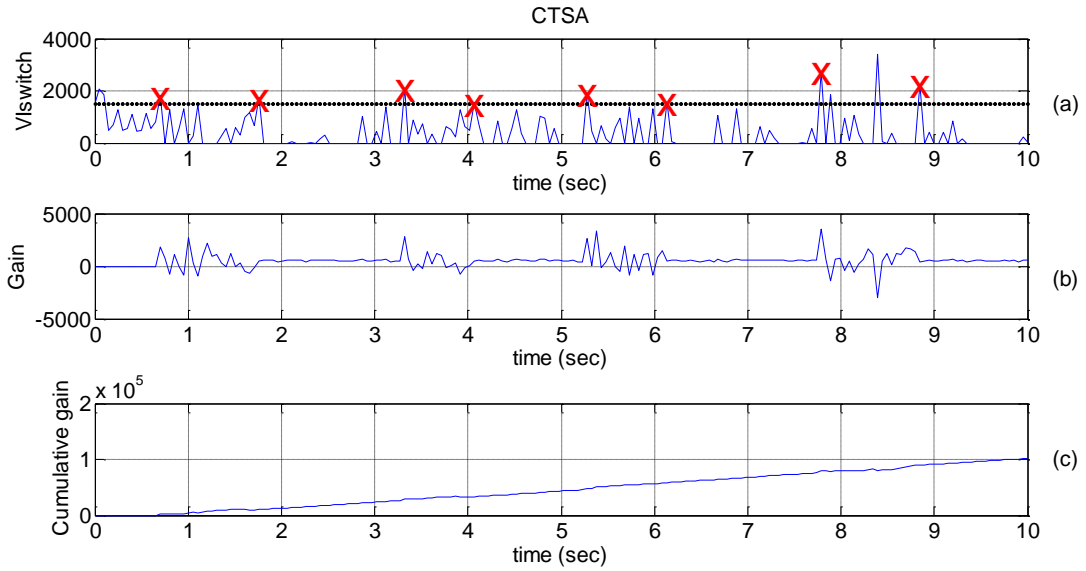
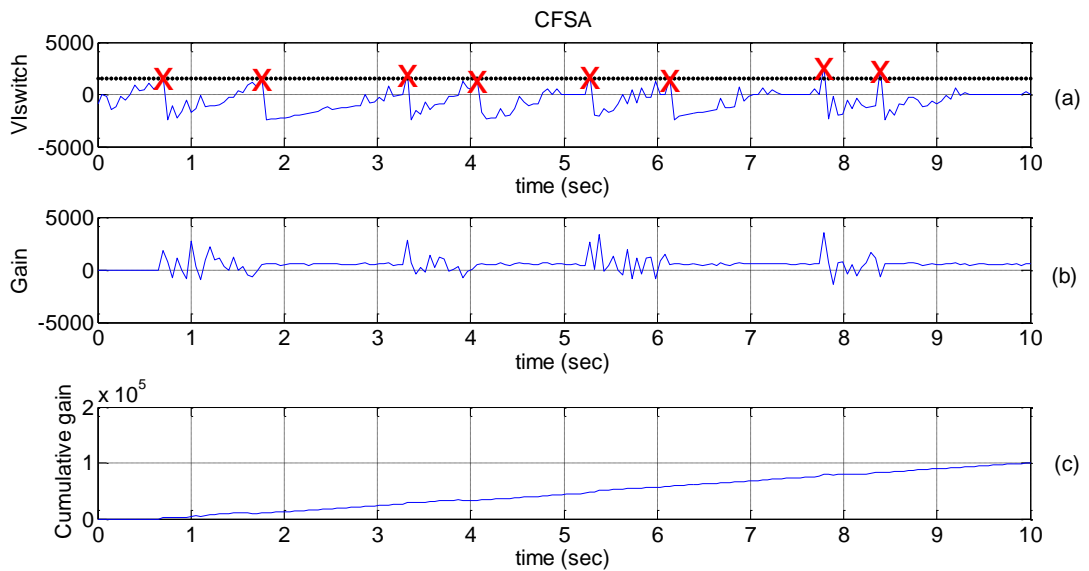


Figure A-17: The cost for non optimal work with switching in random normal Ps distribution – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.

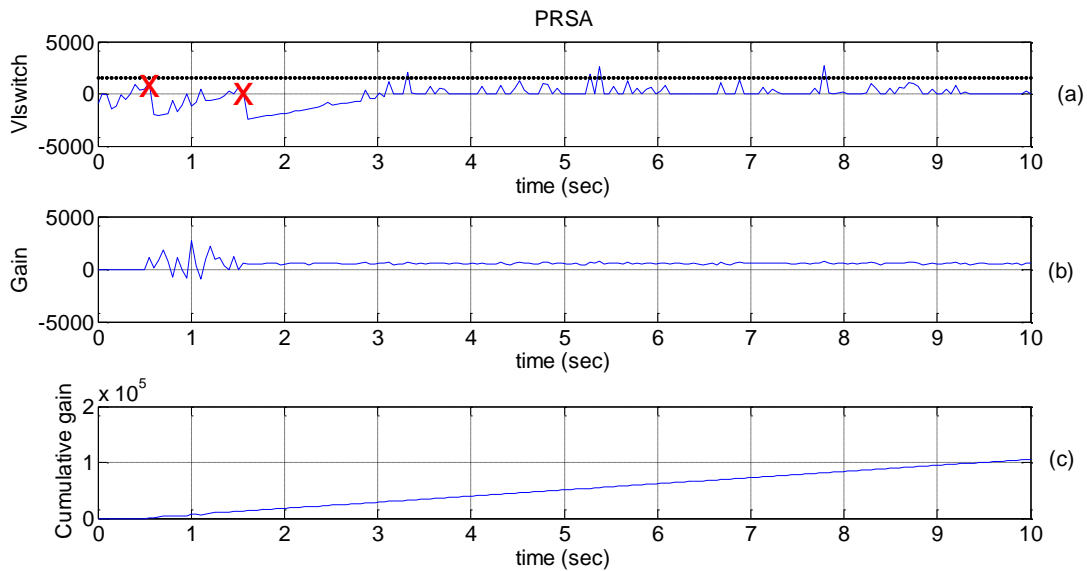




**Figure A-18: The cost for non optimal work with switching in random normal  $P_s$  distribution – algorithm “CTSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

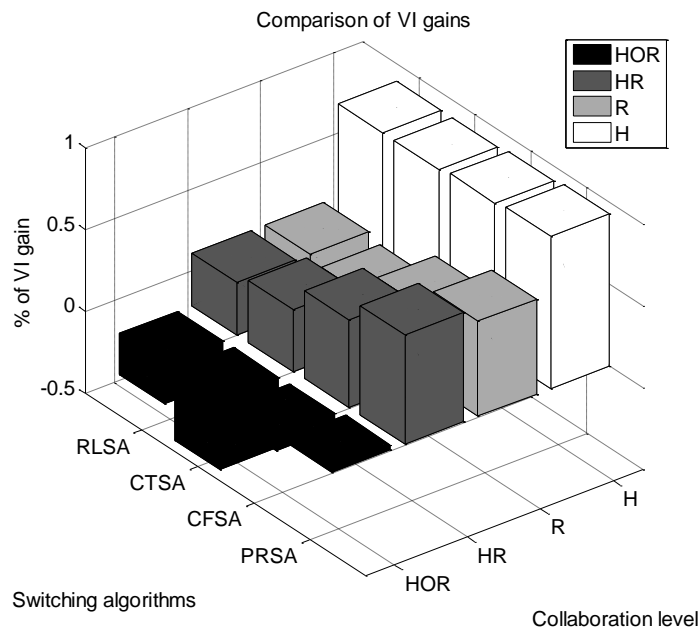


**Figure A-19: The cost for non optimal work with switching in random normal  $P_s$  distribution – algorithm “CFSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-20: The cost for non optimal work with switching in random normal  $P_s$  distribution – algorithm “PRSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

Figure A-21 presents a comparison of the gains in global objective function score for each algorithm for different starting collaboration levels. The gains are in percentage in scale of 0 to 1 (100%) relative to case with no switching at all. The negative values indicate negative effect of the dynamic switching operation to the system that results in decreasing the objective function score.



**Figure A-21: Summary of all algorithm gains for different starting collaboration levels in random normal  $P_s$  distribution in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.**

### Ascending normal distribution Ps from 0.1 to 1

Here represented the graphs for the simulation result of random normal distribution in ascending order of target probability Ps. The Ps values are normally distributed with mean 0.5 and standard deviation 0.5.

Figure A-22 represents values of  $V_{Iswitch}$  for each of four different starting collaboration levels.

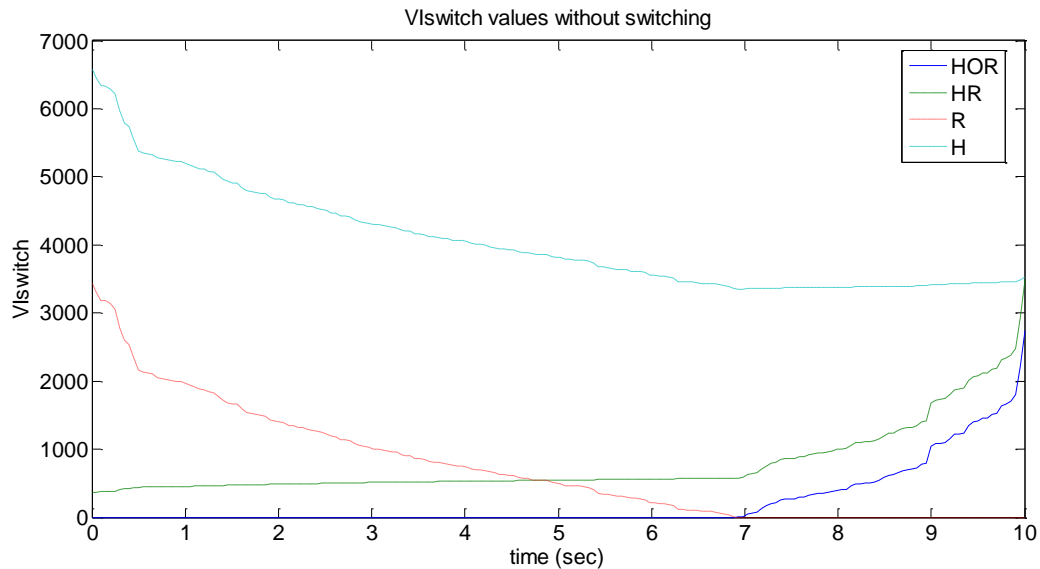


Figure A-22: The cost for non optimal work (without switching) in ascending normal Ps distribution.

Graphical representation of “RLSA”, “CTSA”, “CFSA” and “PRSA” algorithms performance is shown below. The graphs present the dynamic switching operation of each algorithm and the gain in global objective function score which has been achieved by the algorithms during the simulation of the target recognition task of the system.

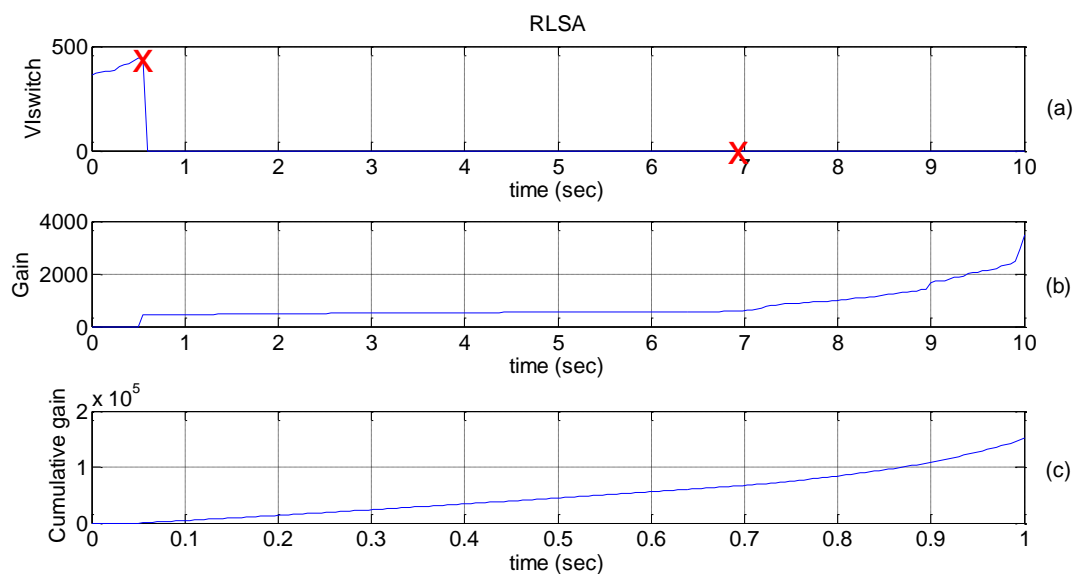
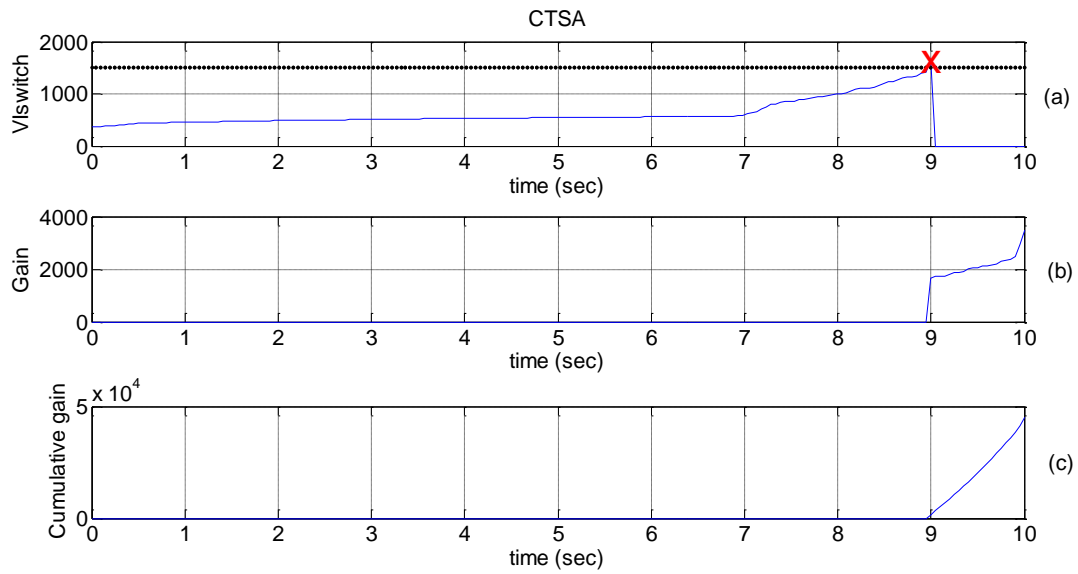
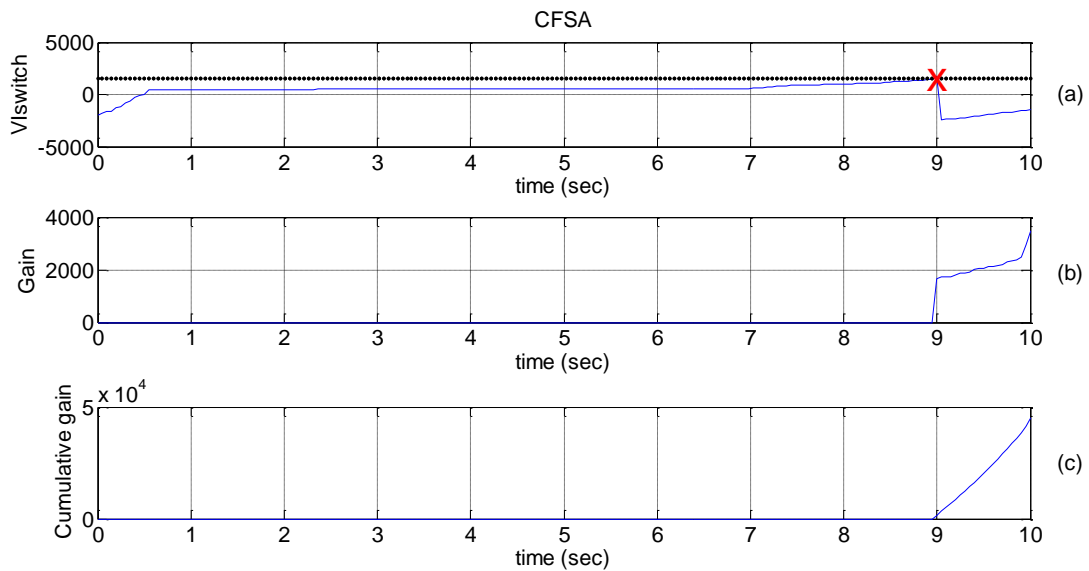


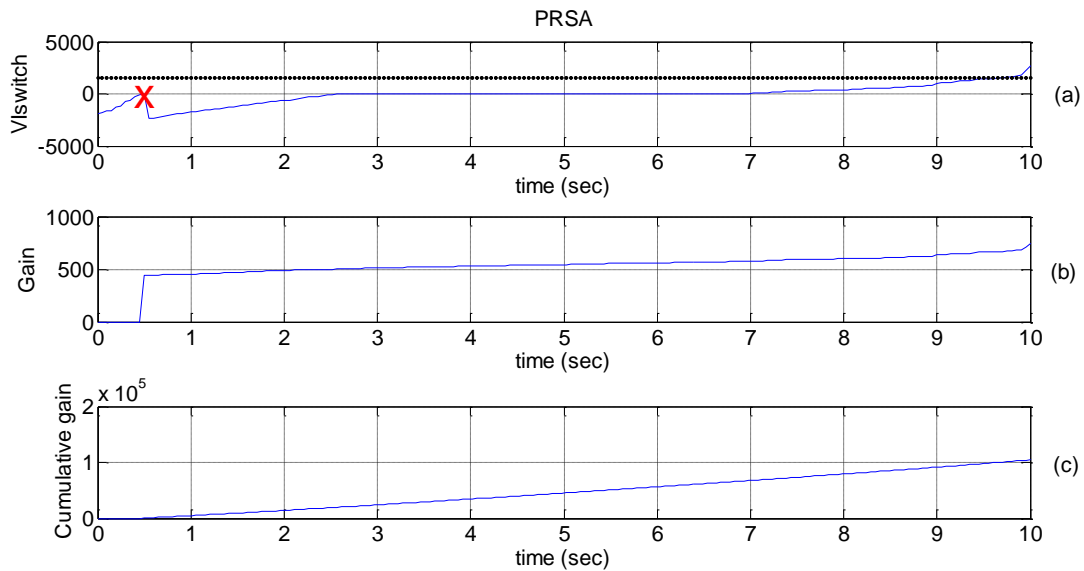
Figure A-23: The cost for non optimal work with switching in ascending normal Ps distribution – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.



**Figure A-24: The cost for non optimal work with switching in ascending normal Ps distribution – algorithm “CTSA”. (a) represents the dynamic changes in  $V_{switch}$  values over time and dynamic switching operation, (b) graph represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

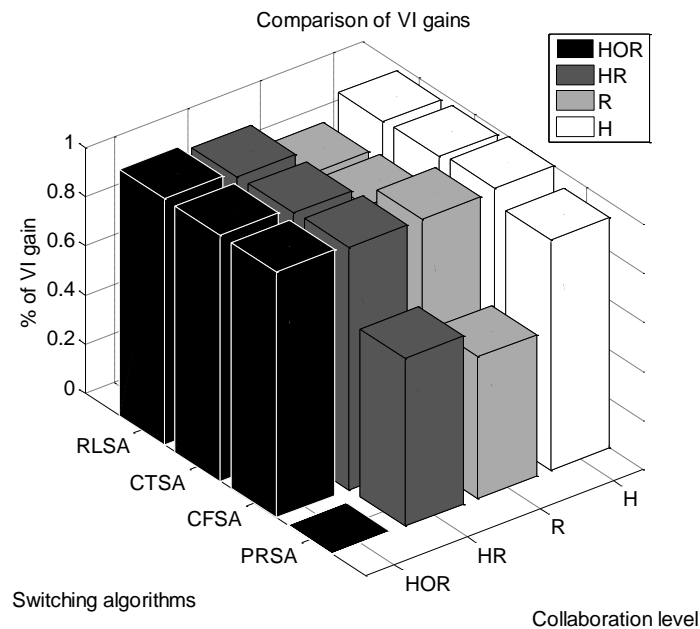


**Figure A-25: The cost for non optimal work with switching in ascending normal Ps distribution – algorithm “CFSA”. (a) represents the dynamic changes in  $V_{switch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-26: The cost for non optimal work with switching in ascending normal Ps distribution – algorithm “PRSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

Figure A-27 presents a comparison of the gains in global objective function score for each algorithm for different starting collaboration levels. The gains are in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.



**Figure A-27: Summary of all algorithm gains for different starting collaboration levels in ascending normal Ps distribution in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.**

### Numerical analysis of algorithms operation in Ps domain

In this section an analysis of the algorithms for Ps domain will be discussed. The algorithms were executed for normally distributed random values of target probability Ps. This analysis made by transform of the time domain of system's task performance to the target probability domain. The aim of this analysis is to present the switching operation of the proposed algorithms according to target probability values and not as a time series. This analysis can reveal switching at which Ps values have high potential increasing the systems performance. The simulation parameters are summarized in Table A-1:

**Table A-1: Summary of simulation parameters**

Parameter	Value
N	1000
$V_H$	10
$V_{AR}$	-1
$V_{CR}$	3
$V_M$	5
Ps	ranged from 0.1 to 0.9
$V_C$	-2
$V_t$	-2000 hr <sup>-1</sup>
decision time, $t_D$	5 s/object
motoric time, $t_M$	2 s/(detected object)
robot time, $t_r$	0.01 s/object
Robot sensitivity, $d'_r$	0.3
Human sensitivity, $d'_h$	0.7
System response time, $t_{response}$	5 sec
Frequency time, $\Psi$	0.000001 sec
Threshold	1500

Presentation of "RLSA" algorithm (Figure A-28): the system performs many switches at points that have poor  $V_{Iswitch}$  value. It is visible that the best values for switching operation are at the two edges of the Ps scale, while on the middle there are values that have poor  $V_{Iswitch}$  value.

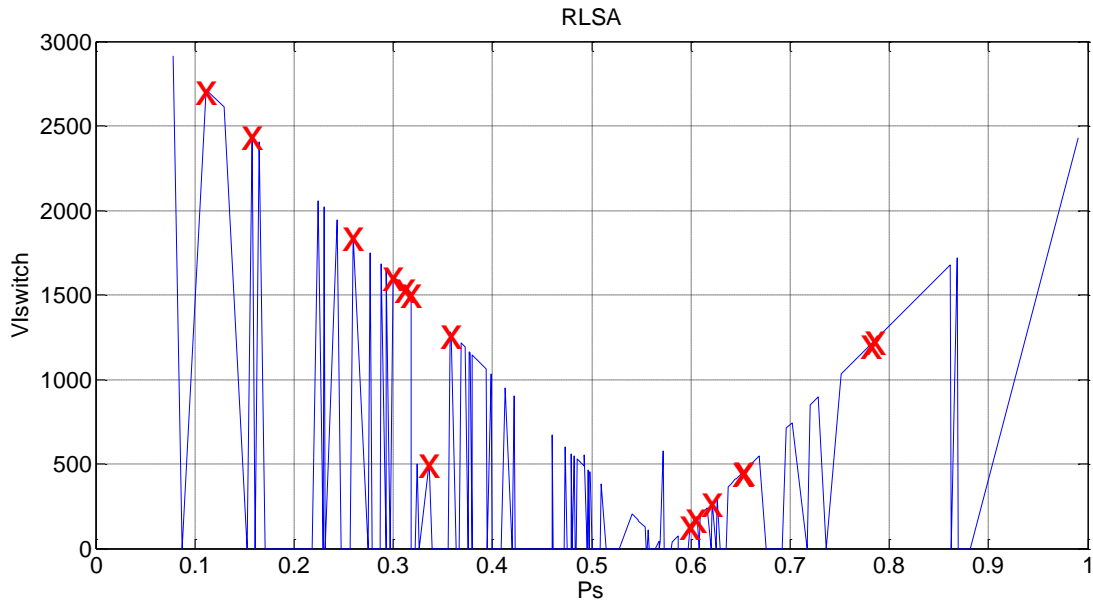


Figure A-28: The cost for non optimal work with switching in random normal Ps distribution – algorithm “RLSA” in Ps domain. The graph represents dynamic changes in  $V_{Iswitch}$  values over Ps values and dynamic switching operation (marked by ‘X’).

Presentation of “CTSA” algorithm (Figure A-29): system performs switches in selective high  $V_{Iswitch}$  value points due to switching frequency. Here the switching frequency value  $\psi$  doesn’t allow the algorithm to perform switches at all high valued  $V_{Iswitch}$  points.

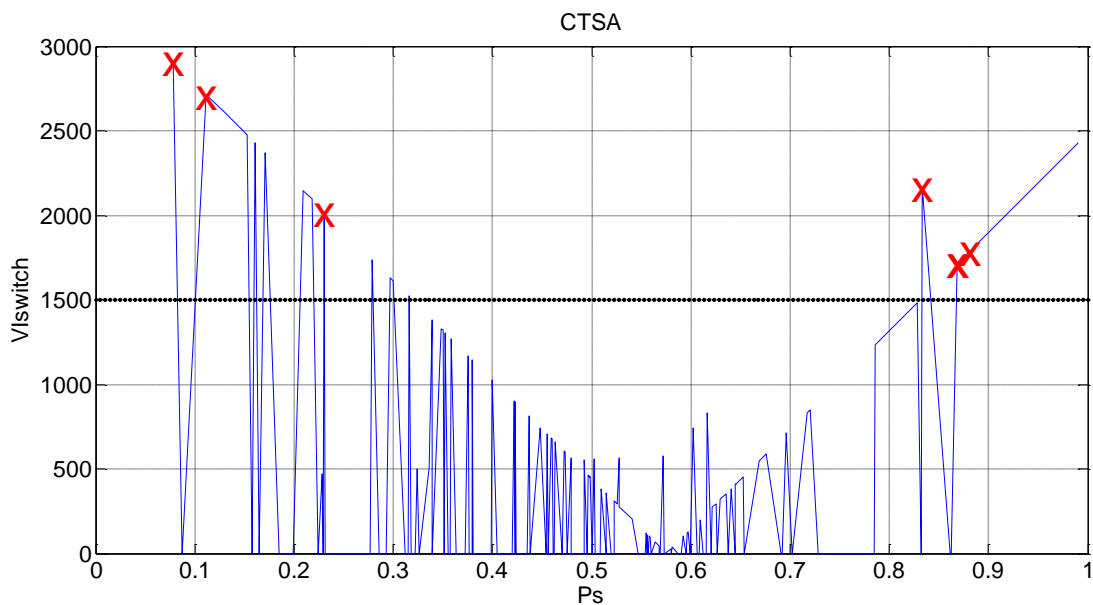


Figure A-29: The cost for non optimal work with switching in random normal Ps distribution – algorithm “CTSA” in Ps domain. The graph represents dynamic changes in  $V_{Iswitch}$  values over Ps values and dynamic switching operation (marked by ‘X’).

Presentation of “CFSA” algorithm (Figure A-30): a visible improvement – the controller performs switches at all high valued  $V_{Iswitch}$  points.

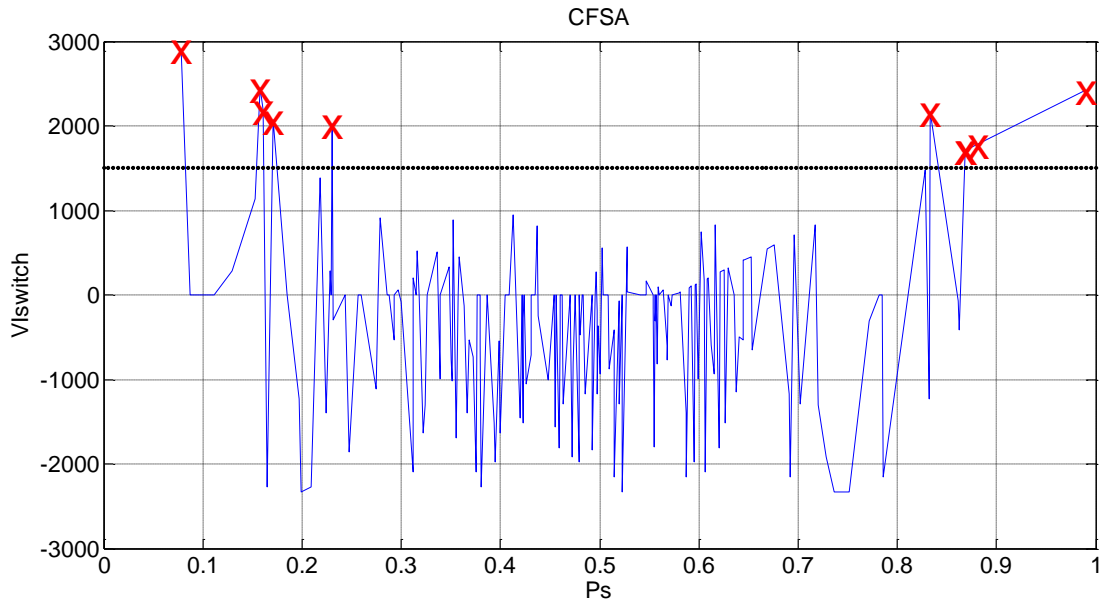


Figure A-30: The cost for non optimal work with switching in random normal  $P_s$  distribution – algorithm “CFSA” in  $P_s$  domain. The graph represents dynamic changes in  $V_{Iswitch}$  values over  $P_s$  values and dynamic switching operation (marked by ‘X’).

Presentation of “PRSA” algorithm (Figure A-31): the controller performs switches at the most strategic points. Here we can see clearly that despite the fact that the controller makes fewer switches than in “CFSA” algorithm, it manages to achieve even better gain in the objective function score. This idea is presented by the fact that the majority of the values in Figure A- have  $V_{Iswitch}$  value of 0 or negative, which means that these values are optimal.

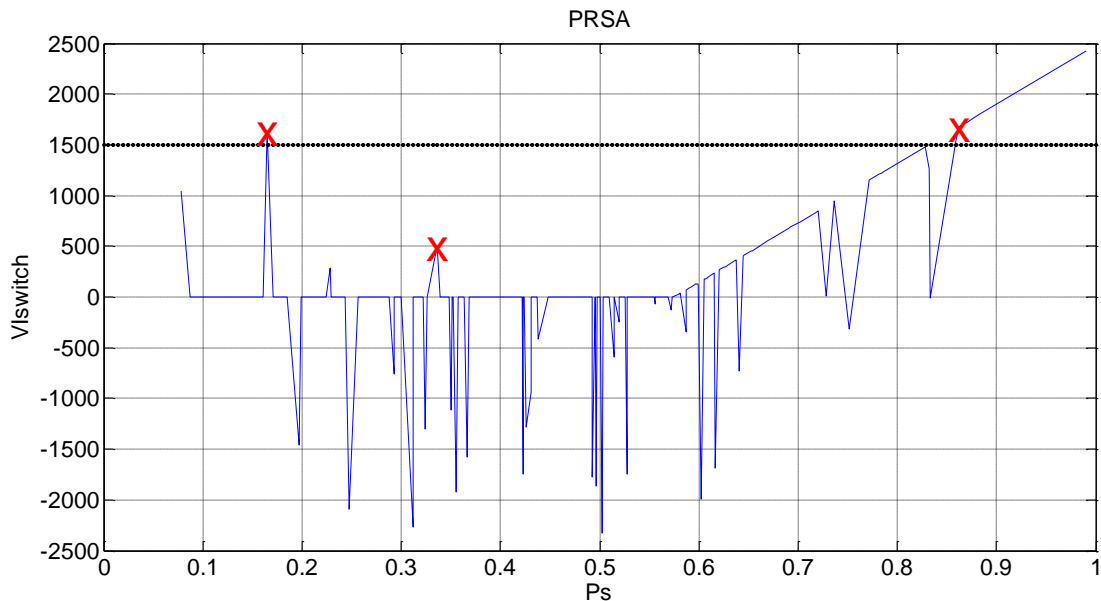


Figure A-31: The cost for non optimal work with switching in random normal  $P_s$  distribution – algorithm “PRSA” in  $P_s$  domain. The graph represents dynamic changes in  $V_{Iswitch}$  values over  $P_s$  values and dynamic switching operation (marked by ‘X’).



**Analysis for constant d'r value and normally distributed random values of d'h and Ps values**

In this simulation Ps and d'h parameters receive random values with normal distribution with mean value of 0.5 and standard deviation 0.5. In this section, two parameters (d'h and Ps) are changed dynamically during the task performance instead of one (Ps) that was analyzed in section 6.1. The aim of this simulation is to analyze the performance of the proposed algorithms to changes of the two parameters which makes a more difficult scenario for the system's operation. The robot's sensitivity, d'r, was set as a constant value due to the finding of Oren (2006) that robot's sensitivity changes have neglectible influence on the system performance compared to human sensitivity and target probability changes. The simulation parameters are summarized in Table A-2:

**Table A-2: Summary of simulation parameters for analysis of rand normal dh and Ps values.**

Parameter	Value
N	1000
V <sub>H</sub>	10
V <sub>AR</sub>	-1
V <sub>CR</sub>	3
V <sub>M</sub>	5
Ps	ranged from 0.1 to 0.9
V <sub>C</sub>	-2
V <sub>t</sub>	-2000 hr <sup>-1</sup>
decision time, t <sub>D</sub>	5 s/object
motoric time, t <sub>M</sub>	2 s/(detected object)
robot time, t <sub>r</sub>	0.01 s/object
Robot sensitivity, d' <sub>r</sub>	0.3
Human sensitivity, d' <sub>h</sub>	Ranged from -3 to 3
System response time, t <sub>response</sub>	5 sec
Frequency time, Ψ	0.000001 sec
Threshold	1500

Figure A-32 represents values of V<sub>Iswitch</sub> for each of four different starting collaboration levels.

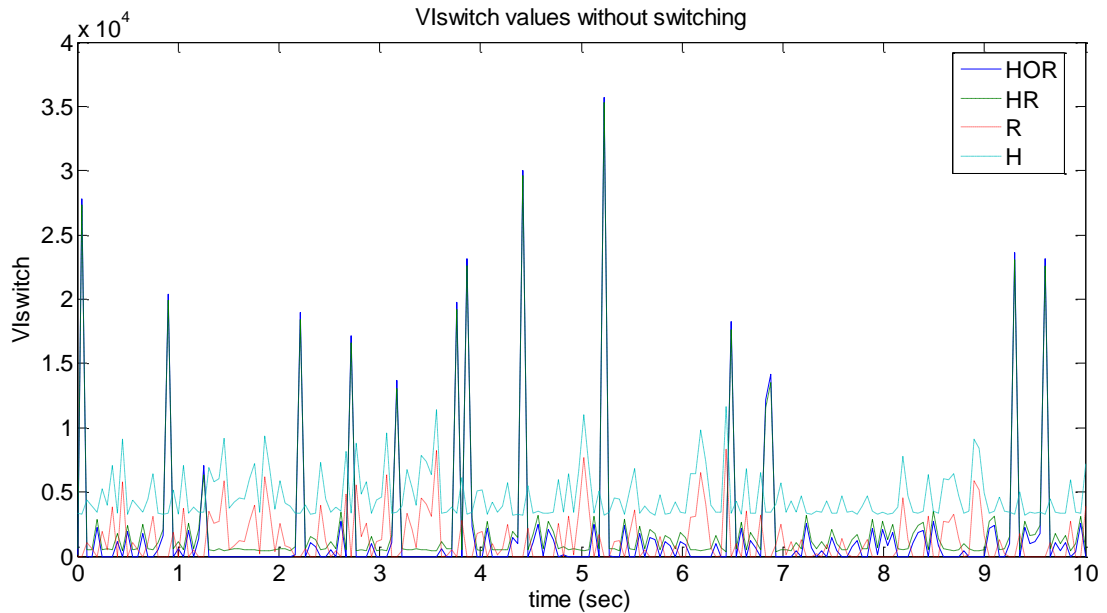


Figure A-32: The cost for non optimal work (without switching) in random normal Ps and d'h distributions.

Graphical representation of “RLSA”, “CTSA”, “CFSA” and “PRSA” algorithms performance is shown below. The graphs present the dynamic switching operation of each algorithm and the gain in global objective function score which has been achieved by the algorithms during the simulation of the target recognition task of the system.

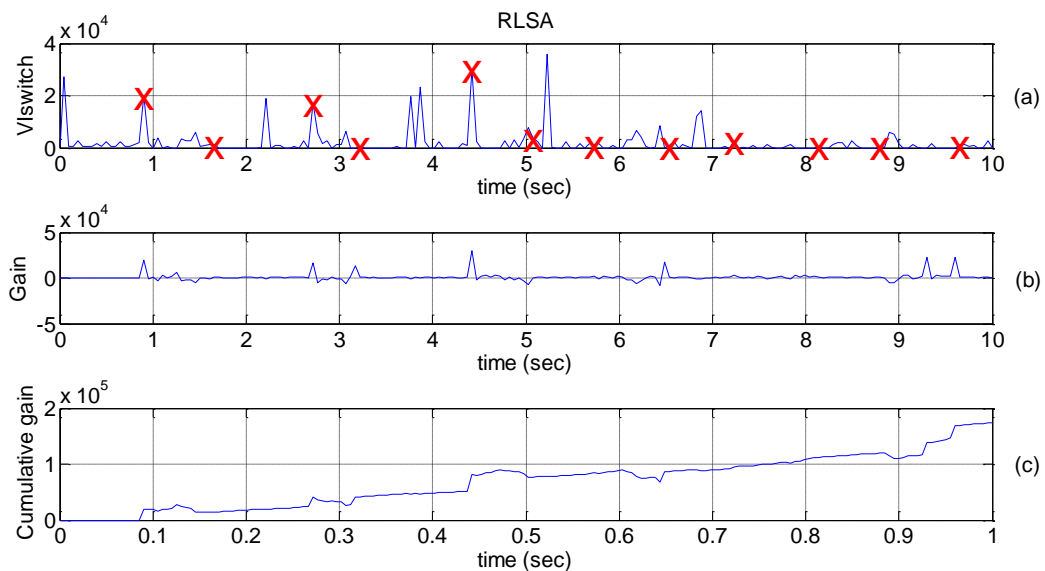
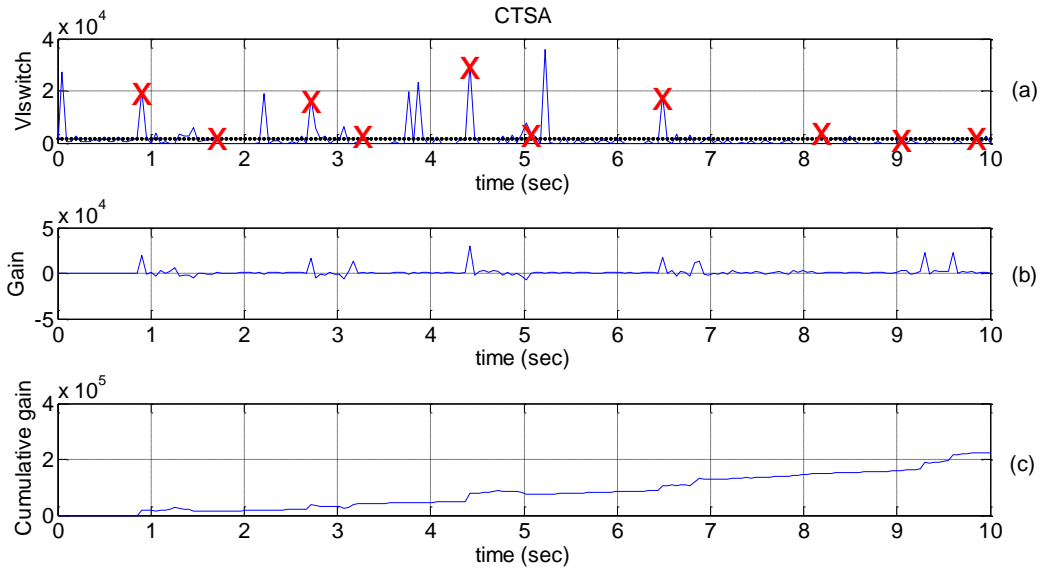
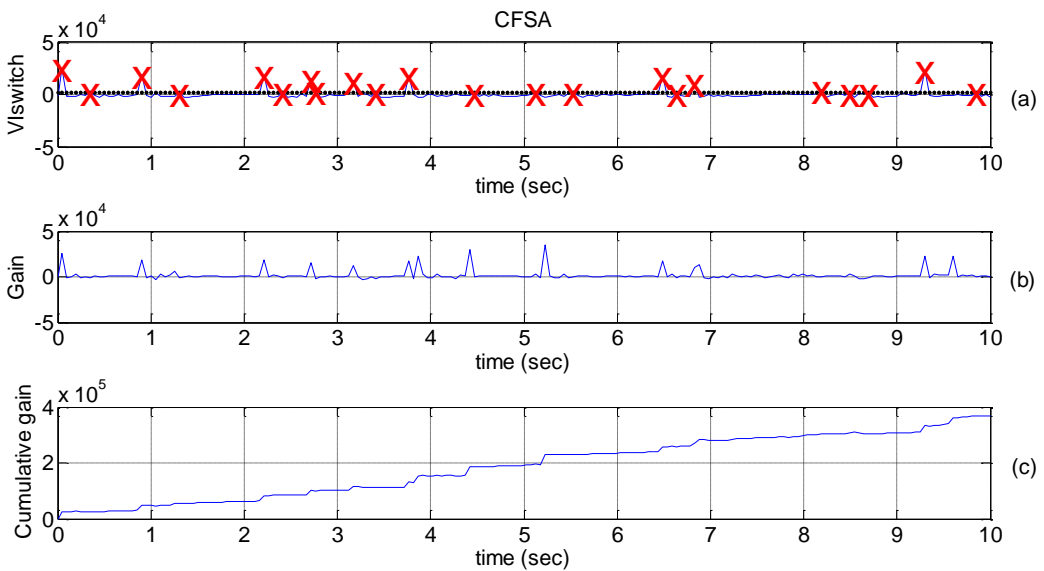


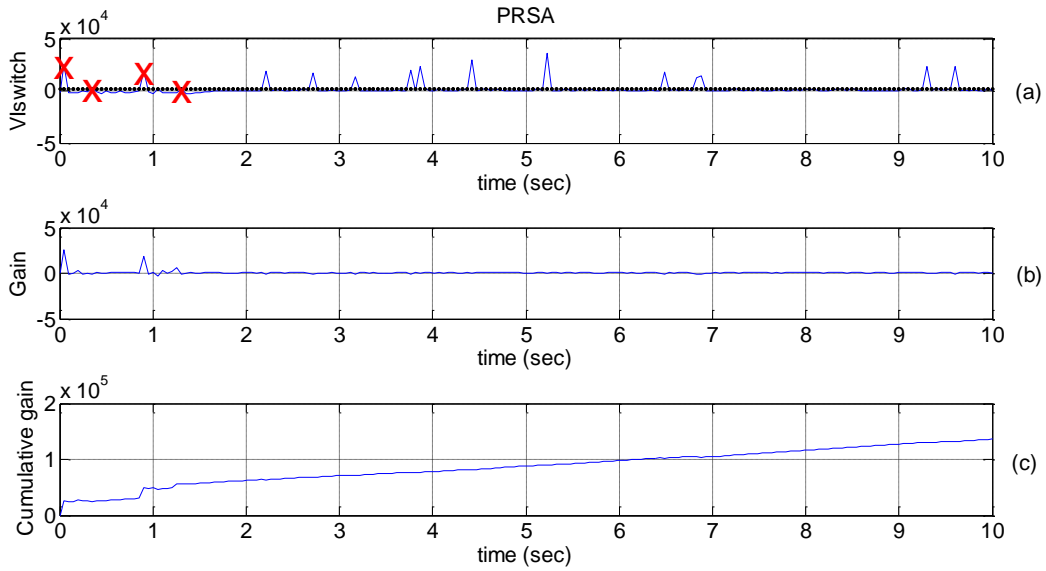
Figure A-33: The cost for non optimal work with switching in random normal Ps and d'h distributions – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{switch}$  values over time and dynamic switching operation, (b) graph represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.



**Figure A-25: The cost for non optimal work with switching in random normal Ps and d'h distributions – algorithm “CTSA”. (a) represents the dynamic changes in  $V_{\text{switch}}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

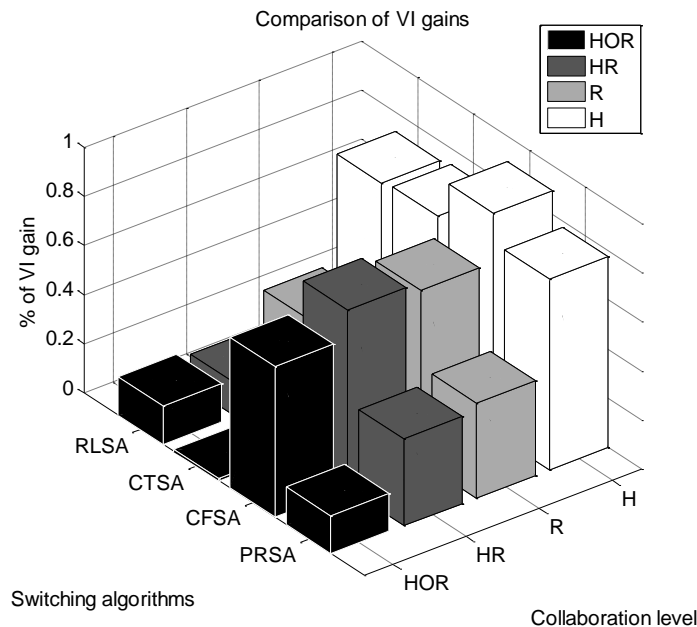


**Figure A-35: The cost for non optimal work with switching in random normal Ps and d'h distributions – algorithm “CFSA”. (a) represents the dynamic changes in  $V_{\text{switch}}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-36: The cost for non optimal work with switching in random normal Ps and d'h distributions – algorithm “PRSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

Figure A-37 presents a comparison of the gains in global objective function score for each algorithm for different starting collaboration levels. The gains are in percentage in scale of 0 to 1 (100%) relative to case with no switching at all. The negative values indicate negative effect of the dynamic switching operation to the system that results in decreasing the objective function score.



**Figure A-37: Summary of all algorithm gains for different starting collaboration levels in random normal Ps and d'h distributions in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.**

The analysis results indicate that the algorithms performance is not very much influenced by an introduction of other dynamically changing parameters to the system. In fact, the algorithms performed well under the conditions given in this analysis.

### Analysis of a case when $P_s$ change rate isn't equal to image sampling rate

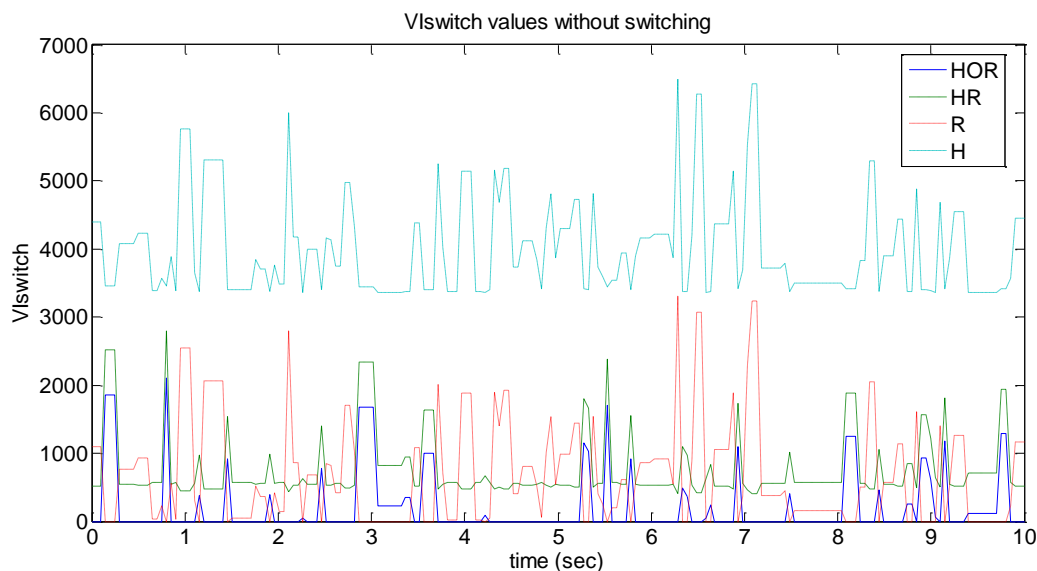
$P_s$  changes randomly over time with normally distributed values. The change rate of  $P_s$  values is also a random normal distribution which is not influenced by the image sampling rate of the system.

The simulation parameters are summarized in Table A-3:

**Table A-3: Summary of simulation parameters for analysis of a case when  $P_s$  change rate isn't equal to image sampling rate.**

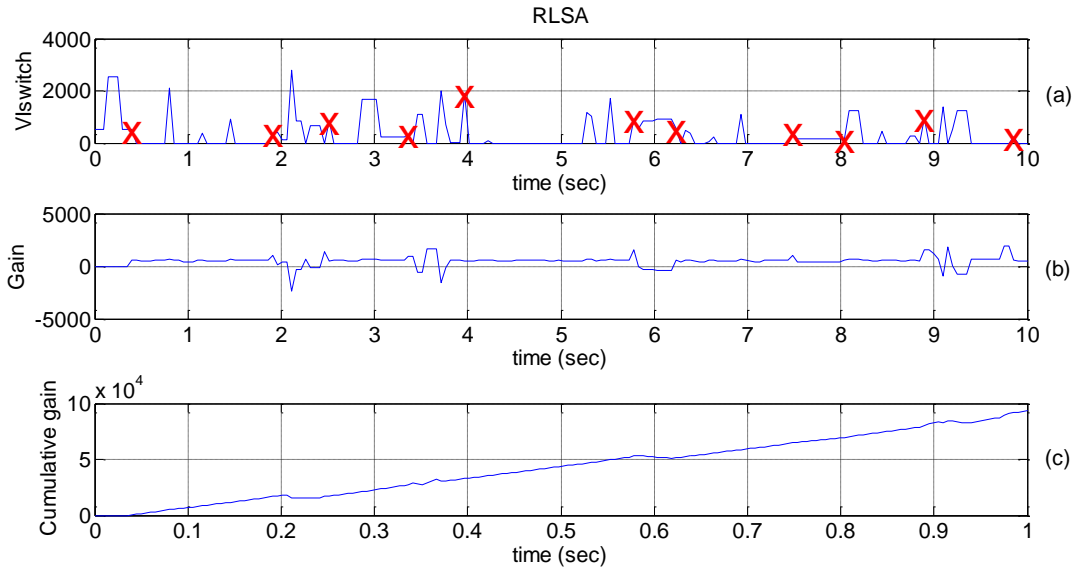
Parameter	Value
N	1000
$V_H$	10
$V_{AR}$	-1
$V_{CR}$	3
$V_M$	5
$P_s$	ranged from 0.1 to 0.9
$V_C$	-2
$V_t$	-2000 hr <sup>-1</sup>
decision time, $t_D$	5 s/object
motoric time, $t_M$	2 s/(detected object)
robot time, $t_r$	0.01 s/object
Robot sensitivity, $d'_r$	0.3
Human sensitivity, $d'_h$	0.7
System response time, $t_{response}$	5 sec
Frequency time, $\Psi$	0.000001 sec
Threshold	1500

Figure A-38 represents values of  $V_{Iswitch}$  for each of four different starting collaboration levels.

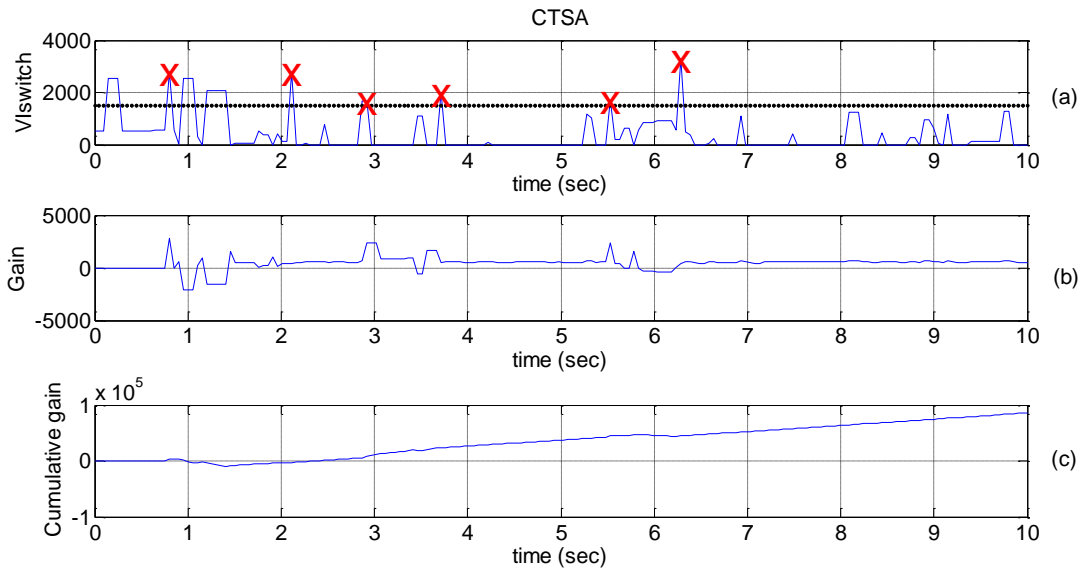


**Figure A-38: The cost for non optimal work (without switching) in random normal  $P_s$  change rate distribution.**

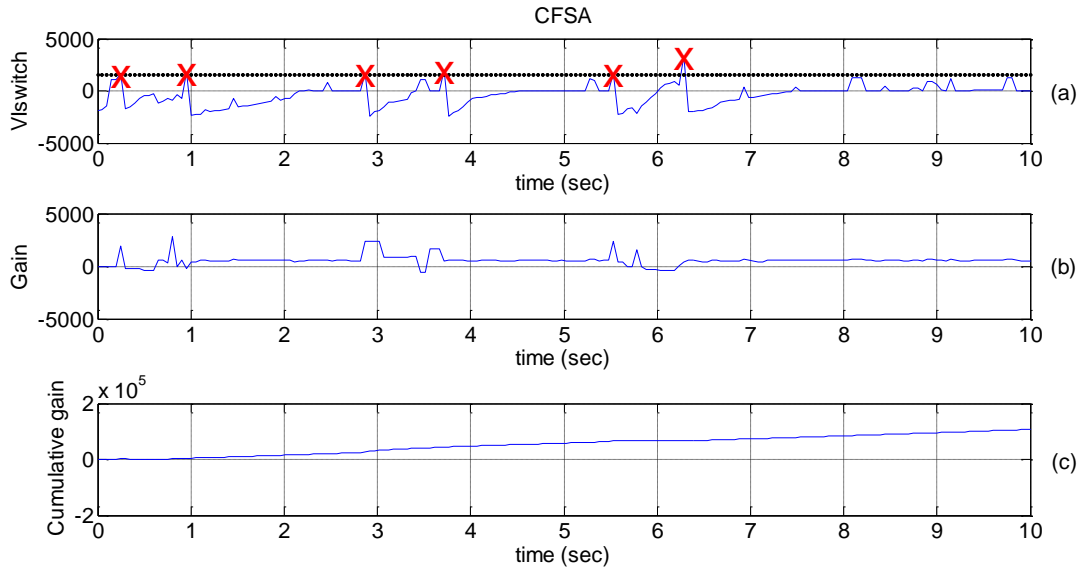
Graphical representation of “RLSA”, “CTSA”, “CFSA” and “PRSA” algorithms performance is shown below. The graphs present the dynamic switching operation of each algorithm and the gain in global objective function score which has been achieved by the algorithms during the simulation of the target recognition task of the system.



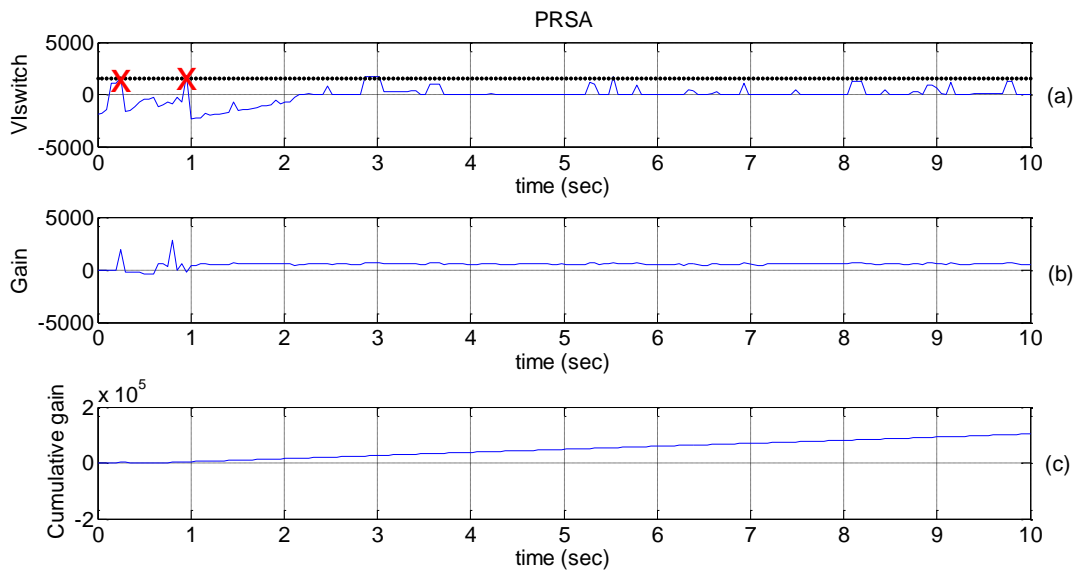
**Figure A-39: The cost for non optimal work with switching in random normal  $P_s$  change rate distribution – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-40: The cost for non optimal work with switching in random normal  $P_s$  change rate distribution – algorithm “CTSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-41: The cost for non optimal work with switching in random normal  $P_s$  change rate distribution – algorithm “CFSA”. (a) represents the dynamic changes in  $V_{\text{Switch}}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-42: The cost for non optimal work with switching in random normal  $P_s$  change rate distribution – algorithm “PRSA”. (a) represents the dynamic changes in  $V_{\text{Switch}}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

Figure A-43 presents a comparison of the gains in global objective function score for each algorithm for different starting collaboration levels. The gain is in percentage in scale of 0 to 1 (100%) relative to case with no switching at all. The negative values indicate negative effect of the dynamic switching operation to the system that results in decreasing the objective function score.

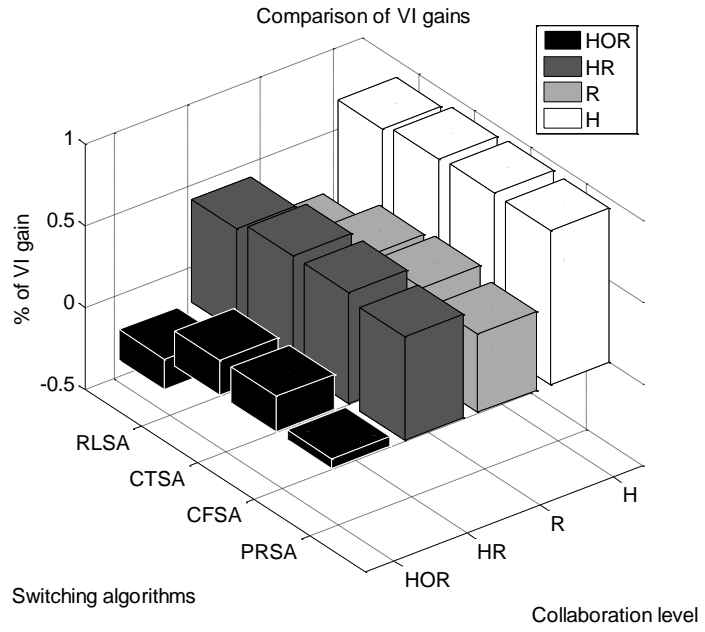


Figure A-43: Summary of all algorithm gains for different starting collaboration levels in random normal  $P_s$  change rate distribution in percentage from scale of 0 to 1 (100%) relative to case with no switching at all.



### Analysis of a case when switching frequency rate is higher than Ps change frequency

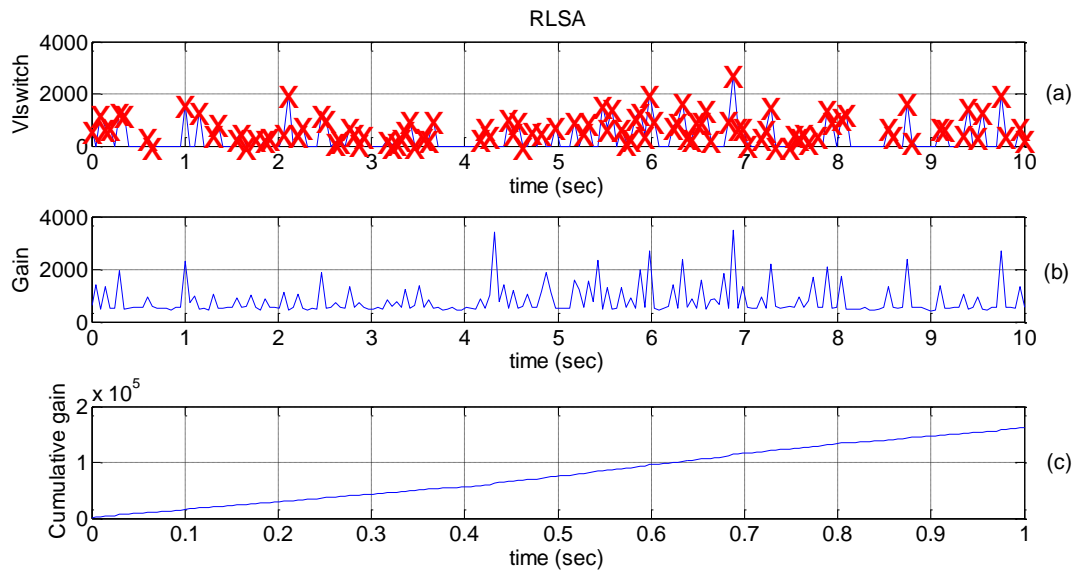
In this analysis, target probability  $P_s$  changes randomly over time with random normal distribution values. The rate of switching frequency was set in this simulation to be higher than the rate of changes in  $P_s$  values. It means that switching could be performed after every  $P_s$  change without any penalty for breaking the switching frequency. The threshold was set to zero in this simulation, in order not to make unnecessary constraint for the controller. The aim of this analysis is to test the influence of the switching frequency on the performance of the proposed algorithms.

The simulation parameters are summarized in Table A-4:

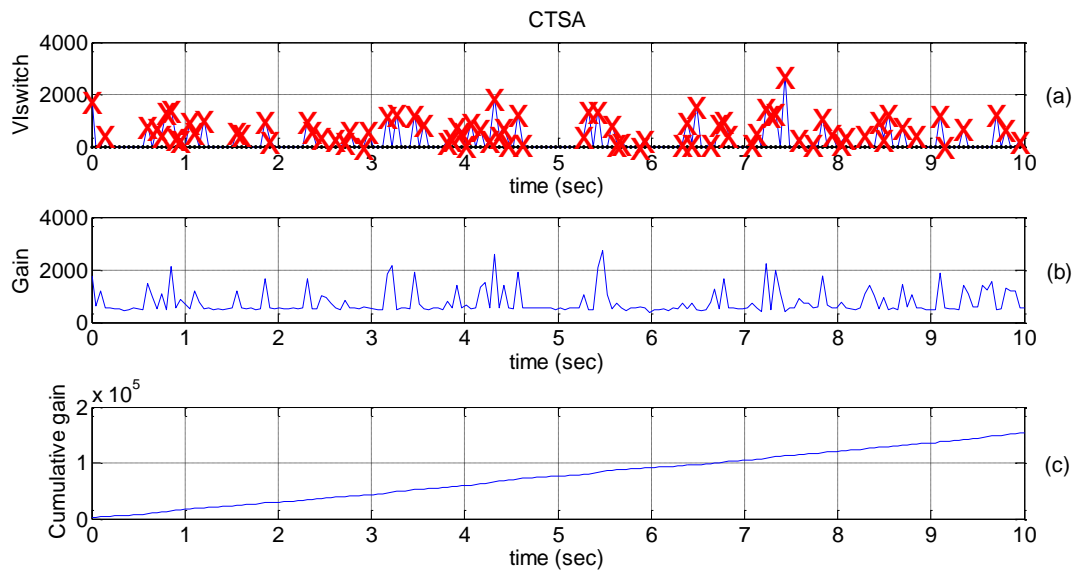
**Table A-4: Summary of simulation parameters for analysis of a case when  $P_s$  change frequency is larger than switching frequency.**

Parameter	Value
N	1000
$V_H$	10
$V_{AR}$	-1
$V_{CR}$	3
$V_M$	5
$P_s$	ranged from 0.1 to 0.9
$V_C$	-2
$V_t$	-2000 $\text{hr}^{-1}$
decision time, $t_D$	5 s/object
motoric time, $t_M$	2 s/(detected object)
robot time, $t_r$	0.01 s/object
Robot sensitivity, $d'_r$	0.3
Human sensitivity, $d'_h$	0.7
System response time, $t_{\text{response}}$	5 sec
Frequency time, $\Psi$	0.000000000001 sec
Threshold	0

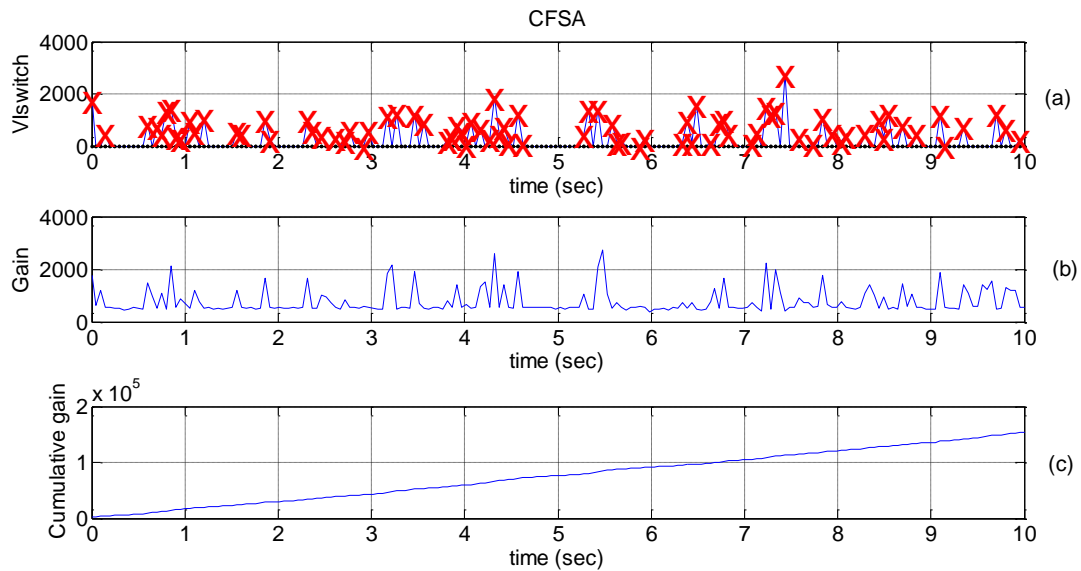
Graphical representation of “RLSA”, “CTSA”, “CFSA” and “PRSA” algorithms performance is shown below. The graphs present the dynamic switching operation of each algorithm and the gain in global objective function score which has been achieved by the algorithms during the simulation of the target recognition task of the system.



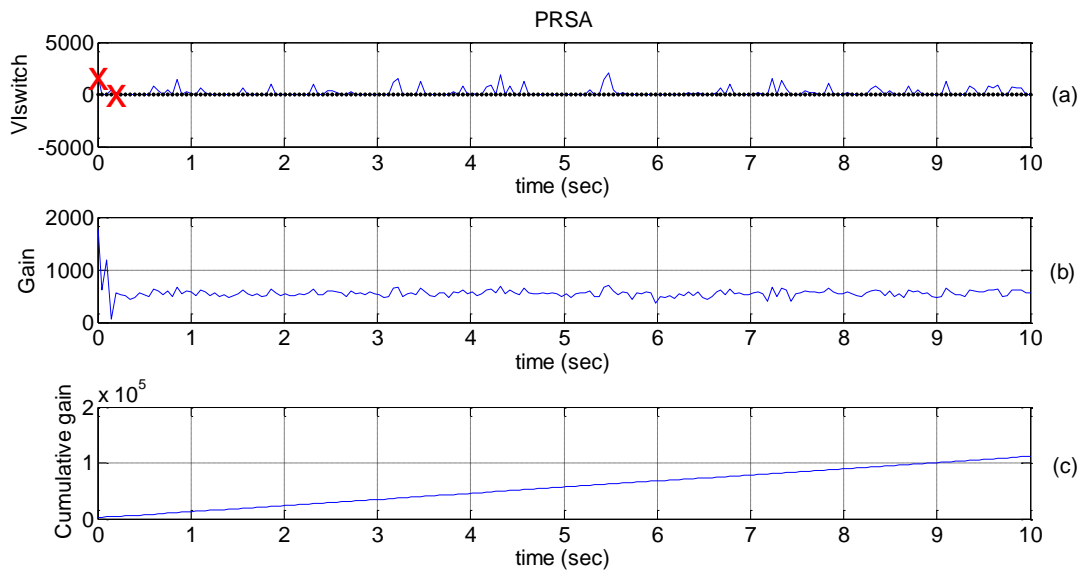
**Figure A-44: The cost for non optimal work with switching in random normal Ps distribution and different frequencies – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-45: The cost for non optimal work with switching in random normal Ps distribution and different frequencies – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{Iswitch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

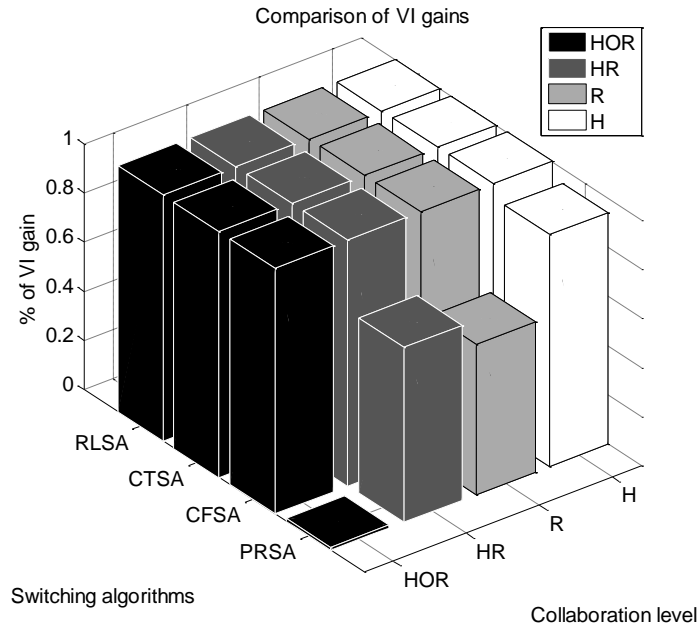


**Figure A-46: The cost for non optimal work with switching in random normal  $P_s$  distribution and different frequencies – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{switch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**



**Figure A-47: The cost for non optimal work with switching in random normal  $P_s$  distribution and different frequencies – algorithm “RLSA”. (a) represents the dynamic changes in  $V_{switch}$  values over time and dynamic switching operation, (b) represents the gain in objective function score achieved by each system’s operation, (c) represents the cumulative gain for the entire operation of the system. The switch points are marked with ‘X’.**

Figure A-48 presents a comparison of the gains in global objective function score for each algorithm for different starting collaboration levels. The gain is in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.



**Figure A-48: Summary of all algorithm gains for different starting collaboration levels in ascending normal Ps distribution and different frequencies in percentage in scale of 0 to 1 (100%) relative to case with no switching at all.**

The simulation revealed that in the case when switching frequency rate is higher than Ps change frequency, system performance can be improved by 100% (illustrated in “RLSA”, “CTSA” and “CFSA” algorithms) which equals to optimal system performance. In fact even when switching frequency rate is equal to Ps change frequency, 100% of improvement in the system performance also will be achieved by the dynamic switching operation.

## Appendix III: Dynamic switching pseudocodes

**Table A-5: Basic switching pseudocode**

1. Initialize:
  - a. Set  $k:=1$  { $k$  is the image sampling counter}
2. Get inputs:  $d'$ ,  $\beta$  and  $P_s$ , CCL
3. Calculate OCL
4. If  $OCL \neq CCL$  then
  - a. Make a switch and send output to the process (new collaboration level)
  - b. Update parameters:  
Set  $CCL:=OCL$
  - c. Calculate the gain in global objective function score:  

$$\bar{\sigma}(\zeta_k) = V_{IS_{current}}(\zeta_k) - V_{IS_{reference}}(\zeta_k) + V_{IS_{switch}}(\zeta_k)$$

$$V_{IS_{switch}}(\zeta_k) = \begin{cases} (V_{IS_{optimal}}(\omega_k) - V_{IS_{current}}(\omega_k)) & \text{if } k \text{ is switch point} \\ 0 & \text{otherwise} \end{cases}$$
  - d. Jump to step 5.
- Else
  - e. Go to step 5.
5. If  $k=m$  then
  - a. Calculate the cumulative gain of the entire operation of the system:  

$$f := \sum_{k=1}^m \bar{\sigma}(\zeta_k)$$
  - b. Stop
- Else
  - c. Set  $k:=k+1$
  - d. Go to step 2.

**Table A-6: “RLSA” pseudocode**

1. Initialize:
  - a. Set  $k:=1$  { $k$  is the image sampling counter}
  - b. Set  $\tau :=0$  { $\tau$  is the time between switching operations}
  - c. Set timer:=0 {timer is counting time during systems' operation}
2. Get inputs:  $d'$ ,  $\beta$  and  $P_s$ , CCL
3. Calculate OCL
4. Update timer
5. Set  $\tau := \tau + \text{timer}$
6. If  $\text{OCL} \neq \text{CCL}$  and  $(V_{IS_{\text{optimal}}} - V_{IS_{\text{current}}}) + t_{\text{response}} \times V_t > 0$  and  $\tau \geq \psi$  then
  - a. Make a switch and send output to the process (new collaboration level)
  - b. Update parameters:  
Set  $\text{CCL}:=\text{OCL}$   
Set  $\tau :=0$
  - c. Calculate the gain in global objective function score:  

$$\bar{\sigma}(\zeta_k) = V_{IS_{\text{current}}}(\zeta_k) - V_{IS_{\text{reference}}}(\zeta_k) + V_{IS_{\text{switch}}}(\zeta_k)$$

$$V_{IS_{\text{switch}}}(\zeta_k) = \begin{cases} V_{IS_{\text{switch}}}(\omega_k) & \text{if } k \text{ is switch point} \\ 0 & \text{otherwise} \end{cases}$$
  - d. Jump to step 7.

Else

  - e. Display: “Collaboration level not optimal”
  - f. If human operator wants to intervene  
Go to step 6.a.  
Else  
Go to step 7.
7. If  $k=m$  then
  - a. Calculate the cumulative gain of the entire operation of the system:  

$$f := \sum_{k=1}^m \bar{\sigma}(\zeta_k)$$
  - b. Stop

Else

  - c. Set  $k:=k+1$
  - d. Go to step 2.

**Table A-7: “CTSA” pseudocode**

1. Initialize:
  - a. Set  $k:=1$  { $k$  is the image sampling counter}
  - b. Set  $\tau :=0$  { $\tau$  is the time between switching operations}
  - c. Set timer:=0 {timer is counting time during systems' operation}
2. Get inputs:  $d'$ ,  $\beta$  and  $P_s$
3. Calculate OCL
4. Update timer
5. Set  $\tau := \tau + \text{timer}$
6. If  $\text{OCL} \neq \text{CCL}$  and  $(V_{\text{IS}_{\text{optimal}}} - V_{\text{IS}_{\text{current}}}) + t_{\text{response}} \times V_t > B$  and  $\tau \geq \psi$  then
  - a. Make a switch and send output to the process (new collaboration level)
  - b. Update parameters:  
Set  $\text{CCL} := \text{OCL}$   
Set  $\tau := 0$
  - c. Calculate the gain in global objective function score:  

$$\bar{\sigma}(\zeta_k) = V_{\text{IS}_{\text{current}}}(\zeta_k) - V_{\text{IS}_{\text{reference}}}(\zeta_k) + V_{\text{IS}_{\text{switch}}}(\zeta_k)$$

$$V_{\text{IS}_{\text{switch}}}(\zeta_k) = \begin{cases} V_{\text{IS}_{\text{switch}}}(\omega_k) & \text{if } k \text{ is switch point} \\ 0 & \text{otherwise} \end{cases}$$
  - d. Jump to step 7.
- Else
  - e. Display: “Collaboration level not optimal”
  - f. If human operator wants to intervene  
Go to step 6.a.
  - Else  
Go to step 7.
7. If  $k=m$  then
  - a. Calculate the cumulative gain of the entire operation of the system:  

$$f := \sum_{k=1}^m \bar{\sigma}(\zeta_k)$$
  - b. Stop
- Else
  - c. Set  $k:=k+1$
  - d. Go to step 2.

**Table A-8: “CFSA” pseudocode**

1. Initialize:
  - a. Set  $k:=1$  { $k$  is the image sampling counter}
  - b. Set  $\tau :=0$  { $\tau$  is the time between switching operations}
  - c. Set timer:=0 {timer is counting time during systems' operation}
2. Get inputs:  $d'$ ,  $\beta$  and  $P_s$
3. Calculate OCL
4. Update timer
5. Set  $\tau := \tau + \text{timer}$
6. If  $\text{OCL} \neq \text{CCL}$  and  $(V_{\text{IS}_{\text{optimal}}} - V_{\text{IS}_{\text{current}}}) + t_{\text{response}} \times V_t + \Psi \times V_p > B$  then
  - a. Make a switch and send output to the process (new collaboration level)
  - b. Update parameters:  
Set  $\text{CCL} := \text{OCL}$   
Set  $\tau := 0$
  - c. Calculate the gain in global objective function score:  

$$\bar{\sigma}(\zeta_k) = V_{\text{IS}_{\text{current}}}(\zeta_k) - V_{\text{IS}_{\text{reference}}}(\zeta_k) + V_{\text{IS}_{\text{switch}}}(\zeta_k)$$

$$V_{\text{IS}_{\text{switch}}}(\zeta_k) = \begin{cases} V_{\text{IS}_{\text{switch}}}(\omega_k) & \text{if } k \text{ is switch point} \\ 0 & \text{otherwise} \end{cases}$$
  - d. Jump to step 7.
- Else
  - e. Display: “Collaboration level not optimal”
  - f. If human operator wants to intervene  
Go to step 6.a.
  - Else  
Go to step 7.
7. If  $k=m$  then
  - a. Calculate the cumulative gain of the entire operation of the system:  

$$f := \sum_{k=1}^m \bar{\sigma}(\zeta_k)$$
  - b. Stop
- Else
  - c. Set  $k:=k+1$
  - d. Go to step 2.



**Table A-9: “PRSA” pseudocode**

1. Initialize:
  - e. Set  $k:=1$  {  $k$  is the image sampling counter }
  - f. Set  $\tau :=0$  {  $\tau$  is the time between switching operations }
  - g. Set timer:=0 {timer is counting time during systems’ operation}
  - h. Set  $i:=0$  {  $i$  is the image sampling counter between switches }
  - i. Set  $C_j:=0 \ j \in \{1,2,3,4\}$  {  $C_j$  is the counters of collaboration levels 1 to 4 }
2. Get inputs:  $d'$ ,  $\beta$  and  $P_s$
3. Calculate OCL
4. Update timer
5. Set  $\tau := \tau + \text{timer}$
6. Set  $i:=i+1$
7. Case of
  - a. OCL=HOR then  $C_1:= C_1+1$
  - b. OCL=HR then  $C_2:= C_2+1$
  - c. OCL=R then  $C_3:= C_3+1$
  - d. OCL=H then  $C_4:= C_4+1$
8. If  $\text{OCL} \neq \text{CCL}$  and  $[(V_{IS_{\text{optimal}}} - V_{IS_{\text{current}}}) + t_{\text{response}} \times V_t + \Psi \times V_p > B$  or  $\tau \geq \psi]$  and  $C_j > (i/2)$  then
  - a. Make a switch and send output to the process (new collaboration level)
  - b. Update parameters:
    - Set  $\text{CCL}:=\text{OCL}$
    - Set  $\tau :=0$
    - Set  $i:=0$
  - c. Calculate the gain in global objective function score:
 
$$\bar{\sigma}(\zeta_k) = V_{IS_{\text{current}}}(\zeta_k) - V_{IS_{\text{reference}}}(\zeta_k) + V_{IS_{\text{switch}}}(\zeta_k)$$

$$V_{IS_{\text{switch}}}(\zeta_k) = \begin{cases} V_{IS_{\text{switch}}}(\omega_k) & \text{if } k \text{ is switch point} \\ 0 & \text{otherwise} \end{cases}$$
  - d. Jump to step 9.
- Else
  - e. Display: “Collaboration level not optimal”
  - f. If human operator wants to intervene
    - Go to step 8.a.
  - Else
    - Go to step 9.
9. If  $k=m$  then
  - a. Calculate the cumulative gain of the entire operation of the system:
 
$$f := \sum_{k=1}^m \bar{\sigma}(\zeta_k)$$
  - b. Stop
- Else
  - c. Set  $k:=k+1$
  - d. Go to step 2.

## Appendix IV: Matlab programs

### Simulation for linear distribution of Ps:

% This program performs dynamic switching at linear distribution of Ps parameters  
% the program calculates the hit and false alarm probabilities and the  
% value of the operational cost according to betas and dtags  
% This version will show the objective function values for all collaboration  
% levels and for all betas combinations.

```
clc
close all
clear all
Psvector=linspace(0.1,0.9,200);
for rr=1:4
j=1;
i=1;
l=1;
p=1;
CCL=rr %current collaboration level
CCLw=rr;
CCLf=rr;
CCLnew=rr;
VI1(rr)=0;
VI2(rr)=0;
VI3(rr)=0;
VI4(rr)=0;
VI5(rr)=0;
bound1=1;
Xaxis=linspace(0,1,200);
N=1000; % # of objects
Nstr=num2str(N);
tmin=0.0000005; %frequency of the switch
tlastswitch=now; %time of the last switch for "RLSA"
tlastswitchw=now; %time of the last switch for "CTSA"
tlastswitchf=now; %time of the last switch for "CFSA"
tlastswitchnew=now;%time of the last switch for "PRSA"
ttemp=now;
Vp=-5000000000; %penalty for frequency
minv=1000; %minimal value of VIs witch for switching
maxv=1500; %maximal value of VIs witch for switching
VFA2H=10; % VFA/VH aspect ratio range
VAR=1;
VARstr=num2str(VFA2H(VAR)*10);
if VFA2H(VAR)==0.333
    VARstr=num2str(3);
end

for Pscount=1:1:200;
Ps=Psvector(Pscount);
Psstr=num2str(Ps*100);
dtag=0.6;
dtag=dtag-1; %d' for human (second detector)
Dh=num2str(-dtag*10);
dtagR=0.1; %d' for robot
dtagR=dtagR-1;
Dr=num2str(-dtagR*10);

VH=10;
VM=5;
VCR=3;
VHstr=num2str(VH);
VFA=-VH.*VFA2H(VAR);
Vc=-2;
VCstr=num2str(-Vc);
Vt=-2000/3600;
Vtstr=num2str(-Vt*3600);

trespond=1;
tr=0.01;

c3=1;
lnbetar=0;
```

```

c2=1;
lnbetah=0;
c1=1;
lnbetarh=2;

```

```

% the probabilities of the robot
Zsr(c1,c2,c3)=(-2.*lnbetah+dtagR.^2)/(2.*dtagR);
Znr(c1,c2,c3)=(-2.*lnbetar-dtagR.^2)/(2.*dtagR);
phr(c1,c2,c3)=1-normcdf(Zsr(c1,c2,c3));
pfar(c1,c2,c3)=1-normcdf(Znr(c1,c2,c3));
ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));

```

```

if lnbetah==0 & lnbetar==0 & lnbetarh==0
    Zsrtest=(-2.*lnbetah+dtagR.^2)/(2.*dtagR);
    Znrtest=(-2.*lnbetar-dtagR.^2)/(2.*dtagR);
    phrtest=1-normcdf(Zsr(c1,c2,c3));
    pfartest=1-normcdf(Znr(c1,c2,c3));
    ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
    ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));

```

```
end
```

```

% the optimal parameters of the robot if he was a single detector
betastar(c1,c2,c3)=(1-Ps)/Ps.*VFA2H(VAR); % calculating Beta*
ZsRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))+dtagR.^2)/(2.*dtagR);
ZnRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))-dtagR.^2)/(2.*dtagR);
phrstar(c1,c2,c3)=1-normcdf(ZsRstar(c1,c2,c3));
pfarstar(c1,c2,c3)=1-normcdf(ZnRstar(c1,c2,c3));

```

```

% the probabilities of the HO (second detector) for object that the
% robot didn't detect
ZsH(c1,c2,c3)=(-2.*lnbetah+dtag.^2)/(2.*dtag);
ZnH(c1,c2,c3)=(-2.*lnbetar-dtag.^2)/(2.*dtag);
phh(c1,c2,c3)=1-normcdf(ZsH(c1,c2,c3));
pfah(c1,c2,c3)=1-normcdf(ZnH(c1,c2,c3));

```

```

% the probabilities of the HO (second detector) for object that the robot
% already detected
ZsRH(c1,c2,c3)=(-2.*lnbetarh+dtag.^2)/(2.*dtag);
ZnRH(c1,c2,c3)=(-2.*lnbetarh-dtag.^2)/(2.*dtag);
phrh(c1,c2,c3)=1-normcdf(ZsRH(c1,c2,c3));
pfarh(c1,c2,c3)=1-normcdf(ZnRH(c1,c2,c3));

```

```
% the time parameters
```

```

tHh(c1,c2,c3)=5;
tFAh(c1,c2,c3)=5;
tHrh(c1,c2,c3)=5;
tFArh(c1,c2,c3)=5;

```

```

tMh(c1,c2,c3)=5;
tCRh(c1,c2,c3)=5;
tMrh(c1,c2,c3)=5;
tCRrh(c1,c2,c3)=5;
tmotor=2;

```

```

PHs(c1,c2,c3)=phr(c1,c2,c3).*phrh(c1,c2,c3)+(1-phr(c1,c2,c3)).*phh(c1,c2,c3);
VHs(c1,c2,c3)=N.*Ps.*PHs(c1,c2,c3).*VH;
PMs(c1,c2,c3)=phr(c1,c2,c3).*(1-phrh(c1,c2,c3))+(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3));
VMs(c1,c2,c3)=N.*Ps.*PMs(c1,c2,c3).*VM;
FFAs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3);
VFAs(c1,c2,c3)=FFAs(c1,c2,c3).*VFA;
FCRs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3))+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3));
VCRs(c1,c2,c3)=FCRs(c1,c2,c3).*VCR;

```

```

ts(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3).*tHh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*tFArh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*tFAh(c1,c2,c3)...
+N.*Ps.*phr(c1,c2,c3).*(1-phrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

```

```

tsHORr(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*(tHrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-
phr(c1,c2,c3)).*phh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*(tFArh(c1,c2,c3)+tmotor)+N.*(1-
Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...

```

+N.\*Ps.\*phr(c1,c2,c3).\*(1-phrh(c1,c2,c3)).\*tMrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*(1-phh(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*(1-pfarh(c1,c2,c3)).\*tCRrh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*(1-pfah(c1,c2,c3)).\*tCRh(c1,c2,c3)+tr;

tsHOR(c1,c2,c3)=N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3).\*tHrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3).\*(tHh(c1,c2,c3)+tmotor)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3).\*tFArh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3).\*(tFAh(c1,c2,c3)+tmotor)...

+N.\*Ps.\*phr(c1,c2,c3).\*(1-phrh(c1,c2,c3)).\*(tMrh(c1,c2,c3)+tmotor)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*(1-phh(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*(1-pfarh(c1,c2,c3)).\*(tCRrh(c1,c2,c3)+tmotor)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*(1-pfah(c1,c2,c3)).\*tCRh(c1,c2,c3)+tr;

Ndetect(c1,c2,c3)=(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3));

VTs(c1,c2,c3)=ts(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VTsHORr(c1,c2,c3)=tsHORr(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VTsHOR(c1,c2,c3)=tsHOR(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VIs(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTs(c1,c2,c3);

VIsHORr(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHORr(c1,c2,c3);

VIsHOR(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHOR(c1,c2,c3);

PHsR(c1,c2,c3)=phr(c1,c2,c3);

VHsR(c1,c2,c3)=N.\*Ps.\*PHsR(c1,c2,c3).\*VH;

FFAsR(c1,c2,c3)=N.\*(1-Ps).\*pfar(c1,c2,c3);

VFAsR(c1,c2,c3)=FFAsR(c1,c2,c3).\*VFA;

tsR(c1,c2,c3)=tr;

VTsR(c1,c2,c3)=tsR(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3)).\*Vc;

VIsR(c1,c2,c3)=VHsR(c1,c2,c3)+VFAsR(c1,c2,c3)+VTsR(c1,c2,c3);

% the probabilities of the HO collaboration level were taken from the  
 % robot probabilities and the difference between the HO and the R  
 % collaboration levels is just on the times parameters.

PHsHO(c1,c2,c3)=phr(c1,c2,c3);

VHsHO(c1,c2,c3)=N.\*Ps.\*PHsHO(c1,c2,c3).\*VH;

FFAsHO(c1,c2,c3)=N.\*(1-Ps).\*pfar(c1,c2,c3);

VFAsHO(c1,c2,c3)=FFAsHO(c1,c2,c3).\*VFA;

tsHO(c1,c2,c3)=N.\*Ps.\*PHsHO(c1,c2,c3).\*(tHh(c1,c2,c3)+tmotor)+FFAsHO(c1,c2,c3).\*(tFAh(c1,c2,c3)+tmotor)...

+N.\*Ps.\*(1-PHsHO(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*tCRh(c1,c2,c3);

VTsHO(c1,c2,c3)=tsHO(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3)).\*Vc;

VIsHO(c1,c2,c3)=VHsHO(c1,c2,c3)+VFAsHO(c1,c2,c3)+VTsHO(c1,c2,c3);

%-----  
 %switching calculations  
 %-----

CL(Pscount,:)= [VIsHORr(c1,c2,c3) VIsHOR(c1,c2,c3) VIsR(c1,c2,c3) VIsHO(c1,c2,c3)]; % values' matrix of different collaboration levels

[C,CCLtemp]=max(CL(Pscount,:)); %C=value, CCLtemp=location

SV(1,Pscount)=[C]; % switch vector of best collaboration values for each row in CL

VIso=SV(1,Pscount);

VIscl=CL(Pscount,rr);

VIscl=CL(Pscount,CCL);

VIscl=CL(Pscount,CCLw);

VIscl=CL(Pscount,CCLf);

VIsclnew=CL(Pscount,CCLnew);

a1='HOR';

a2='HR';

a3='R';

a4='H';

switch CCLtemp

case 1

label1=a1;

case 2

label1=a2;

case 3

label1=a3;

otherwise

label1=a4;

end

```

switch CCL
case 1
    label2=a1;
case 2
    label2=a2;
case 3
    label2=a3;
otherwise
    label2=a4;
end

SVF(1,Pscount)=[CCLtemp];
%-----
%Switch objective fuction for plotting switch points
%-----

VIsutchl(rr,Pscount)=(VIso-VIscl); %+trespond*Vt+Vp*(ttemp+tmin-now)*(Vp*(ttemp+tmin-now)<0); %
VIsutch(rr,Pscount)=(VIso-VIscl); %"RLSA" for calculation
VIsutch1(rr,Pscount)=(VIso-VIscl); %"RLSA" for plot
VIsutchw(rr,Pscount)=(VIso-VIschw); %"CTSA" for calculation
VIsutchw1(rr,Pscount)=(VIso-VIschw); %"CTSA" for plot
VIsutchfw1(rr,Pscount)=(VIso-VIschw)+Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %"CFSA" for plot
VIsutchfw(rr,Pscount)=(VIso-VIschw); %"CFSA" for calculation
VIsutchfnew1(rr,Pscount)=(VIso-VIschwnew)+trespond*Vt+Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
%"PRSA" for plot
VIsutchfnew(rr,Pscount)=(VIso-VIschwnew); %"PRSA" for calculation

%-----

%-----
%switching algorithm
%-----

%"RLSA" algorithm

if CCLtemp~=CCL
tcurrent=now;
    if tcurrent-tlastswitch>tmin %checking the frequency

        disp('switch to');
        disp(label1);
        disp('from');
        disp(label2);

        CCL=CCLtemp
        coY(rr,j)=C;
        coX(rr,j)=Pscount;
        coYs(rr,j)=VIsutch1(rr,Pscount); %for the switching objective function plot
        tlastswitch=tcurrent;
        j=j+1;
        VIsutch(rr,Pscount)=-trespond*Vt; %updating VIsutch as optimal value after switching
    else
        disp('Collaboration level not optimal');
    end
end

%"CTSA" algorithm

if CCLtemp~=CCLw %optimization for switching
tcurrent=now;
if tcurrent-tlastswitchw>tmin %checking the frequency
    if VIsutchw(rr,Pscount)>maxv
        CCLw=CCLtemp;
        coYw(l)=C;
        coXw(rr,l)=Pscount;
        coYsw(rr,l)=VIsutchw1(rr,Pscount); %for the switching objective function plot
        tlastswitchw=tcurrent;
        l=l+1;
        VIsutchw(rr,Pscount)=-trespond*Vt; %updating VIsutch as optimal value after switching
    end
end
end

%"CFSA" algorithm

```

```

if CCLtemp~=CCLf %optimization for switching with frequency
tcurrent=now;
if VIs witchfw1(rr,Pscout)>maxv
CCLf=CCLtemp;
coXf(rr,i)=Pscout;
coYsfw(rr,i)=VIs witchfw1(rr,Pscout);
VIs witchfw(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %updating VI with response
time for switching
i=i+1;
tlastswitchf=tcurrent;
end
end

%”PRSA” algorithm

if CCLtemp~=CCLnew %optimization for switching with frequency
tcurrent=now;
if (tcurrent-tlastswitchnew>tmin)|(VIs witchfnew1(rr,Pscout)>maxv)
% if VIs witchfnew(Pscout)>maxv
bound2=Pscout;
BVF(bound1:bound2)=[CCLtemp];
newVF(bound1:bound2)=SVF(bound1:bound2)==BVF(bound1:bound2);
if sum(newVF(bound1:bound2))>((bound2-bound1)/2)
CCLnew=CCLtemp;

coXfnew(rr,p)=Pscout;
coYsfnew(rr,p)=VIs witchfnew1(rr,Pscout);

p=p+1;
bound1=Pscout+1;

VIs witchfnew(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
tlastswitchnew=tcurrent;
end
end
end

VI1(rr)=VIs witchl(rr,Pscout)+VI1(rr); %regular cost - S1
VI2(rr)=VIs witch(rr,Pscout)+VI2(rr); %cost with switch - “RLSA”
VI3(rr)=VIs witchw(rr,Pscout)+VI3(rr); %cost with switch and maxv filter - “CTSA”
VI4(rr)=VIs witchfw(rr,Pscout)+VI4(rr); %cost with switch and maxv filter after frequency penalty - “CFSA”
VI5(rr)=VIs witchfnew(rr,Pscout)+VI5(rr); %cost with switch and maxv filter after frequency penalty and new algorithm -
“PRSA”

end %Pscout
end %rr
%-----
%Figures
%-----

Figure(1)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);

Figure(2)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');

```

```

set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

```

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

```

Figure(3)%the score if we stay always at the same collaboration level
plot(Xaxis,VIswitchl(1,:),'-',Xaxis,VIswitchl(2,:),'--',Xaxis,VIswitchl(3,:),'.',Xaxis,VIswitchl(4,:),'-'.)
xlabel('time');
ylabel('VIswitch');
title('S1');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

```

```

plotvl=2;

```

```

Figure(4)
subplot(3,1,1)
plot(Xaxis,VIswitchl(plotvl,:))
xlabel('time');
ylabel('VIswitch');
title("RLSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coX(plotvl,find(coX(plotvl,:))))),coYs(plotvl,find(coYs(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitch(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitch(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

```

Figure(5)
subplot(3,1,1)
plot(Xaxis,VIswitchw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title("CTSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXw(plotvl,find(coXw(plotvl,:))))),coYsw(plotvl,find(coYsw(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchw(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchw(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

```

Figure(6)
subplot(3,1,1)
plot(Xaxis,VIswitchfw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title("CFSA");

```

```

set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXf(plotvl,find(coXf(plotvl,:))))),coYsfw(plotvl,find(coYsfw(plotvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfw(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfw(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(7)
subplot(3,1,1)
plot(Xaxis,VIswitchfnew1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title("PRSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXfnew(plotvl,find(coXfnew(plotvl,:))))),coYsfnew(plotvl,find(coYsfnew(plotvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(8)
subplot(2,1,1)
Xaxis=[1 2 3 4 5];
Yaxis=[(VI1-VI2)' (VI1-VI3)' (VI1-VI4)' (VI1-VI5)']';
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA','CTSA','CFSA','PRSA'})
xlabel('type of algorithm');
ylabel('VI gain');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

subplot(2,1,2)
Xaxis=[1 2 3 4 5];
Yaxis=[((VI1-VI2)/VI1)' ((VI1-VI3)/VI1)' ((VI1-VI4)/VI1)' ((VI1-VI5)/VI1)']';
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA','CTSA','CFSA','PRSA'})
xlabel('type of algorithm');
ylabel('VI gain in percentage');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

```



## Simulation for random normal distribution of Ps:

%This program performs dynamic switching at random normal distribution of Ps parameters  
 % the program calculates the hit and false alarm probabilities and the  
 % value of the operational cost according to betas and dtags  
 % This version will show the objective function values for all collaboration  
 % levels and for all betas combinations.

```

clc
close all
clear all
Psvector=randn(1,200);
ctmp=max(abs(Psvector));
Psvector=Psvector./(ctmp*2);
Psvector=abs(Psvector+0.49);
for rr=1:4
j=1;
i=1;
l=1;
p=1;
CCL=rr %current collaboration level
CCLw=rr;
CCLf=rr;
CCLnew=rr;
VII(rr)=0;
VI2(rr)=0;
VI3(rr)=0;
VI4(rr)=0;
VI5(rr)=0;
bound1=1;
Xaxis=linspace(0,1,200);
N=1000; % # of objects
Nstr=num2str(N);
tmin=0.0000005; %frequency of the switch
tlastswitch=now; %time of the last switch for "RLSA"
tlastswitchw=now; %time of the last switch for "CTSA"
tlastswitchf=now; %time of the last switch for "CFSA"
tlastswitchnew=now;%time of the last switch for "PRSA"
ttemp=now;
Vp=-5000000000; %penalty for frequency
minv=1000; %minimal value of VIs witch for switching
maxv=1500; %maximal value of VIs witch for switching
VFA2H=10; % VFA/VH aspect ratio range
VAR=1;
VARstr=num2str(VFA2H(VAR)*10);
if VFA2H(VAR)==0.333
    VARstr=num2str(3);
end

for Pscount=1:1:200;
Ps=Psvector(Pscount);
Psstr=num2str(Ps*100);
dtag=0.6;
dtag=dtag-1; %d' for human (second detector)
Dh=num2str(-dtag*10);
dtagR=0.1; %d' for robot
dtagR=dtagR-1;
Dr=num2str(-dtagR*10);

VH=10;
VM=5;
VCR=3;
VHstr=num2str(VH);
VFA=-VH.*VFA2H(VAR);
Vc=-2;
VCstr=num2str(-Vc);
Vt=-2000/3600;
Vtstr=num2str(-Vt*3600);

respond=1;
tr=0.01;

c3=1;
lnbetar=0;
c2=1;

```

```
Inbetah=0;
c1=1;
Inbetarh=2;
```

```
% the probabilities of the robot
Zsr(c1,c2,c3)=(-2.*Inbetah+dtagR.^2)/(2.*dtagR);
Znr(c1,c2,c3)=(-2.*Inbetah-dtagR.^2)/(2.*dtagR);
phr(c1,c2,c3)=1-normcdf(Zsr(c1,c2,c3));
pfar(c1,c2,c3)=1-normcdf(Znr(c1,c2,c3));
ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));
```

```
if Inbetah==0 & Inbetah==0 & Inbetarh==0
    Zsrtest=(-2.*Inbetah+dtagR.^2)/(2.*dtagR);
    Znrtest=(-2.*Inbetah-dtagR.^2)/(2.*dtagR);
    phrtest=1-normcdf(Zsr(c1,c2,c3));
    pfartest=1-normcdf(Znr(c1,c2,c3));
    ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
    ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));
```

```
end
```

```
% the optimal parameters of the robot if he was a single detector
betastar(c1,c2,c3)=(1-Ps)/Ps.*VFA2H(VAR); % calculating Beta*
ZsRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))+dtagR.^2)/(2.*dtagR);
ZnRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))-dtagR.^2)/(2.*dtagR);
phrstar(c1,c2,c3)=1-normcdf(ZsRstar(c1,c2,c3));
pfarstar(c1,c2,c3)=1-normcdf(ZnRstar(c1,c2,c3));
```

```
% the probabilities of the HO (second detector) for object that the
% robot didn't detect
ZsH(c1,c2,c3)=(-2.*Inbetah+dtag.^2)/(2.*dtag);
ZnH(c1,c2,c3)=(-2.*Inbetah-dtag.^2)/(2.*dtag);
phh(c1,c2,c3)=1-normcdf(ZsH(c1,c2,c3));
pfah(c1,c2,c3)=1-normcdf(ZnH(c1,c2,c3));
```

```
% the probabilities of the HO (second detector) for object that the robot
% already detected
ZsRH(c1,c2,c3)=(-2.*Inbetarh+dtag.^2)/(2.*dtag);
ZnRH(c1,c2,c3)=(-2.*Inbetarh-dtag.^2)/(2.*dtag);
phrh(c1,c2,c3)=1-normcdf(ZsRH(c1,c2,c3));
pfarh(c1,c2,c3)=1-normcdf(ZnRH(c1,c2,c3));
```

```
% the time parameters
```

```
tHh(c1,c2,c3)=5;
tFAh(c1,c2,c3)=5;
tHrh(c1,c2,c3)=5;
tFArh(c1,c2,c3)=5;
```

```
tMh(c1,c2,c3)=5;
tCRh(c1,c2,c3)=5;
tMrh(c1,c2,c3)=5;
tCRrh(c1,c2,c3)=5;
tmotor=2;
```

```
PHs(c1,c2,c3)=phr(c1,c2,c3).*phrh(c1,c2,c3)+(1-phr(c1,c2,c3)).*phh(c1,c2,c3);
VHs(c1,c2,c3)=N.*Ps.*PHs(c1,c2,c3).*VH;
PMs(c1,c2,c3)=phr(c1,c2,c3).*(1-phrh(c1,c2,c3))+(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3));
VMs(c1,c2,c3)=N.*Ps.*PMs(c1,c2,c3).*VM;
FFAs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3);
VFAs(c1,c2,c3)=FFAs(c1,c2,c3).*VFA;
FCRs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3))+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3));
VCRs(c1,c2,c3)=FCRs(c1,c2,c3).*VCR;
```

```
ts(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3).*tHh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*tFArh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*tFAh(c1,c2,c3)...
+N.*Ps.*phr(c1,c2,c3).*(1-phrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;
```

```
tsHORr(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*(tHrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-
phr(c1,c2,c3)).*phh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*(tFArh(c1,c2,c3)+tmotor)+N.*(1-
Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
```

+N.\*Ps.\*phr(c1,c2,c3).\*(1-phrh(c1,c2,c3)).\*tMrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*(1-phh(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*(1-pfarh(c1,c2,c3)).\*tCRrh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*(1-pfah(c1,c2,c3)).\*tCRh(c1,c2,c3)+tr;

tsHOR(c1,c2,c3)=N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3).\*tHrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3).\*(tHh(c1,c2,c3)+tmotor)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3).\*tFArh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3).\*(tFAh(c1,c2,c3)+tmotor)...

+N.\*Ps.\*phr(c1,c2,c3).\*(1-phrh(c1,c2,c3)).\*(tMrh(c1,c2,c3)+tmotor)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*(1-phh(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*(1-pfarh(c1,c2,c3)).\*(tCRrh(c1,c2,c3)+tmotor)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*(1-pfah(c1,c2,c3)).\*tCRh(c1,c2,c3)+tr;

Ndetect(c1,c2,c3)=(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3));

VTs(c1,c2,c3)=ts(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VTsHORr(c1,c2,c3)=tsHORr(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VTsHOR(c1,c2,c3)=tsHOR(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VIs(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTs(c1,c2,c3);

VIsHORr(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHORr(c1,c2,c3);

VIsHOR(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHOR(c1,c2,c3);

PHsR(c1,c2,c3)=phr(c1,c2,c3);

VHsR(c1,c2,c3)=N.\*Ps.\*PHsR(c1,c2,c3).\*VH;

FFAsR(c1,c2,c3)=N.\*(1-Ps).\*pfar(c1,c2,c3);

VFAsR(c1,c2,c3)=FFAsR(c1,c2,c3).\*VFA;

tsR(c1,c2,c3)=tr;

VTsR(c1,c2,c3)=tsR(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3)).\*Vc;

VIsR(c1,c2,c3)=VHsR(c1,c2,c3)+VFAsR(c1,c2,c3)+VTsR(c1,c2,c3);

% the probabilities of the HO collaboration level were taken from the  
 % robot probabilities and the difference between the HO and the R  
 % collaboration levels is just on the times parameters.

PHsHO(c1,c2,c3)=phr(c1,c2,c3);

VHsHO(c1,c2,c3)=N.\*Ps.\*PHsHO(c1,c2,c3).\*VH;

FFAsHO(c1,c2,c3)=N.\*(1-Ps).\*pfar(c1,c2,c3);

VFAsHO(c1,c2,c3)=FFAsHO(c1,c2,c3).\*VFA;

tsHO(c1,c2,c3)=N.\*Ps.\*PHsHO(c1,c2,c3).\*(tHh(c1,c2,c3)+tmotor)+FFAsHO(c1,c2,c3).\*(tFAh(c1,c2,c3)+tmotor)...

+N.\*Ps.\*(1-PHsHO(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*tCRh(c1,c2,c3);

VTsHO(c1,c2,c3)=tsHO(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3)).\*Vc;

VIsHO(c1,c2,c3)=VHsHO(c1,c2,c3)+VFAsHO(c1,c2,c3)+VTsHO(c1,c2,c3);

%-----  
 %switching calculations  
 %-----

CL(Pscount,:)= [VIsHORr(c1,c2,c3) VIsHOR(c1,c2,c3) VIsR(c1,c2,c3) VIsHO(c1,c2,c3)]; % values' matrix of different collaboration levels

[C,CCLtemp]=max(CL(Pscount,:)); % C=value, CCLtemp=location

SV(1,Pscount)=[C]; % switch vector of best collaboration values for each row in CL

VIso=SV(1,Pscount);

VIscl=CL(Pscount,rr);

VIscc=CL(Pscount,CCL);

VIsccw=CL(Pscount,CCLw);

VIsccf=CL(Pscount,CCLf);

VIsccnew=CL(Pscount,CCLnew);

a1='HOR';

a2='HR';

a3='R';

a4='H';

switch CCLtemp

case 1

label1=a1;

case 2

label1=a2;

case 3

label1=a3;

otherwise

label1=a4;

end

```

switch CCL
case 1
    label2=a1;
case 2
    label2=a2;
case 3
    label2=a3;
otherwise
    label2=a4;
end

SVF(1,Pscount)=[CCLtemp];
%-----
%Switch objective fuction for plotting switch points
%-----

VIsutchl(rr,Pscount)=(VIso-VIscl); %+trespond*Vt+Vp*(ttemp+tmin-now)*(Vp*(ttemp+tmin-now)<0); %S1
VIsutch(rr,Pscount)=(VIso-VIsc); %"RLSA" for calculation
VIsutchl1(rr,Pscount)=(VIso-VIsc);%"RLSA" for plot
VIsutchw(rr,Pscount)=(VIso-VIscw); %"CTSA" for calculation
VIsutchw1(rr,Pscount)=(VIso-VIscw); %"CTSA" for plot
VIsutchfw1(rr,Pscount)=(VIso-VIsfc)+Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %"CFSA" for plot
VIsutchfw(rr,Pscount)=(VIso-VIsfc); %"CFSA" for calculation
VIsutchfnew1(rr,Pscount)=(VIso-VIsfcnew)+trespond*Vt+Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
%"PRSA" for plot
VIsutchfnew(rr,Pscount)=(VIso-VIsfcnew); %"PRSA" for calculation

%-----

%-----
%switching algorithm
%-----

%"RLSA" algorithm

if CCLtemp~=CCL
    tcurrent=now;
        if tcurrent-tlastswitch>tmin %checking the frequency

                disp('switch to');
                disp(label1);
                disp('from');
                disp(label2);

                CCL=CCLtemp
                coY(rr,j)=C;
                coX(rr,j)=Pscount;
                coYs(rr,j)=VIsutchl1(rr,Pscount); %for the switching objective function plot
                tlastswitch=tcurrent;
                j=j+1;
                VIsutch(rr,Pscount)=-trespond*Vt; %updating VIsutch as optimal value after switching
        else
            disp('Collaboration level not optimal');
        end
    end

%"CTSA" algorithm

if CCLtemp~=CCLw %optimization for switching
    tcurrent=now;
    if tcurrent-tlastswitch>tmin %checking the frequency
        if VIsutchw(rr,Pscount)>maxv
            CCLw=CCLtemp;
            coYw(l)=C;
            coXw(rr,l)=Pscount;
            coYsw(rr,l)=VIsutchw1(rr,Pscount); %for the switching objective function plot
            tlastswitchw=tcurrent;
            l=l+1;
            VIsutchw(rr,Pscount)=-trespond*Vt; %updating VIsutch as optimal value after switching
        end
    end
end

%"CFSA" algorithm

```

```

if CCLtemp~=CCLf %optimization for switching with frequency
tcurrent=now;
if VIs witchfw1(rr,Pscout)>maxv
CCLf=CCLtemp;
coXf(rr,i)=Pscout;
coYsfw(rr,i)=VIs witchfw1(rr,Pscout);
VIs witchfw(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %updating VI with response
time for switching
i=i+1;
tlastswitchf=tcurrent;
end
end

%”PRSA” algorithm

if CCLtemp~=CCLnew %optimization for switching with frequency
tcurrent=now;
if (tcurrent-tlastswitchnew>tmin)|(VIs witchfnew1(rr,Pscout)>maxv)
% if VIs witchfnew(Pscout)>maxv
bound2=Pscout;
BVF(bound1:bound2)=[CCLtemp];
newVF(bound1:bound2)=SVF(bound1:bound2)==BVF(bound1:bound2);
if sum(newVF(bound1:bound2))>((bound2-bound1)/2)
CCLnew=CCLtemp;

coXfnew(rr,p)=Pscout;
coYsfnew(rr,p)=VIs witchfnew1(rr,Pscout);

p=p+1;
bound1=Pscout+1;

VIs witchfnew(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
tlastswitchnew=tcurrent;
end
end
end

VI1(rr)=VIs witchl(rr,Pscout)+VI1(rr); %regular cost - S1
VI2(rr)=VIs witch(rr,Pscout)+VI2(rr); %cost with switch - “RLSA”
VI3(rr)=VIs witchw(rr,Pscout)+VI3(rr); %cost with switch and maxv filter - “CTSA”
VI4(rr)=VIs witchfw(rr,Pscout)+VI4(rr); %cost with switch and maxv filter after frequency penalty - “CFSA”
VI5(rr)=VIs witchfnew(rr,Pscout)+VI5(rr); %cost with switch and maxv filter after frequency penalty and new algorithm -
“PRSA”

end %Pscout
end %rr
%-----
%Figures
%-----

Figure(1)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);

Figure(2)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');

```

```

set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

```

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

Figure(3)%the score if we stay always at the same collaboration level

```

plot(Xaxis,VIswitchl(1,:),'-',Xaxis,VIswitchl(2,:),'--',Xaxis,VIswitchl(3,:),'.',Xaxis,VIswitchl(4,:),'-.')
xlabel('time');
ylabel('VIswitch');
title('S1');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

```

plotvl=2;

Figure(4)

```

subplot(3,1,1)
plot(Xaxis,VIswitchl(plotvl,:))
xlabel('time');
ylabel('VIswitch');
title('“RLSA”');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coX(plotvl,find(coX(plotvl,:))))),coYs(plotvl,find(coYs(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitch(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitch(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

Figure(5)

```

subplot(3,1,1)
plot(Xaxis,VIswitchw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title('“CTSA”');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXw(plotvl,find(coXw(plotvl,:))))),coYsw(plotvl,find(coYsw(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchw(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchw(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

Figure(6)

```

subplot(3,1,1)
plot(Xaxis,VIswitchfw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title('“CFSA”');

```

```

set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXf(plotvl,find(coXf(plotvl,:))))),coYsfw(plotvl,find(coYsfw(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfw(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfw(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(7)
subplot(3,1,1)
plot(Xaxis,VIswitchfnew1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title("PRSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXfnew(plotvl,find(coXfnew(plotvl,:))))),coYsfnew(plotvl,find(coYsfnew(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(8)
subplot(2,1,1)
Xaxis=[1 2 3 4 5];
Yaxis=[(VI1-VI2)' (VI1-VI3)' (VI1-VI4)' (VI1-VI5)'];
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA','CTSA','CFSA','PRSA'})
xlabel('type of algorithm');
ylabel('VI gain');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

subplot(2,1,2)
Xaxis=[1 2 3 4 5];
Yaxis=[((VI1-VI2)/VI1)' ((VI1-VI3)/VI1)' ((VI1-VI4)/VI1)' ((VI1-VI5)/VI1)'];
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA','CTSA','CFSA','PRSA'})
xlabel('type of algorithm');
ylabel('VI gain in percentage');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

```

## Simulation for random uniform distribution of Ps:

```
%This program performs dynamic switching at random uniform distribution of Ps parameters
% the program calculates the hit and false alarm probabilities and the
% value of the operational cost according to betas and dtags
% This version will show the objective function values for all collaboration
% levels and for all betas combinations.
```

```
clc
close all
clear all
Psvector=rand(1,200);
for rr=1:4
j=1;
i=1;
l=1;
p=1;
CCL=rr %current collaboration level
CCLw=rr;
CCLf=rr;
CCLnew=rr;
V1(rr)=0;
V2(rr)=0;
V3(rr)=0;
V4(rr)=0;
V5(rr)=0;
bound1=1;
Xaxis=linspace(0,1,200);
N=1000; % # of objects
Nstr=num2str(N);
tmin=0.0000005; %frequency of the switch
tlastswitch=now; %time of the last switch for "RLSA"
tlastswitchw=now; %time of the last switch for "CTSA"
tlastswitchf=now; %time of the last switch for "CFSA"
tlastswitchnew=now;%time of the last switch for "PRSA"
ttemp=now;
Vp=-5000000000; %penalty for frequency
minv=1000; %minimal value of VIswitcch for switching
maxv=1500; %maximal value of VIswitcch for switching
VFA2H=10; % VFA/VH aspect ratio range
VAR=1;
VARstr=num2str(VFA2H(VAR)*10);
if VFA2H(VAR)==0.333
    VARstr=num2str(3);
end

for Pscount=1:1:200;
Ps=Psvector(Pscount);
Psstr=num2str(Ps*100);
dtag=0.6;
dtag=dtag-1; %d' for human (second detector)
Dh=num2str(-dtag*10);
dtagR=0.1; %d' for robot
dtagR=dtagR-1;
Dr=num2str(-dtagR*10);

VH=10;
VM=5;
VCR=3;
VHstr=num2str(VH);
VFA=-VH.*VFA2H(VAR);
Vc=-2;
VCstr=num2str(-Vc);
Vt=-2000/3600;
Vtstr=num2str(-Vt*3600);

trespond=1;
tr=0.01;

c3=1;
lnbetar=0;
c2=1;
lnbetah=0;
c1=1;
lnbetarh=2;
```



```

% the probabilities of the robot
Zsr(c1,c2,c3)=(-2.*Inbetar+dtagR.^2)/(2.*dtagR);
Znr(c1,c2,c3)=(-2.*Inbetar-dtagR.^2)/(2.*dtagR);
pfr(c1,c2,c3)=1-normcdf(Zsr(c1,c2,c3));
pfar(c1,c2,c3)=1-normcdf(Znr(c1,c2,c3));
ratio1=pfar(c1,c2,c3)/pfr(c1,c2,c3);
ratio2=(1-pfar(c1,c2,c3))/(1-pfr(c1,c2,c3));

if Inbetar==0 & Inbetah==0 & Inbetarh==0
    Zsrtest=(-2.*Inbetar+dtagR.^2)/(2.*dtagR);
Znrtest=(-2.*Inbetar-dtagR.^2)/(2.*dtagR);
pfrtest=1-normcdf(Zsr(c1,c2,c3));
pfartest=1-normcdf(Znr(c1,c2,c3));
ratio1=pfar(c1,c2,c3)/pfr(c1,c2,c3);
ratio2=(1-pfar(c1,c2,c3))/(1-pfr(c1,c2,c3));

end

% the optimal parameters of the robot if he was a single detector
betastar(c1,c2,c3)=((1-Ps)/Ps).*VFA2H(VAR); % calculating Beta*
ZsRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))+dtagR.^2)/(2.*dtagR);
ZnRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))-dtagR.^2)/(2.*dtagR);
pfrstar(c1,c2,c3)=1-normcdf(ZsRstar(c1,c2,c3));
pfarstar(c1,c2,c3)=1-normcdf(ZnRstar(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the
% robot didn't detect
ZsH(c1,c2,c3)=(-2.*Inbetah+dtag.^2)/(2.*dtag);
ZnH(c1,c2,c3)=(-2.*Inbetah-dtag.^2)/(2.*dtag);
phh(c1,c2,c3)=1-normcdf(ZsH(c1,c2,c3));
pfah(c1,c2,c3)=1-normcdf(ZnH(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the robot
% already detected
ZsRH(c1,c2,c3)=(-2.*Inbetarh+dtag.^2)/(2.*dtag);
ZnRH(c1,c2,c3)=(-2.*Inbetarh-dtag.^2)/(2.*dtag);
phrh(c1,c2,c3)=1-normcdf(ZsRH(c1,c2,c3));
pfarh(c1,c2,c3)=1-normcdf(ZnRH(c1,c2,c3));

% the time parameters

tHh(c1,c2,c3)=5;
tFAh(c1,c2,c3)=5;
tHrh(c1,c2,c3)=5;
tFARh(c1,c2,c3)=5;

tMh(c1,c2,c3)=5;
tCRh(c1,c2,c3)=5;
tMrh(c1,c2,c3)=5;
tCRrh(c1,c2,c3)=5;
tmotor=2;

PHs(c1,c2,c3)=pfr(c1,c2,c3).*phrh(c1,c2,c3)+(1-pfr(c1,c2,c3)).*phh(c1,c2,c3);
VHs(c1,c2,c3)=N.*Ps.*PHs(c1,c2,c3).*VH;
PMs(c1,c2,c3)=pfr(c1,c2,c3).*pfrh(c1,c2,c3)+(1-pfr(c1,c2,c3)).*phh(c1,c2,c3);
VMs(c1,c2,c3)=N.*Ps.*PMs(c1,c2,c3).*VM;
FFAs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*pfar(c1,c2,c3).*pfah(c1,c2,c3);
VFAs(c1,c2,c3)=FFAs(c1,c2,c3).*VFA;
FCRs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).(1-pfarh(c1,c2,c3))+N.*(1-Ps).(1-pfar(c1,c2,c3)).(1-pfah(c1,c2,c3));
VCRs(c1,c2,c3)=FCRs(c1,c2,c3).*VCR;

ts(c1,c2,c3)=N.*Ps.*pfr(c1,c2,c3).*phrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-pfr(c1,c2,c3)).*phh(c1,c2,c3).*tHh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*tFARh(c1,c2,c3)+N.*(1-Ps).(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*tFAh(c1,c2,c3)...
+N.*Ps.*pfr(c1,c2,c3).(1-pfrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-pfr(c1,c2,c3)).(1-phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).(1-pfarh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).(1-pfar(c1,c2,c3)).(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

tsHORr(c1,c2,c3)=N.*Ps.*pfr(c1,c2,c3).*phrh(c1,c2,c3).(tHrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-
pfr(c1,c2,c3)).*phh(c1,c2,c3).(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).(tFARh(c1,c2,c3)+tmotor)+N.*(1-
Ps).(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*pfr(c1,c2,c3).(1-pfrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-pfr(c1,c2,c3)).(1-phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).(1-pfarh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).(1-pfar(c1,c2,c3)).(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

```

```

tsHOR(c1,c2,c3)=N.*Ps.*pfr(c1,c2,c3).*phrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-
pfr(c1,c2,c3)).*phh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*tFArh(c1,c2,c3)+N.*(1-Ps).*(1-
pfr(c1,c2,c3)).*pfah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*pfr(c1,c2,c3).*(1-phrh(c1,c2,c3)).*(tMrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-phr(c1,c2,c3)).*(1-
phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3)).*(tCRrh(c1,c2,c3)+tmotor)+N.*(1-Ps).*(1-
pfr(c1,c2,c3)).*(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

```

```

Ndetect(c1,c2,c3)=(N.*Ps.*pfr(c1,c2,c3).*phrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3));
VTs(c1,c2,c3)=ts(c1,c2,c3).*Vt+(N.*Ps.*pfr(c1,c2,c3).*phrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3)).*Vc;
VTsHORr(c1,c2,c3)=tsHORr(c1,c2,c3).*Vt+(N.*Ps.*pfr(c1,c2,c3).*phrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3)).*Vc;
VTsHOR(c1,c2,c3)=tsHOR(c1,c2,c3).*Vt+(N.*Ps.*pfr(c1,c2,c3).*phrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3)).*Vc;
VIs(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTs(c1,c2,c3);
VIsHORr(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHORr(c1,c2,c3);
VIsHOR(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHOR(c1,c2,c3);

```

```

PHsR(c1,c2,c3)=pfr(c1,c2,c3);
VHsR(c1,c2,c3)=N.*Ps.*PHsR(c1,c2,c3).*VH;
FFAsR(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3);
VFAsR(c1,c2,c3)=FFAsR(c1,c2,c3).*VFA;
tsR(c1,c2,c3)=tr;
VTsR(c1,c2,c3)=tsR(c1,c2,c3).*Vt+(N.*Ps.*pfr(c1,c2,c3)+N.*(1-Ps).*pfar(c1,c2,c3)).*Vc;
VIsR(c1,c2,c3)=VHsR(c1,c2,c3)+VFAsR(c1,c2,c3)+VTsR(c1,c2,c3);

```

```

% the probabilities of the HO collaboration level were taken from the
% robot probabilities and the difference between the HO and the R
% collaboration levels is just on the times parameters.

```

```

PHsHO(c1,c2,c3)=pfr(c1,c2,c3);
VHsHO(c1,c2,c3)=N.*Ps.*PHsHO(c1,c2,c3).*VH;
FFAsHO(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3);
VFAsHO(c1,c2,c3)=FFAsHO(c1,c2,c3).*VFA;
tsHO(c1,c2,c3)=N.*Ps.*PHsHO(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+FFAsHO(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*(1-PHsHO(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*tCRh(c1,c2,c3);
VTsHO(c1,c2,c3)=tsHO(c1,c2,c3).*Vt+(N.*Ps.*pfr(c1,c2,c3)+N.*(1-Ps).*pfar(c1,c2,c3)).*Vc;
VIsHO(c1,c2,c3)=VHsHO(c1,c2,c3)+VFAsHO(c1,c2,c3)+VTsHO(c1,c2,c3);

```

```

%-----
%switching calculations
%-----

```

```

CL(Pscount,:)= [VIsHORr(c1,c2,c3) VIsHOR(c1,c2,c3) VIsR(c1,c2,c3) VIsHO(c1,c2,c3)]; % values' matrix of different collaboration
levels
[C,CCLtemp]=max(CL(Pscount,:)); %C=value, CCLtemp=location
SV(1,Pscount)= [C]; %switch vector of best collaboration values for each row in CL
VIsO=SV(1,Pscount);
VIscl=CL(Pscount,rr);
VIsC=CL(Pscount,CCL);
VIsCw=CL(Pscount,CCLw);
VIsCf=CL(Pscount,CCLf);
VIsCnew=CL(Pscount,CCLnew);

```

```

a1='HOR';
a2='HR';
a3='R';
a4='H';

```

```

switch CCLtemp
case 1
    label1=a1;
case 2
    label1=a2;
case 3
    label1=a3;
otherwise
    label1=a4;
end

```

```

switch CCL
case 1

```

```

    label2=a1;
case 2
    label2=a2;
case 3
    label2=a3;
otherwise
    label2=a4;
end

SVF(1,Pscount)=[CCLtemp];
%-----
%Switch objective fuction for plotting switch points
%-----

VIs witchl(rr,Pscount)=(V Iso-VI scl); % +trespond*Vt+Vp*(ttemp+tmin-now)*(Vp*(ttemp+tmin-now)<0); %S1
VIs witch(rr,Pscount)=(V Iso-VI sc); %"RLSA" for calculation
VIs witchl(rr,Pscount)=(V Iso-VI sc);%"RLSA" for plot
VIs witchw(rr,Pscount)=(V Iso-VI scw); %"CTSA" for calculation
VIs witchw l(rr,Pscount)=(V Iso-VI scw); %"CTSA" for plot
VIs witchfw l(rr,Pscount)=(V Iso-VI scf)+Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %"CFSA" for plot
VIs witchfw(rr,Pscount)=(V Iso-VI scf); %"CFSA" for calculation
VIs witchfnw l(rr,Pscount)=(V Iso-VI scnew)+trespond*Vt+Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
%"PRSA" for plot
VIs witchfnw(rr,Pscount)=(V Iso-VI scnew); %"PRSA" for calculation

%-----

%-----
%switching algorithm
%-----

%"RLSA" algorithm
if CCLtemp~=-CCL
tcurrent=now;
    if tcurrent-tlastswitch>tmin %checking the frequency

        disp('switch to');
        disp(label1);
        disp('from');
        disp(label2);

        CCL=CCLtemp
        coY(rr,j)=C;
        coX(rr,j)=Pscount;
        coYs(rr,j)=VIs witchl(rr,Pscount); %for the switching objective function plot
        tlastswitch=tcurrent;
        j=j+1;
        VIs witch(rr,Pscount)=-trespond*Vt; %updating VIs witch as optimal value after switching
    else
        disp('Collaboration level not optimal');
    end
end

%"CTSA" algorithm
if CCLtemp~=-CCLw %optimization for switching
tcurrent=now;
if tcurrent-tlastswitchw>tmin %checking the frequency
    if VIs witchw(rr,Pscount)>maxv
        CCLw=CCLtemp;
        coYw(l)=C;
        coXw(rr,l)=Pscount;
        coYsw(rr,l)=VIs witchw l(rr,Pscount); %for the switching objective function plot
        tlastswitchw=tcurrent;
        l=l+1;
        VIs witchw(rr,Pscount)=-trespond*Vt; %updating VIs witchw as optimal value after switching
    end
end
end

%"CFSA" algorithm
if CCLtemp~=-CCLf %optimization for switching with frequency
tcurrent=now;
    if VIs witchfw l(rr,Pscount)>maxv

```

```

    CCLf=CCLtemp;
    coXf(rr,i)=Pscount;
    coYsfw(rr,i)=VIschwfw1(rr,Pscount);
    VIschwfw(rr,Pscount)=-trespond*Vt-Vp*(tlastswchf+tmin-now)*(Vp*(tlastswchf+tmin-now)<0); %updating VI with response
time for switching
    i=i+1;
    tlastswchf=tcurent;
end
end

%”PRSA” algorithm

if CCLtemp~=-CCLnew %optimization for switching with frequency
    tcurent=now;
    if (tcurent-tlastswchnew>tmin)|(VIschwfnw1(rr,Pscount)>maxv)
    % if VIschwfnw(Pscount)>maxv
        bound2=Pscount;
        BVF(bound1:bound2)=[CCLtemp];
        newVF(bound1:bound2)=SVF(bound1:bound2)==BVF(bound1:bound2);
        if sum(newVF(bound1:bound2))>((bound2-bound1)/2)
            CCLnew=CCLtemp;

            coXfnw(rr,p)=Pscount;
            coYsfnw(rr,p)=VIschwfnw1(rr,Pscount);

            p=p+1;
            bound1=Pscount+1;

            VIschwfnw(rr,Pscount)=-trespond*Vt-Vp*(tlastswchnew+tmin-now)*(Vp*(tlastswchnew+tmin-now)<0);
            tlastswchnew=tcurent;
        end
    end
end

VI1(rr)=VIschw1(rr,Pscount)+VI1(rr); %regular cost - S1
VI2(rr)=VIschw(rr,Pscount)+VI2(rr); %cost with switch - “RLSA”
VI3(rr)=VIschwrr(rr,Pscount)+VI3(rr); %cost with switch and maxv filter - “CTSA”
VI4(rr)=VIschwfw(rr,Pscount)+VI4(rr); %cost with switch and maxv filter after frequency penalty - “CFSA”
VI5(rr)=VIschwfnw(rr,Pscount)+VI5(rr); %cost with switch and maxv filter after frequency penalty and new algorithm -
“PRSA”

end %Pscount
end %rr
%-----
%Figures
%-----

Figure(1)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);

Figure(2)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

```

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

Figure(3)%the score if we stay always at the same collaboration level

```

plot(Xaxis,VIs witchl(1,:),'-',Xaxis,VIs witchl(2,:),'--',Xaxis,VIs witchl(3,:),'.',Xaxis,VIs witchl(4,:),'-')
xlabel('time');
ylabel('VIs witch');
title('S1');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

```

```

plotvl=2;

```

Figure(4)

```

subplot(3,1,1)
plot(Xaxis,VIs witchl(plotvl,:))
xlabel('time');
ylabel('VIs witch');
title('RLSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coX(plotvl,find(coX(plotvl,:))))),coYs(plotvl,find(coYs(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIs witchl(plotvl,:)-VIs witch(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIs witchl(plotvl,:)-VIs witch(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

Figure(5)

```

subplot(3,1,1)
plot(Xaxis,VIs witchw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIs witch');
title('CTSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXw(plotvl,find(coXw(plotvl,:))))),coYsw(plotvl,find(coYsw(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIs witchl(plotvl,:)-VIs witchw(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIs witchl(plotvl,:)-VIs witchw(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

Figure(6)

```

subplot(3,1,1)
plot(Xaxis,VIs witchfw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIs witch');
title('CFSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXf(plotvl,find(coXf(plotvl,:))))),coYsfw(plotvl,find(coYsfw(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');

```

```

grid on

subplot(3,1,2)
plot(Xaxis,VIsitchl(plotlv1,:)-VIsitchfw(plotlv1,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIsitchl(plotlv1,:)-VIsitchfw(plotlv1,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(7)
subplot(3,1,1)
plot(Xaxis,VIsitchfnew1(plotlv1,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIsitch');
title("PRSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXfnew(plotlv1,find(coXfnew(plotlv1,:))))),coYsfnew(plotlv1,find(coYsfnew(plotlv1,:))), 'x','FontSize',20,'color','r','HorizontalAlig
nment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIsitchl(plotlv1,:)-VIsitchfnew(plotlv1,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIsitchl(plotlv1,:)-VIsitchfnew(plotlv1,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(8)
subplot(2,1,1)
Xaxis=[1 2 3 4 5];
Yaxis=[(VI1-VI2)' (VI1-VI3)' (VI1-VI4)' (VI1-VI5)']';
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA',"CTSA", "CFSA", "PRSA"})
xlabel('type of algorithm');
ylabel('VI gain');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

subplot(2,1,2)
Xaxis=[1 2 3 4 5];
Yaxis=[((VI1-VI2)/VI1)' ((VI1-VI3)/VI1)' ((VI1-VI4)/VI1)' ((VI1-VI5)/VI1)']';
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA',"CTSA", "CFSA", "PRSA"})
xlabel('type of algorithm');
ylabel('VI gain in percentage');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

```

## Simulation for ascending normal distribution of Ps:

```
%This program performs dynamic switching at ascending normal distribution of Ps parameters
% the program calculates the hit and false alarm probabilities and the
% value of the operational cost according to betas and dtags
% This version will show the objective function values for all collaboration
% levels and for all betas combinations.
```

```
clc
close all
clear all
Psvector=randn(1,200);
ctmp=max(abs(Psvector));
Psvector=Psvector./(ctmp*2);
Psvector=abs(Psvector+0.49);
Psvector=sort(Psvector);
for rr=1:4
j=1;
i=1;
l=1;
p=1;
CCL=rr %current collaboration level
CCLw=rr;
CCLf=rr;
CCLnew=rr;
VI1(rr)=0;
VI2(rr)=0;
VI3(rr)=0;
VI4(rr)=0;
VI5(rr)=0;
bound1=1;
Xaxis=linspace(0,1,200);
N=1000; % # of objects
Nstr=num2str(N);
tmin=0.0000005; %frequency of the switch
tlastswitch=now; %time of the last switch for "RLSA"
tlastswitchw=now; %time of the last switch for "CTSA"
tlastswitchf=now; %time of the last switch for "CFSA"
tlastswitchnew=now;%time of the last switch for "PRSA"
ttemp=now;
Vp=-5000000000; %penalty for frequency
minv=1000; %minimal value of VIs witch for switching
maxv=1500; %maximal value of VIs witch for switching
VFA2H=10; % VFA/VH aspect ratio range
VAR=1;
VARstr=num2str(VFA2H(VAR)*10);
if VFA2H(VAR)==0.333
    VARstr=num2str(3);
end

for Pscount=1:1:200;
Ps=Psvector(Pscount);
Psstr=num2str(Ps*100);
dtag=0.6;
dtag=dtag-1; % d' for human (second detector)
Dh=num2str(-dtag*10);
dtagR=0.1; % d' for robot
dtagR=dtagR-1;
Dr=num2str(-dtagR*10);

VH=10;
VM=5;
VCR=3;
VHstr=num2str(VH);
VFA=-VH.*VFA2H(VAR);
Vc=-2;
VCstr=num2str(-Vc);
Vt=-2000/3600;
Vtstr=num2str(-Vt*3600);

trespond=1;
tr=0.01;

c3=1;
lnbetar=0;
```

```

c2=1;
lnbetah=0;
c1=1;
lnbetarh=2;

% the probabilities of the robot
Zsr(c1,c2,c3)=(-2.*lnbetar+dtagR.^2)/(2.*dtagR);
Znr(c1,c2,c3)=(-2.*lnbetar-dtagR.^2)/(2.*dtagR);
phr(c1,c2,c3)=1-normcdf(Zsr(c1,c2,c3));
pfar(c1,c2,c3)=1-normcdf(Znr(c1,c2,c3));
ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));

if lnbetar==0 & lnbetah==0 & lnbetarh==0
    Zsrtest=(-2.*lnbetar+dtagR.^2)/(2.*dtagR);
    Znrtest=(-2.*lnbetar-dtagR.^2)/(2.*dtagR);
    phrtest=1-normcdf(Zsr(c1,c2,c3));
    pfartest=1-normcdf(Znr(c1,c2,c3));
    ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
    ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));
end

% the optimal parameters of the robot if he was a single detector
betastar(c1,c2,c3)=(1-Ps)/Ps.*VFA2H(VAR); % calculating Beta*
ZsRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))+dtagR.^2)/(2.*dtagR);
ZnRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))-dtagR.^2)/(2.*dtagR);
phrstar(c1,c2,c3)=1-normcdf(ZsRstar(c1,c2,c3));
pfarstar(c1,c2,c3)=1-normcdf(ZnRstar(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the
% robot didn't detect
ZsH(c1,c2,c3)=(-2.*lnbetah+dtag.^2)/(2.*dtag);
ZnH(c1,c2,c3)=(-2.*lnbetah-dtag.^2)/(2.*dtag);
phh(c1,c2,c3)=1-normcdf(ZsH(c1,c2,c3));
pfah(c1,c2,c3)=1-normcdf(ZnH(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the robot
% already detected
ZsRH(c1,c2,c3)=(-2.*lnbetarh+dtag.^2)/(2.*dtag);
ZnRH(c1,c2,c3)=(-2.*lnbetarh-dtag.^2)/(2.*dtag);
phrh(c1,c2,c3)=1-normcdf(ZsRH(c1,c2,c3));
pfarh(c1,c2,c3)=1-normcdf(ZnRH(c1,c2,c3));

% the time parameters
tHh(c1,c2,c3)=5;
tFAh(c1,c2,c3)=5;
tHrh(c1,c2,c3)=5;
tFArh(c1,c2,c3)=5;

tMh(c1,c2,c3)=5;
tCRh(c1,c2,c3)=5;
tMrh(c1,c2,c3)=5;
tCRrh(c1,c2,c3)=5;
tmotor=2;

PHs(c1,c2,c3)=phr(c1,c2,c3).*phrh(c1,c2,c3)+(1-phr(c1,c2,c3)).*phh(c1,c2,c3);
VHs(c1,c2,c3)=N.*Ps.*PHs(c1,c2,c3).*VH;
PMs(c1,c2,c3)=phr(c1,c2,c3).*(1-phrh(c1,c2,c3))+(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3));
VMs(c1,c2,c3)=N.*Ps.*PMs(c1,c2,c3).*VM;
FFAs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3);
VFAs(c1,c2,c3)=FFAs(c1,c2,c3).*VFA;
FCRs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3))+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3));
VCRs(c1,c2,c3)=FCRs(c1,c2,c3).*VCR;

ts(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3).*tHh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*tFArh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*tFAh(c1,c2,c3)...
+N.*Ps.*phr(c1,c2,c3).*(1-phrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

tsHORr(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*(tHrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-
phr(c1,c2,c3)).*phh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*(tFArh(c1,c2,c3)+tmotor)+N.*(1-
Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...

```



+N.\*Ps.\*phr(c1,c2,c3).\*(1-phrh(c1,c2,c3)).\*tMrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*(1-phh(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*(1-pfarh(c1,c2,c3)).\*tCRrh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*(1-pfah(c1,c2,c3)).\*tCRh(c1,c2,c3)+tr;

tsHOR(c1,c2,c3)=N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3).\*tHrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3).\*(tHh(c1,c2,c3)+tmotor)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3).\*tFArh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3).\*(tFAh(c1,c2,c3)+tmotor)...

+N.\*Ps.\*phr(c1,c2,c3).\*(1-phrh(c1,c2,c3)).\*(tMrh(c1,c2,c3)+tmotor)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*(1-phh(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*(1-pfarh(c1,c2,c3)).\*(tCRrh(c1,c2,c3)+tmotor)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*(1-pfah(c1,c2,c3)).\*tCRh(c1,c2,c3)+tr;

Ndetect(c1,c2,c3)=(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3));

VTs(c1,c2,c3)=ts(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VTsHORr(c1,c2,c3)=tsHORr(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VTsHOR(c1,c2,c3)=tsHOR(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3).\*phrh(c1,c2,c3)+N.\*Ps.\*(1-phr(c1,c2,c3)).\*phh(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3).\*pfarh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*pfah(c1,c2,c3)).\*Vc;

VIs(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTs(c1,c2,c3);

VIsHORr(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHORr(c1,c2,c3);

VIsHOR(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHOR(c1,c2,c3);

PHsR(c1,c2,c3)=phr(c1,c2,c3);

VHsR(c1,c2,c3)=N.\*Ps.\*PHsR(c1,c2,c3).\*VH;

FFAsR(c1,c2,c3)=N.\*(1-Ps).\*pfar(c1,c2,c3);

VFAsR(c1,c2,c3)=FFAsR(c1,c2,c3).\*VFA;

tsR(c1,c2,c3)=tr;

VTsR(c1,c2,c3)=tsR(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3)).\*Vc;

VIsR(c1,c2,c3)=VHsR(c1,c2,c3)+VFAsR(c1,c2,c3)+VTsR(c1,c2,c3);

% the probabilities of the HO collaboration level were taken from the  
% robot probabilities and the difference between the HO and the R  
% collaboration levels is just on the times parameters.

PHsHO(c1,c2,c3)=phr(c1,c2,c3);

VHsHO(c1,c2,c3)=N.\*Ps.\*PHsHO(c1,c2,c3).\*VH;

FFAsHO(c1,c2,c3)=N.\*(1-Ps).\*pfar(c1,c2,c3);

VFAsHO(c1,c2,c3)=FFAsHO(c1,c2,c3).\*VFA;

tsHO(c1,c2,c3)=N.\*Ps.\*PHsHO(c1,c2,c3).\*(tHh(c1,c2,c3)+tmotor)+FFAsHO(c1,c2,c3).\*(tFAh(c1,c2,c3)+tmotor)...

+N.\*Ps.\*(1-PHsHO(c1,c2,c3)).\*tMh(c1,c2,c3)+N.\*(1-Ps).\*(1-pfar(c1,c2,c3)).\*tCRh(c1,c2,c3);

VTsHO(c1,c2,c3)=tsHO(c1,c2,c3).\*Vt+(N.\*Ps.\*phr(c1,c2,c3)+N.\*(1-Ps).\*pfar(c1,c2,c3)).\*Vc;

VIsHO(c1,c2,c3)=VHsHO(c1,c2,c3)+VFAsHO(c1,c2,c3)+VTsHO(c1,c2,c3);

%-----  
%switching calculations  
%-----

CL(Pscount,:)= [VIsHORr(c1,c2,c3) VIsHOR(c1,c2,c3) VIsR(c1,c2,c3) VIsHO(c1,c2,c3)]; % values' matrix of different collaboration levels

[C,CCLtemp]=max(CL(Pscount,:)); %C=value, CCLtemp=location

SV(1,Pscount)=[C]; %switch vector of best collaboration values for each row in CL

VIso=SV(1,Pscount);

VIscl=CL(Pscount,rr);

VIscl=CL(Pscount,CCL);

VIscl=CL(Pscount,CCLw);

VIscl=CL(Pscount,CCLf);

VIscl=CL(Pscount,CCLnew);

a1='HOR';

a2='HR';

a3='R';

a4='H';

switch CCLtemp

case 1

label1=a1;

case 2

label1=a2;

case 3

label1=a3;

otherwise

label1=a4;

end

```

switch CCL
case 1
    label2=a1;
case 2
    label2=a2;
case 3
    label2=a3;
otherwise
    label2=a4;
end

SVF(1,Pscount)=[CCLtemp];
%-----
%Switch objective fuction for plotting switch points
%-----

VIsutchl(rr,Pscount)=(VIso-VIscl); %+trespond*Vt+Vp*(ttemp+tmin-now)*(Vp*(ttemp+tmin-now)<0); %S1
VIsutch(rr,Pscount)=(VIso-VIscl); %"RLSA" for calculation
VIsutchl1(rr,Pscount)=(VIso-VIscl);%"RLSA" for plot
VIsutchw(rr,Pscount)=(VIso-VIschw); %"CTSA" for calculation
VIsutchw1(rr,Pscount)=(VIso-VIschw); %"CTSA" for plot
VIsutchfw1(rr,Pscount)=(VIso-VIschw)+Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %"CFSA" for plot
VIsutchfw(rr,Pscount)=(VIso-VIschw); %"CFSA" for calculation
VIsutchfnew1(rr,Pscount)=(VIso-VIschwnew)+trespond*Vt+Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
%"PRSA" for plot
VIsutchfnew(rr,Pscount)=(VIso-VIschwnew); %"PRSA" for calculation

%-----

%-----
%switching algorithm
%-----

%"RLSA" algorithm

if CCLtemp~=CCL
    tcurrent=now;
        if tcurrent-tlastswitch>tmin %checking the frequency

                disp('switch to');
                disp(label1);
                disp('from');
                disp(label2);

                CCL=CCLtemp
                coY(rr,j)=C;
                coX(rr,j)=Pscount;
                coYs(rr,j)=VIsutchl1(rr,Pscount); %for the switching objective function plot
                tlastswitch=tcurrent;
                j=j+1;
                VIsutch(rr,Pscount)=-trespond*Vt; %updating VIsutch as optimal value after switching
        else
            disp('Collaboration level not optimal');
        end
    end

%"CTSA" algorithm

if CCLtemp~=CCLw %optimization for switching
    tcurrent=now;
    if tcurrent-tlastswitch>tmin %checking the frequency
        if VIsutchw(rr,Pscount)>maxv
            CCLw=CCLtemp;
            coYw(l)=C;
            coXw(rr,l)=Pscount;
            coYsw(rr,l)=VIsutchw1(rr,Pscount); %for the switching objective function plot
            tlastswitchw=tcurrent;
            l=l+1;
            VIsutchw(rr,Pscount)=-trespond*Vt; %updating VIsutch as optimal value after switching
        end
    end
end

%"CFSA" algorithm

```

```

if CCLtemp~=CCLf %optimization for switching with frequency
tcurrent=now;
if VIs witchfw1(rr,Pscout)>maxv
CCLf=CCLtemp;
coXf(rr,i)=Pscout;
coYsfw(rr,i)=VIs witchfw1(rr,Pscout);
VIs witchfw(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %updating VI with response
time for switching
i=i+1;
tlastswitchf=tcurrent;
end
end

%”PRSA” algorithm

if CCLtemp~=CCLnew %optimization for switching with frequency
tcurrent=now;
if (tcurrent-tlastswitchnew>tmin)|(VIs witchfnew1(rr,Pscout)>maxv)
% if VIs witchfnew(Pscout)>maxv
bound2=Pscout;
BVF(bound1:bound2)=[CCLtemp];
newVF(bound1:bound2)=SVF(bound1:bound2)==BVF(bound1:bound2);
if sum(newVF(bound1:bound2))>((bound2-bound1)/2)
CCLnew=CCLtemp;

coXfnew(rr,p)=Pscout;
coYsfnew(rr,p)=VIs witchfnew1(rr,Pscout);

p=p+1;
bound1=Pscout+1;

VIs witchfnew(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
tlastswitchnew=tcurrent;
end
end
end

VI1(rr)=VIs witchl(rr,Pscout)+VI1(rr); %regular cost - S1
VI2(rr)=VIs witch(rr,Pscout)+VI2(rr); %cost with switch - “RLSA”
VI3(rr)=VIs witchw(rr,Pscout)+VI3(rr); %cost with switch and maxv filter - “CTSA”
VI4(rr)=VIs witchfw(rr,Pscout)+VI4(rr); %cost with switch and maxv filter after frequency penalty - “CFSA”
VI5(rr)=VIs witchfnew(rr,Pscout)+VI5(rr); %cost with switch and maxv filter after frequency penalty and new algorithm -
“PRSA”

end %Pscout
end %rr
%-----
%Figures
%-----

Figure(1)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);

Figure(2)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.-')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');

```

```

set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

```

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.',Xaxis,SV,'black')
xlabel('time');
ylabel('VI');
title('Dynamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

```

Figure(3)%the score if we stay always at the same collaboration level

```

plot(Xaxis,VIswitchl(1,:),'-',Xaxis,VIswitchl(2,:),'--',Xaxis,VIswitchl(3,:),'.',Xaxis,VIswitchl(4,:),'-.')
xlabel('time');
ylabel('VIswitch');
title('S1');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

```

plotvl=2;

Figure(4)

```

subplot(3,1,1)
plot(Xaxis,VIswitchl(plotvl,:))
xlabel('time');
ylabel('VIswitch');
title("“RLSA”");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coX(plotvl,find(coX(plotvl,:))))),coYs(plotvl,find(coYs(plotvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl:)-VIswitch(plotvl:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl:)-VIswitch(plotvl:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

Figure(5)

```

subplot(3,1,1)
plot(Xaxis,VIswitchw1(plotvl:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title("“CTSA”");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXw(plotvl,find(coXw(plotvl:))))),coYsw(plotvl,find(coYsw(plotvl:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

```

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl:)-VIswitchw(plotvl:))
grid on
xlabel('time');
ylabel('Gain');

```

```

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl:)-VIswitchw(plotvl:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

```

Figure(6)

```

subplot(3,1,1)
plot(Xaxis,VIswitchfw1(plotvl:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title("“CFSA”");

```

```

set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXf(plotvl,find(coXf(plotvl,:))))),coYsfw(plotvl,find(coYsfw(plotvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfw(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfw(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(7)
subplot(3,1,1)
plot(Xaxis,VIswitchfnew1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time');
ylabel('VIswitch');
title("PRSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXfnew(plotvl,find(coXfnew(plotvl,:))))),coYsfnew(plotvl,find(coYsfnew(plotvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:))
grid on
xlabel('time');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:)))
grid on
xlabel('time');
ylabel('Cumulative gain');

Figure(8)
subplot(2,1,1)
Xaxis=[1 2 3 4 5];
Yaxis=[(VI1-VI2)' (VI1-VI3)' (VI1-VI4)' (VI1-VI5)']';
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA','CTSA','CFSA','PRSA'})
xlabel('type of algorithm');
ylabel('VI gain');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

subplot(2,1,2)
Xaxis=[1 2 3 4 5];
Yaxis=[((VI1-VI2)/VI1)' ((VI1-VI3)/VI1)' ((VI1-VI4)/VI1)' ((VI1-VI5)/VI1)']';
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA','CTSA','CFSA','PRSA'})
xlabel('type of algorithm');
ylabel('VI gain in percentage');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

```

## Simulation for Ps domain:

```
% This program performs dynamic switching at normal distribution of Ps parameters
% in Ps domain instead of time domain
% the program calculates the hit and false alarm probabilities and the
% value of the operational cost according to betas and dtags
% This version will show the objective function values for all collaboration
% levels and for all betas combinations.
```

```
clc
close all
clear all
Psvector=randn(1,200);%random normal
ctmp=max(abs(Psvector));
Psvector=Psvector./(ctmp*2);
Psvector=abs(Psvector+0.49);
[sorted_Psvector,INDEX]=sort(Psvector);

for rr=1:4
j=1;
i=1;
l=1;
p=1;
CCL=rr %current collaboration level
CCLw=rr;
CCLf=rr;
CCLnew=rr;
VI1(rr)=0;
VI2(rr)=0;
VI3(rr)=0;
VI4(rr)=0;
VI5(rr)=0;
bound1=1;
Xaxis=linspace(0,1,200);
N=1000; % # of objects
Nstr=num2str(N);
tmin=0.0000005; %frequency of the switch
tlastswitch=now; %time of the last switch
tlastswitchw=now; %time of the last switch
tlastswitchf=now;
tlastswitchnew=now;
ttemp=now;
Vp=-5000000000; %penalty for frequency
minv=1000; %minimal value of VIswitctch for switching
maxv=1500; %maximal value of VIswitctch for switching
VFA2H=[5]; % [0.05:0.05:1,1:0.1:10,10:1:100]; % VFA/VH aspect ratio range
VAR=1;
VARstr=num2str(VFA2H(VAR)*10);
if VFA2H(VAR)==0.333
    VARstr=num2str(3);
end
%format short

for Pscount=1:1:200;

    Ps=Psvector(Pscount);

    Psstr=num2str(Ps*100);
    dtag=0.7;
    %for dh=1:1:40
    dtag=dtag-1; % [-0.1:-0.1:-4]; % the range of d' for human (second detector)
    Dh=num2str(-dtag*10);
    dtagR=0.3;
    %dr=1;
    dtagR=dtagR-1; % [-0.1:-0.1:-4]% -2
    Dr=num2str(-dtagR*10);
    %lnbetar=1 % [-3:0.1:3]

    VH=10;
    VM=5;
    VCR=3;
    VHstr=num2str(VH);
```

```

VFA=-VH.*VFA2H(VAR);
Vc=-2;
VCstr=num2str(-Vc);
Vt=-2000/3600;
Vtstr=num2str(-Vt*3600);

trespond=1;
tr=0.01;

c3=1;
Inbetar=0;
c2=1;
Inbetah=0;

c1=1;
Inbetarh=2;

% the probabilities of the robot
Zsr(c1,c2,c3)=(-2.*Inbetar+dtagR.^2)/(2.*dtagR);
Znr(c1,c2,c3)=(-2.*Inbetar-dtagR.^2)/(2.*dtagR);
phr(c1,c2,c3)=1-normcdf(Zsr(c1,c2,c3));
pfar(c1,c2,c3)=1-normcdf(Znr(c1,c2,c3));
ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));

if Inbetar==0 & Inbetah==0 & Inbetarh==0
    Zsrtest=(-2.*Inbetar+dtagR.^2)/(2.*dtagR);
    Znrtest=(-2.*Inbetar-dtagR.^2)/(2.*dtagR);
    phrtest=1-normcdf(Zsr(c1,c2,c3));
    pfartest=1-normcdf(Znr(c1,c2,c3));
    ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
    ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));
end

% the optimal parameters of the robot if he was a single detector
betastar(c1,c2,c3)=(1-Ps)/Ps.*VFA2H(VAR); % calculating Beta*
ZsRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))+dtagR.^2)/(2.*dtagR);
ZnRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))-dtagR.^2)/(2.*dtagR);
phrstar(c1,c2,c3)=1-normcdf(ZsRstar(c1,c2,c3));
pfarstar(c1,c2,c3)=1-normcdf(ZnRstar(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the
% robot didn't detect
ZsH(c1,c2,c3)=(-2.*Inbetah+dtag.^2)/(2.*dtag);
ZnH(c1,c2,c3)=(-2.*Inbetah-dtag.^2)/(2.*dtag);
phh(c1,c2,c3)=1-normcdf(ZsH(c1,c2,c3));
pfah(c1,c2,c3)=1-normcdf(ZnH(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the robot
% already detected
ZsRH(c1,c2,c3)=(-2.*Inbetarh+dtag.^2)/(2.*dtag);
ZnRH(c1,c2,c3)=(-2.*Inbetarh-dtag.^2)/(2.*dtag);
phrh(c1,c2,c3)=1-normcdf(ZsRH(c1,c2,c3));
pfarh(c1,c2,c3)=1-normcdf(ZnRH(c1,c2,c3));

% the time parameters
tHh(c1,c2,c3)=5;
tFAh(c1,c2,c3)=5;
tHrh(c1,c2,c3)=5;
tFArh(c1,c2,c3)=5;

tMh(c1,c2,c3)=5;
tCRh(c1,c2,c3)=5;
tMrh(c1,c2,c3)=5;
tCRrh(c1,c2,c3)=5;
tmotor=2;

PHs(c1,c2,c3)=phr(c1,c2,c3).*phrh(c1,c2,c3)+(1-phr(c1,c2,c3)).*phh(c1,c2,c3);
VHs(c1,c2,c3)=N.*Ps.*PHs(c1,c2,c3).*VH;
PMs(c1,c2,c3)=phr(c1,c2,c3).*(1-phrh(c1,c2,c3))+(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3));
VMs(c1,c2,c3)=N.*Ps.*PMs(c1,c2,c3).*VM;
FFAs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3);

```

```

VFAs(c1,c2,c3)=FFAs(c1,c2,c3)*VFA;
FCRs(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3))+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3));
VCRs(c1,c2,c3)=FCRs(c1,c2,c3)*VCR;

ts(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3).*tHh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*pfarh(c1,c2,c3).*tFArh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*tFAh(c1,c2,c3)...
+N.*Ps.*phr(c1,c2,c3).*(1-phrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

tsHORr(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*(tHrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-
phr(c1,c2,c3)).*phh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfarh(c1,c2,c3).*(tFArh(c1,c2,c3)+tmotor)+N.*(1-
Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*phr(c1,c2,c3).*(1-phrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*(1-phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3).*(1-pfarh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

tsHOR(c1,c2,c3)=N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-
phr(c1,c2,c3)).*phh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfarh(c1,c2,c3).*(tFArh(c1,c2,c3)+N.*(1-Ps).*(1-
pfar(c1,c2,c3)).*pfah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*phr(c1,c2,c3).*(1-phrh(c1,c2,c3)).*(tMrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-phr(c1,c2,c3)).*(1-
phh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*(1-pfarh(c1,c2,c3)).*(tCRrh(c1,c2,c3)+tmotor)+N.*(1-Ps).*(1-
pfar(c1,c2,c3)).*(1-pfah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

Ndetect(c1,c2,c3)=(N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3)).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3));
VTs(c1,c2,c3)=ts(c1,c2,c3).*Vt+(N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3)).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3)).*Vc;
VTsHORr(c1,c2,c3)=tsHORr(c1,c2,c3).*Vt+(N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3)+N.*Ps.*(1-phr(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3)).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3)).*Vc;
VTsHOR(c1,c2,c3)=tsHOR(c1,c2,c3).*Vt+(N.*Ps.*phr(c1,c2,c3).*phrh(c1,c2,c3)).*phh(c1,c2,c3)+N.*(1-
Ps).*pfar(c1,c2,c3)).*pfarh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*pfah(c1,c2,c3)).*Vc;
VIs(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTs(c1,c2,c3);
VIsHORr(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHORr(c1,c2,c3);
VIsHOR(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHOR(c1,c2,c3);

PHsR(c1,c2,c3)=phr(c1,c2,c3);
VHsR(c1,c2,c3)=N.*Ps.*PHsR(c1,c2,c3)*VH;
FFAsR(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3);
VFAsR(c1,c2,c3)=FFAsR(c1,c2,c3)*VFA;
tsR(c1,c2,c3)=tr;
VTsR(c1,c2,c3)=tsR(c1,c2,c3).*Vt+(N.*Ps.*phr(c1,c2,c3)+N.*(1-Ps).*pfar(c1,c2,c3)).*Vc;
VIsR(c1,c2,c3)=VHsR(c1,c2,c3)+VFAsR(c1,c2,c3)+VTsR(c1,c2,c3);

% the probabilities of the HO collaboration level were taken from the
% robot probabilities and the difference between the HO and the R
% collaboration levels is just on the times parameters.
PHsHO(c1,c2,c3)=phr(c1,c2,c3);
VHsHO(c1,c2,c3)=N.*Ps.*PHsHO(c1,c2,c3)*VH;
FFAsHO(c1,c2,c3)=N.*(1-Ps).*pfar(c1,c2,c3);
VFAsHO(c1,c2,c3)=FFAsHO(c1,c2,c3)*VFA;
tsHO(c1,c2,c3)=N.*Ps.*PHsHO(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+FFAsHO(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*(1-PHsHO(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-Ps).*(1-pfar(c1,c2,c3)).*tCRh(c1,c2,c3);
VTsHO(c1,c2,c3)=tsHO(c1,c2,c3).*Vt+(N.*Ps.*phr(c1,c2,c3)+N.*(1-Ps).*pfar(c1,c2,c3)).*Vc;
VIsHO(c1,c2,c3)=VHsHO(c1,c2,c3)+VFAsHO(c1,c2,c3)+VTsHO(c1,c2,c3);

%-----
% switching calculations
%-----

CL(Pscount,:)= [VIsHORr(c1,c2,c3) VIsHOR(c1,c2,c3) VIsR(c1,c2,c3) VIsHO(c1,c2,c3)]; % values' matrix of different collaboration
levels
[C,CCLtemp]=max(CL(Pscount,:)); % C=value, CCLtemp=location
SV(1,Pcount)= [C]; % switch vector of best collaboration values for each row in CL
VIso=SV(1,Pcount);
VIscl=CL(Pscount,rr);
VIscl=CL(Pscount,CCL);
VIscl=CL(Pscount,CCLw);
VIscl=CL(Pscount,CCLf);
VIscl=CL(Pscount,CCLnew);

a1='HOR';
a2='HR';
a3='R';

```



```

a4='H';

switch CCLtemp
case 1
    label1=a1;
case 2
    label1=a2;
case 3
    label1=a3;
otherwise
    label1=a4;
end

switch CCL
case 1
    label2=a1;
case 2
    label2=a2;
case 3
    label2=a3;
otherwise
    label2=a4;
end

SVF(1,Pscount)=[CCLtemp];
%-----
%Switch objective fuction for plotting switch points
%-----

VIswh1(rr,Pscount)=(VIso-VIscl); % +trespond*Vt+Vp*(ttemp+tmin-now)*(Vp*(ttemp+tmin-now)<0); %S1
VIswh(rr,Pscount)=(VIso-VIsc); %"RLSA" for calculation
VIswh1(rr,Pscount)=(VIso-VIsc); %"RLSA" for plot
VIswhw(rr,Pscount)=(VIso-VIscw); %"CTSA" for calculation
VIswhw1(rr,Pscount)=(VIso-VIscw); %"CTSA" for plot
VIswhfw1(rr,Pscount)=(VIso-VIsf)+Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %"CFSA" for plot
VIswhfw(rr,Pscount)=(VIso-VIsf); %"CFSA" for calculation
VIswhfnw1(rr,Pscount)=(VIso-VIsnew)+trespond*Vt+Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
%"PRSA" for plot
VIswhfnw(rr,Pscount)=(VIso-VIsnew); %"PRSA" for calculation

%-----

%-----
%switching algorithm
%-----

%"RLSA" algorithm

if CCLtemp~=CCL
tcurrent=now;
    if tcurrent-tlastswitch>tmin %checking the frequency

        disp('switch to');
        disp(label1);
        disp('from');
        disp(label2);

        CCL=CCLtemp
        coY(rr,j)=C;
        coX(rr,j)=Pscount;
        coYs(rr,j)=VIswh1(rr,Pscount); %for the switching objective function plot
        tlastswitch=tcurrent;
        j=j+1;

        VIswh(rr,Pscount)=-trespond*Vt; %updating VIswh as optimal value after switching

    else
        disp('Collaboration level not optimal');
    end
end

%"CTSA" algorithm

if CCLtemp~=CCLw %optimization for switching
tcurrent=now;
    if tcurrent-tlastswitchw>tmin %checking the frequency

```

```

if VIs witchw(rr,Pscout)>maxv

    CCLw=CCLtemp;

        coYw(l)=C;
        coXw(rr,l)=Pscout;
        coYsw(rr,l)=VIs witchw1(rr,Pscout); %for the switching objective function plot
        tlastswitchw=tcurrent;
        l=l+1;

    VIs witchw(rr,Pscout)=-trespond*Vt; %updating VIs witch as optimal value after switching

end
end
end

%”CFSA”

if CCLtemp~=CCLf %optimization for switching with frequency
tcurrent=now;
if VIs witchfw1(rr,Pscout)>maxv

    CCLf=CCLtemp;

        %coYw(i)=C;
        coXf(rr,i)=Pscout;
        coYsfw(rr,i)=VIs witchfw1(rr,Pscout);
        VIs witchfw(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %updating VI with response
time for switching
        i=i+1;
        tlastswitchf=tcurrent;
    end
end

%”PRSA”

if CCLtemp~=CCLnew %optimization for switching with frequency
tcurrent=now;
if (tcurrent-tlastswitchnew>tmin)|(VIs witchfnew1(rr,Pscout)>maxv)
% if VIs witchfnew(Pscout)>maxv
    bound2=Pscout;
    BVF(bound1:bound2)=[CCLtemp];
    newVF(bound1:bound2)=SVF(bound1:bound2)==BVF(bound1:bound2);
    if sum(newVF(bound1:bound2))>((bound2-bound1)/2)
        CCLnew=CCLtemp;

        coXfnew(rr,p)=Pscout;
        coYsfnew(rr,p)=VIs witchfnew1(rr,Pscout);

        p=p+1;
        bound1=Pscout+1;

        VIs witchfnew(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
        tlastswitchnew=tcurrent;
    end
end
end

V11(rr)=VIs witchl(rr,Pscout)+V11(rr); %regular cost - S1
V12(rr)=VIs witch(rr,Pscout)+V12(rr); %cost with switch - “RLSA”
V13(rr)=VIs witchw(rr,Pscout)+V13(rr); %cost with switch and maxv filter - “CTSA”
V14(rr)=VIs witchfw(rr,Pscout)+V14(rr); %cost with switch and maxv filter after frequency penalty - “CFSA”
V15(rr)=VIs witchfnew(rr,Pscout)+V15(rr); %cost with switch and maxv filter after frequency penalty and new algorithm -
“PRSA”

end %Pscout
end %rr
%-----

```

```

%Figures
%-----

Figure(1)
subplot(1,2,1)
plot(sorted_Psvector,CL(:,1),sorted_Psvector,CL(:,2),sorted_Psvector,CL(:,3),sorted_Psvector,CL(:,4))
xlabel('time');
ylabel('VI');
title('Dinamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(sorted_Psvector(10),CL(10,1),'\leftarrow HOR','HorizontalAlignment','left','FontSize',10);
text(sorted_Psvector(15),CL(15,2),'\leftarrow HR','FontSize',10);
text(sorted_Psvector(5),CL(5,3),'\leftarrow R','FontSize',10);
text(sorted_Psvector(20),CL(20,4),'\leftarrow H','FontSize',10);

subplot(1,2,2)
plot(sorted_Psvector,CL(:,1),sorted_Psvector,CL(:,2),sorted_Psvector,CL(:,3),sorted_Psvector,CL(:,4),sorted_Psvector,SV,'black')
xlabel('time');
ylabel('VI');
title('Dinamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(sorted_Psvector(10),CL(10,1),'\leftarrow HOR','HorizontalAlignment','left','FontSize',10);
text(sorted_Psvector(15),CL(15,2),'\leftarrow HR','FontSize',10);
text(sorted_Psvector(5),CL(5,3),'\leftarrow R','FontSize',10);
text(sorted_Psvector(20),CL(20,4),'\leftarrow H','FontSize',10);
text(sorted_Psvector(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);

Figure(2)
subplot(1,2,1)
plot(sorted_Psvector,CL(:,1),sorted_Psvector,CL(:,2),sorted_Psvector,CL(:,3),sorted_Psvector,CL(:,4))
xlabel('Ps');
ylabel('VI');
title('Dinamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(sorted_Psvector(10),CL(10,1),'\leftarrow HOR','HorizontalAlignment','left','FontSize',10);
text(sorted_Psvector(15),CL(15,2),'\leftarrow HR','FontSize',10);
text(sorted_Psvector(5),CL(5,3),'\leftarrow R','FontSize',10);
text(sorted_Psvector(20),CL(20,4),'\leftarrow H','FontSize',10);
text(sorted_Psvector(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

subplot(1,2,2)
plot(sorted_Psvector,CL(:,1),sorted_Psvector,CL(:,2),sorted_Psvector,CL(:,3),sorted_Psvector,CL(:,4),sorted_Psvector,SV,'black')
xlabel('Ps');
ylabel('VI');
title('Dinamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(sorted_Psvector(10),CL(10,1),'\leftarrow HOR','HorizontalAlignment','left','FontSize',10);
text(sorted_Psvector(15),CL(15,2),'\leftarrow HR','FontSize',10);
text(sorted_Psvector(5),CL(5,3),'\leftarrow R','FontSize',10);
text(sorted_Psvector(20),CL(20,4),'\leftarrow H','FontSize',10);
text(sorted_Psvector(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);
text(sorted_Psvector(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

Figure(3) %the score if we stay always at the same collaboration level
plot(sorted_Psvector,VIswitchl(1,INDEX),sorted_Psvector,VIswitchl(2,INDEX),sorted_Psvector,VIswitchl(3,INDEX),sorted_Psvector,VIswitchl(4,INDEX))
xlabel('Ps');
ylabel('VIswitch');
title('Switch objective function score witout switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

plotlvl=2;

Figure(4)
plot(sorted_Psvector,VIswitchl(plotlvl,INDEX))
xlabel('Ps');
ylabel('VIswitch');
title('RLSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Psvector(coX(plotlvl,find(coX(plotlvl,:))))),coYs(plotlvl,find(coYs(plotlvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

Figure(5)

```

```

plot(sorted_Psvector, VIs witchw1(plotlv1,INDEX),Xaxis,maxv,'--.k')
xlabel('Ps');
ylabel('VIs witch');
title("CTSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Psvector(coXw(plotlv1,find(coXw(plotlv1,:))))),coYsw(plotlv1,find(coYsw(plotlv1,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

Figure(6)

```

plot(sorted_Psvector, VIs witchfw1(plotlv1,INDEX),Xaxis,maxv,'--.k')
xlabel('Ps');
ylabel('VIs witch');
title("CFSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Psvector(coXf(plotlv1,find(coXf(plotlv1,:))))),coYsfw(plotlv1,find(coYsfw(plotlv1,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

Figure(7)

```

plot(sorted_Psvector, VIs witchfnew1(plotlv1,INDEX),Xaxis,maxv,'--.k')
xlabel('Ps');
ylabel('VIs witch');
title("PRSA");
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Psvector(coXfnew(plotlv1,find(coXfnew(plotlv1,:))))),coYsfnew(plotlv1,find(coYsfnew(plotlv1,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

```

Figure(8)

```

Xaxis=[1 2 3 4 5];
Yaxis=[(V11-V12)' (V11-V13)' (V11-V14)' (V11-V15)'];
bar(Yaxis)
set(gca,'XTickLabel',{'regular switch','switch with max value filter', 'switch with max v. filter after freq p.', 'switch with past consideration'})
xlabel('type of algorithm');
ylabel('VI gain');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

```

Figure(9)

```

Xaxis=[1 2 3 4 5];
Yaxis=[((V11-V12)/V11)' ((V11-V13)/V11)' ((V11-V14)/V11)' ((V11-V15)/V11)'];
bar(Yaxis)
set(gca,'XTickLabel',{'regular switch','switch with max value filter', 'switch with max v. filter after freq p.', 'switch with past consideration'})
xlabel('type of algorithm');
ylabel('VI gain in percentage');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

```

## Simulation for random normal distribution of Ps and d'h:

```
% This program performs dynamic switching at normal distribution of Ps and dh parameters
% the program calculates the hit and false alarm probabilities and the
% value of the operational cost according to betas and dtags
% This version will show the objective function values for all collaboration
% levels and for all betas combinations.
```

```
clc
close all
clear all
%Psvector=linspace(0.1,0.9,200);
%Psvector=rand(1,200);%random uniform
Psvector=randn(1,200);%random normal
ctmp=max(abs(Psvector));
Psvector=Psvector./(ctmp*2);
Psvector=abs(Psvector+0.49);
dhvector=randn(1,200);%random normal
ctmp=max(abs(dhvector));
dhvector=dhvector./(ctmp);
dhvector=dhvector+0.5;
for rr=1:4
j=1;
i=1;
l=1;
p=1;
CCL=rr %current collaboration level
CCLw=rr;
CCLf=rr;
CCLnew=rr;
VI1(rr)=0;
VI2(rr)=0;
VI3(rr)=0;
VI4(rr)=0;
VI5(rr)=0;
VI6(rr)=0;
VI7(rr)=0;
VI8(rr)=0;
VI9(rr)=0;
VI10(rr)=0;
bound1=1;
Xaxis=linspace(0,1,200);
N=1000; % # of objects
Nstr=num2str(N);
tmin=0.0000005; %frequency of the switch
tlastswitch=now; %time of the last switch
tlastswitchw=now; %time of the last switch
tlastswitchf=now;
tlastswitchnew=now;
ttemp=now;
Vp=-5000000000; %penalty for frequency
minv=1000; %minimal value of VIs witch for switching
maxv=1500; %maximal value of VIs witch for switching
VFA2H=[5]; % [0.05:0.05:1,1:0.1:10,10:1:100]; % VFA/VH aspect ratio range
VAR=1;
VARstr=num2str(VFA2H(VAR)*10);
if VFA2H(VAR)==0.333
    VARstr=num2str(3);
end
%format short

for Pscount=1:1:200;
Ps=Psvector(Pscount);
Psstr=num2str(Ps*100);
dtag=dhvector(Pscount);
%dtag=0.7;
% for dh=1:1:40
dtag=dtag-1; % [-0.1:-0.1:-4]; % the range of d' for human (second detector)
Dh=num2str(-dtag*10);
dtagR=0.3;
%dr=1;
dtagR=dtagR-1; % [-0.1:-0.1:-4]% -2
Dr=num2str(-dtagR*10);
%lnbetar=1 % [-3:0.1:3]
```

```

VH=10;
VM=5;
VCR=3;
VHstr=num2str(VH);
VFA=-VH.*VFA2H(VAR);
Vc=-2;
VCstr=num2str(-Vc);
Vt=-2000/3600;
Vtstr=num2str(-Vt*3600);

trespond=1;
tr=0.01;

c3=1;
Inbetar=0;
c2=1;
Inbetah=0;

c1=1;
Inbetarh=2;

% the probabilities of the robot
Zsr(c1,c2,c3)=(-2.*Inbetar+dtagR.^2)/(2.*dtagR);
Znr(c1,c2,c3)=(-2.*Inbetar-dtagR.^2)/(2.*dtagR);
phr(c1,c2,c3)=1-normcdf(Zsr(c1,c2,c3));
pfar(c1,c2,c3)=1-normcdf(Znr(c1,c2,c3));
ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));

if Inbetar==0 & Inbetah==0 & Inbetarh==0
    Zsrtest=(-2.*Inbetar+dtagR.^2)/(2.*dtagR);
    Znrtest=(-2.*Inbetar-dtagR.^2)/(2.*dtagR);
    phrtest=1-normcdf(Zsr(c1,c2,c3));
    pfartest=1-normcdf(Znr(c1,c2,c3));
    ratio1=pfar(c1,c2,c3)/phr(c1,c2,c3);
    ratio2=(1-pfar(c1,c2,c3))/(1-phr(c1,c2,c3));
end

% the optimal parameters of the robot if he was a single detector
betastar(c1,c2,c3)=(1-Ps)/Ps.*VFA2H(VAR); % calculating Beta*
ZsRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))+dtagR.^2)/(2.*dtagR);
ZnRstar(c1,c2,c3)=(-2.*log(betastar(c1,c2,c3))-dtagR.^2)/(2.*dtagR);
phrstar(c1,c2,c3)=1-normcdf(ZsRstar(c1,c2,c3));
pfarstar(c1,c2,c3)=1-normcdf(ZnRstar(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the
% robot didn't detect
ZsH(c1,c2,c3)=(-2.*Inbetah+dtag.^2)/(2.*dtag);
ZnH(c1,c2,c3)=(-2.*Inbetah-dtag.^2)/(2.*dtag);
phh(c1,c2,c3)=1-normcdf(ZsH(c1,c2,c3));
pfah(c1,c2,c3)=1-normcdf(ZnH(c1,c2,c3));

% the probabilities of the HO (second detector) for object that the robot
% already detected
ZsRH(c1,c2,c3)=(-2.*Inbetarh+dtag.^2)/(2.*dtag);
ZnRH(c1,c2,c3)=(-2.*Inbetarh-dtag.^2)/(2.*dtag);
phrh(c1,c2,c3)=1-normcdf(ZsRH(c1,c2,c3));
pfarh(c1,c2,c3)=1-normcdf(ZnRH(c1,c2,c3));

% the time parameters
tHh(c1,c2,c3)=5;
tFAh(c1,c2,c3)=5;
tHrh(c1,c2,c3)=5;
tFArh(c1,c2,c3)=5;

tMh(c1,c2,c3)=5;
tCRh(c1,c2,c3)=5;
tMrh(c1,c2,c3)=5;
tCRrh(c1,c2,c3)=5;
tmotor=2;

```

```

PHs(c1,c2,c3)=p hr(c1,c2,c3).*p hrh(c1,c2,c3)+(1-p hr(c1,c2,c3)).*p hh(c1,c2,c3);
VHs(c1,c2,c3)=N.*Ps.*PHs(c1,c2,c3).*VH;
PMs(c1,c2,c3)=p hr(c1,c2,c3).*(1-p hrh(c1,c2,c3))+p hr(c1,c2,c3)).*(1-p hh(c1,c2,c3));
VMs(c1,c2,c3)=N.*Ps.*PMs(c1,c2,c3).*VM;
FFAs(c1,c2,c3)=N.*(1-Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*p fah(c1,c2,c3);
VFAs(c1,c2,c3)=FFAs(c1,c2,c3).*VFA;
FCRs(c1,c2,c3)=N.*(1-Ps).*p far(c1,c2,c3).*(1-p farh(c1,c2,c3))+N.*(1-Ps).(1-p far(c1,c2,c3)).*(1-p fah(c1,c2,c3));
VCRs(c1,c2,c3)=FCRs(c1,c2,c3).*VCR;

ts(c1,c2,c3)=N.*Ps.*p hr(c1,c2,c3).*p hrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-p hr(c1,c2,c3)).*p hh(c1,c2,c3).*tHh(c1,c2,c3)+N.*(1-
Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3).*tFArh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*p fah(c1,c2,c3).*tFAh(c1,c2,c3)...
+N.*Ps.*p hr(c1,c2,c3).*(1-p hrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-p hr(c1,c2,c3)).*(1-p hh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*p far(c1,c2,c3).*(1-p farh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*(1-p fah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

tsHORr(c1,c2,c3)=N.*Ps.*p hr(c1,c2,c3).*p hrh(c1,c2,c3).*(tHrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-
p hr(c1,c2,c3)).*p hh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3).*(tFArh(c1,c2,c3)+tmotor)+N.*(1-
Ps).(1-p far(c1,c2,c3)).*p fah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*p hr(c1,c2,c3).*(1-p hrh(c1,c2,c3)).*tMrh(c1,c2,c3)+N.*Ps.*(1-p hr(c1,c2,c3)).*(1-p hh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-
Ps).*p far(c1,c2,c3).*(1-p farh(c1,c2,c3)).*tCRrh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*(1-p fah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

tsHOR(c1,c2,c3)=N.*Ps.*p hr(c1,c2,c3).*p hrh(c1,c2,c3).*tHrh(c1,c2,c3)+N.*Ps.*(1-
p hr(c1,c2,c3)).*p hh(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+N.*(1-Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3).*tFArh(c1,c2,c3)+N.*(1-Ps).(1-
p far(c1,c2,c3)).*p fah(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*p hr(c1,c2,c3).*(1-p hrh(c1,c2,c3)).*(tMrh(c1,c2,c3)+tmotor)+N.*Ps.*(1-p hr(c1,c2,c3)).*(1-
p hh(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-Ps).*p far(c1,c2,c3).*(1-p farh(c1,c2,c3)).*(tCRrh(c1,c2,c3)+tmotor)+N.*(1-Ps).(1-
p far(c1,c2,c3)).*(1-p fah(c1,c2,c3)).*tCRh(c1,c2,c3)+tr;

Ndetect(c1,c2,c3)=(N.*Ps.*p hr(c1,c2,c3).*p hrh(c1,c2,c3)+N.*Ps.*(1-p hr(c1,c2,c3)).*p hh(c1,c2,c3)+N.*(1-
Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*p fah(c1,c2,c3));
VTs(c1,c2,c3)=ts(c1,c2,c3).*Vt+(N.*Ps.*p hr(c1,c2,c3).*p hrh(c1,c2,c3)+N.*Ps.*(1-p hr(c1,c2,c3)).*p hh(c1,c2,c3)+N.*(1-
Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*p fah(c1,c2,c3)).*Vc;
VTsHORr(c1,c2,c3)=tsHORr(c1,c2,c3).*Vt+(N.*Ps.*p hr(c1,c2,c3).*p hrh(c1,c2,c3)+N.*Ps.*(1-p hr(c1,c2,c3)).*p hh(c1,c2,c3)+N.*(1-
Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*p fah(c1,c2,c3)).*Vc;
VTsHOR(c1,c2,c3)=tsHOR(c1,c2,c3).*Vt+(N.*Ps.*p hr(c1,c2,c3).*p hrh(c1,c2,c3)+N.*Ps.*(1-p hr(c1,c2,c3)).*p hh(c1,c2,c3)+N.*(1-
Ps).*p far(c1,c2,c3).*p farh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*p fah(c1,c2,c3)).*Vc;
VIs(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTs(c1,c2,c3);
VIsHORr(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHORr(c1,c2,c3);
VIsHOR(c1,c2,c3)=VHs(c1,c2,c3)+VMs(c1,c2,c3)+VFAs(c1,c2,c3)+VCRs(c1,c2,c3)+VTsHOR(c1,c2,c3);

PHsR(c1,c2,c3)=p hr(c1,c2,c3);
VHsR(c1,c2,c3)=N.*Ps.*PHsR(c1,c2,c3).*VH;
FFAsR(c1,c2,c3)=N.*(1-Ps).*p far(c1,c2,c3);
VFAsR(c1,c2,c3)=FFAsR(c1,c2,c3).*VFA;
tsR(c1,c2,c3)=tr;
VTsR(c1,c2,c3)=tsR(c1,c2,c3).*Vt+(N.*Ps.*p hr(c1,c2,c3)+N.*(1-Ps).*p far(c1,c2,c3)).*Vc;
VIsR(c1,c2,c3)=VHsR(c1,c2,c3)+VFAsR(c1,c2,c3)+VTsR(c1,c2,c3);

% the probabilities of the HO collaboration level were taken from the
% robot probabilities and the difference between the HO and the R
% collaboration levels is just on the times parameters.
PHsHO(c1,c2,c3)=p hr(c1,c2,c3);
VHsHO(c1,c2,c3)=N.*Ps.*PHsHO(c1,c2,c3).*VH;
FFAsHO(c1,c2,c3)=N.*(1-Ps).*p far(c1,c2,c3);
VFAsHO(c1,c2,c3)=FFAsHO(c1,c2,c3).*VFA;
tsHO(c1,c2,c3)=N.*Ps.*PHsHO(c1,c2,c3).*(tHh(c1,c2,c3)+tmotor)+FFAsHO(c1,c2,c3).*(tFAh(c1,c2,c3)+tmotor)...
+N.*Ps.*(1-PHsHO(c1,c2,c3)).*tMh(c1,c2,c3)+N.*(1-Ps).(1-p far(c1,c2,c3)).*tCRh(c1,c2,c3);
VTsHO(c1,c2,c3)=tsHO(c1,c2,c3).*Vt+(N.*Ps.*p hr(c1,c2,c3)+N.*(1-Ps).*p far(c1,c2,c3)).*Vc;
VIsHO(c1,c2,c3)=VHsHO(c1,c2,c3)+VFAsHO(c1,c2,c3)+VTsHO(c1,c2,c3);

%-----
%switching calculations
%-----

CL(Pscount,:)= [VIsHORr(c1,c2,c3) VIsHOR(c1,c2,c3) VIsR(c1,c2,c3) VIsHO(c1,c2,c3)]; % values' matrix of different collaboration
levels
[C,CCLtemp]=max(CL(Pscount,:)); % C=value, CCLtemp=location
SV(1,Pscount)= [C]; % switch vector of best collaboration values for each row in CL
VIs0=SV(1,Pscount);
VIscl=CL(Pscount,rr);
VIs0=CL(Pscount,CCL);
VIs0=CL(Pscount,CCLw);
VIs0=CL(Pscount,CCLf);
VIs0=CL(Pscount,CCLnew);

```

```

a1='HOR';
a2='HR';
a3='R';
a4='H';

switch CCLtemp
case 1
    label1=a1;
case 2
    label1=a2;
case 3
    label1=a3;
otherwise
    label1=a4;
end

switch CCL
case 1
    label2=a1;
case 2
    label2=a2;
case 3
    label2=a3;
otherwise
    label2=a4;
end

SVF(1,Pscount)=[CCLtemp]; %switch vector of best collaboration levels
%-----
%Switch objective fuction for plotting switch points
%-----

VIsutchl(rr,Pscount)=(VIso-VIscl); %+trespond*Vt+Vp*(ttemp+tmin-now)*(Vp*(ttemp+tmin-now)<0); %S1
VIsutch=(rr,Pscount)=(VIso-VIs); %S2 for calculation
VIsutch1(rr,Pscount)=(VIso-VIs); %S2 for plot
VIsutchw(rr,Pscount)=(VIso-VIs); %S3 for calculation
VIsutchw1(rr,Pscount)=(VIso-VIs); %S3 for plot
VIsutchfw1(rr,Pscount)=(VIso-VIs)+Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %S4 for plot
VIsutchfw(rr,Pscount)=(VIso-VIs); %S4 for calculation
VIsutchfnew1(rr,Pscount)=(VIso-VIsnew)+trespond*Vt+Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0); %S5
for plot
    VIsutchfnew(rr,Pscount)=(VIso-VIsnew); %S5 for calculation

%-----

%-----
%switching algorithm
%-----

%S2 algorithm

if CCLtemp~=CCL
    tcurrent=now;
        if (tcurrent-tlastswitch)>tmin %checking the frequency

                disp('switch to');
                disp(label1);
                disp('from');
                disp(label2);

            CCL=CCLtemp;
            coY(rr,j)=C;
            coX(rr,j)=Pscount;
            coYs(rr,j)=VIsutch1(rr,Pscount); %for the switching objective function plot
            tlastswitch=tcurrent;
            j=j+1;

            VIsutch(rr,Pscount)=-trespond*Vt; %updating VIsutch as optimal value after switching

        else
            disp('Collaboration level not optimal');
        end
    end
end

```



```

%S3 algorithm

if CCLtemp~=CCLw %optimization for switching
tcurrent=now;
if (tcurrent-tlastswitchw)>tmin %checking the frequency
if VIs witchw(rr,Pscout)>maxv

    CCLw=CCLtemp;

    coYw(l)=C;
    coXw(rr,l)=Pscout;
    coYsw(rr,l)=VIs witchw1(rr,Pscout); %for the switching objective function plot
    tlastswitchw=tcurrent;
    l=l+1;

    VIs witchw(rr,Pscout)=-trespond*Vt; %updating VIs witch as optimal value after switching

end
end
end

%S4

if CCLtemp~=CCLf %optimization for switching with frequency
tcurrent=now;
if VIs witchfw1(rr,Pscout)>maxv

    CCLf=CCLtemp;

    %coYw(i)=C;
    coXf(rr,i)=Pscout;
    coYsfw(rr,i)=VIs witchfw1(rr,Pscout);
    VIs witchfw(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchf+tmin-now)*(Vp*(tlastswitchf+tmin-now)<0); %updating VI with response
time for switching
    i=i+1;
    tlastswitchf=tcurrent;
end
end

%S5

if CCLtemp~=CCLnew %optimization for switching with frequency
tcurrent=now;
if ((tcurrent-tlastswitchnew)>tmin)(VIs witchfnew1(rr,Pscout)>maxv)
% if VIs witchfnew(Pscout)>maxv
    bound2=Pscout;
    BVF(bound1:bound2)=[CCLtemp];
    newVF(bound1:bound2)=SVF(bound1:bound2)==BVF(bound1:bound2);
    if sum(newVF(bound1:bound2))>((bound2-bound1)/2)
        CCLnew=CCLtemp;

        coXfnew(rr,p)=Pscout;
        coYsfnew(rr,p)=VIs witchfnew1(rr,Pscout);

        p=p+1;
        bound1=Pscout+1;

        VIs witchfnew(rr,Pscout)=-trespond*Vt-Vp*(tlastswitchnew+tmin-now)*(Vp*(tlastswitchnew+tmin-now)<0);
        tlastswitchnew=tcurrent;
    end
end
end

VI1(rr)=VIs witchl(rr,Pscout)+VI1(rr); %regular cost - S1
VI2(rr)=VIs witch(rr,Pscout)+VI2(rr); %cost with switch - S2
VI3(rr)=VIs witchw(rr,Pscout)+VI3(rr); %cost with switch and maxv filter - S3
VI4(rr)=VIs witchfw(rr,Pscout)+VI4(rr); %cost with switch and maxv filter after frequency penalty - S4
    VI5(rr)=VIs witchfnew(rr,Pscout)+VI5(rr); %cost with switch and maxv filter after frequency penalty and new algorithm - S5

VI6(rr)=VIs cl+VI6(rr);
VI7(rr)=VIs c+VI7(rr);

```

```

VI8(rr)=VIscw+VI8(rr);
VI9(rr)=VIscf+VI9(rr);
VI10(rr)=VIscnew+VI10(rr);

end %Pscount
end %rr
%-----
%figures
%-----

figure(1)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.')
xlabel('time (sec)');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.',Xaxis,SV,'black')
xlabel('time (sec)');
ylabel('VI');
title('Dynamic Switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);

figure(2)
subplot(1,2,1)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.')
xlabel('time (sec)');
ylabel('VI');
title('Dynamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

subplot(1,2,2)
plot(Xaxis,CL(:,1),'-',Xaxis,CL(:,2),'--',Xaxis,CL(:,3),'.',Xaxis,CL(:,4),'-.',Xaxis,SV,'black')
xlabel('time (sec)');
ylabel('VI');
title('Dynamic Switching with switch points');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',3);
legend('HOR','HR','R','H')
text(Xaxis(30),SV(30),'\leftarrow Best CL','HorizontalAlignment','left','FontSize',20);
text(Xaxis(coX(2,find(coX(2,:))))),coY(2,find(coY(2,:))),'\leftarrow switch point','FontSize',15,'color','r');

figure(3)%the score if we stay always at the same collaboration level
plot(Xaxis,VIswitchl(1,:),'-',Xaxis,VIswitchl(2,:),'--',Xaxis,VIswitchl(3,:),'.',Xaxis,VIswitchl(4,:),'-.')
xlabel('time (sec)');
ylabel('VIswitch');
title('VIswitch values without switching');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
legend('HOR','HR','R','H')

plotlvl=2;

figure(4)
subplot(3,1,1)
plot(Xaxis,VIswitch1(plotlvl,:))
xlabel('time (sec)');
ylabel('VIswitch');
title('RLSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coX(plotlvl,find(coX(plotlvl,:))))),coYs(plotlvl,find(coYs(plotlvl,:))),'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotlvl,:)-VIswitch(plotlvl,:))
grid on

```

```

xlabel('time (sec)');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitch(plotvl,:)))
grid on
xlabel('time (sec)');
ylabel('Cumulative gain');

figure(5)
subplot(3,1,1)
plot(Xaxis,VIswitchw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time (sec)');
ylabel('VIswitch');
title('CTSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXw(plotvl,find(coXw(plotvl,:))))),coYsw(plotvl,find(coYsw(plotvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchw(plotvl,:))
grid on
xlabel('time (sec)');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchw(plotvl,:)))
grid on
xlabel('time (sec)');
ylabel('Cumulative gain');

figure(6)
subplot(3,1,1)
plot(Xaxis,VIswitchfw1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time (sec)');
ylabel('VIswitch');
title('CFSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXf(plotvl,find(coXf(plotvl,:))))),coYsfw(plotvl,find(coYsfw(plotvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfw(plotvl,:))
grid on
xlabel('time (sec)');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfw(plotvl,:)))
grid on
xlabel('time (sec)');
ylabel('Cumulative gain');

figure(7)
subplot(3,1,1)
plot(Xaxis,VIswitchfnew1(plotvl,:),Xaxis,maxv,'--.k')
xlabel('time (sec)');
ylabel('VIswitch');
title('PRSA');
set(findobj(gca,'Type','line','Color','black'),'LineWidth',2);
text(Xaxis(coXfnew(plotvl,find(coXfnew(plotvl,:))))),coYsfnew(plotvl,find(coYsfnew(plotvl,:))), 'x','FontSize',20,'color','r','HorizontalAlignment','center');
grid on

subplot(3,1,2)
plot(Xaxis,VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:))
grid on
xlabel('time (sec)');
ylabel('Gain');

subplot(3,1,3)
plot(Xaxis,cumsum(VIswitchl(plotvl,:)-VIswitchfnew(plotvl,:)))
grid on

```

```

xlabel('time (sec)');
ylabel('Cumulative gain');

figure(8)
% subplot(2,1,1)
% Xaxis=[1 2 3 4 5];
% Yaxis=[(VI1-VI2)' (VI1-VI3)' (VI1-VI4)' (VI1-VI5)'];
% bar(Yaxis)
% set(gca,'XTickLabel',{'S2','S3','S4','S5'})
% xlabel('type of algorithm');
% ylabel('VI gain');
% title('Comparison of VI gains');
% legend('HOR','HR','R','H')

% subplot(2,1,2)
Xaxis=[1 2 3 4 5];
Yaxis=[((VI1-VI2)./VI1)' ((VI1-VI3)./VI1)' ((VI1-VI4)./VI1)' ((VI1-VI5)./VI1)'];
bar(Yaxis)
set(gca,'XTickLabel',{'RLSA','CTSA','CFSA','PRSA'})
xlabel('Switching algorithms');
ylabel('% of VI gain');
title('Comparison of VI gains');
legend('HOR','HR','R','H')

```

## תקציר

העבודה מציגה פיתוח ויישום של אלגוריתמים לביצוע מיתוג דינאמי בין רמות שיתוף פעולה שונות של מערכת אדם- רובוט במשימת זיהוי מטרות לשם שמירה על ביצועים אופטימאליים למרות שינויים בפרמטרים בזמן ביצוע המשימה . מחקר זה מבוסס על פונקציית מטרה של המערכת אשר פותחה לשם קביעת הערכים הצפויים של ביצועי המערכת למשימה מוגדרת, בהנתן פרמטרים של המערכת, המשימה והסביבה תוך שימוש בתורת הגילוי (Bechar et al., 2006). רמות שיתוף הפעולה המוצגות במחקר זה, מתבססות על עשר רמות אוטומציה כפי שהוגדרו בסקלה של שרידן<sup>4</sup> (Sheridan, 1992).

ארבעה אלגוריתמים שונים למיתוג דינאמי פותחו ונבדקו באמצעות אנליזה נומרית עבור תרחישים שונים והתפלגויות שונות של הפרמטרים . האלגוריתמים בסימולציה תוכננו למתג את המערכת לרמת שיתוף פעולה אופטימאלית על ידי חישוב הערך האופטימלי של פונקציית המטרה עם פרמטרים נתונים של האדם , הרובוט, הסביבה והמשימה . התכנון של האלגוריתמים הביא בחשבון את תדר המיתוג המותר, מספר רמות שיתוף פעולה שניתן למתג בפעם אחת, זמני תגובה של המערכת ואת שינויי הפרמטרים עקב זמני תגובה אלה . אלגוריתם "RLSA" מבצע מיתוג רגיל אשר מתחשב בזמני תגובה של המערכת ו בהגבלות על תדירות המיתוגים. אלגוריתם "CTSA" מבצע מיתוג כאשר הרווח מפונקציית המיתוג גדול יותר מערך סף מוגדר תוך התחשבות בזמני תגובה של המערכת ו בהגבלות על תדירות המיתוגים. אלגוריתם "CFSA" מבצע מיתוג כאשר הרווח מפונקציית המיתוג גדול יותר מערך סף מוגדר ללא הגבלת תדירות המיתוגים אך עם הגדרת קנסות לשבירת התדירות המותרת. אלגוריתם "PRSA" מבצע מיתוג באמצעות הוספת חיזוי לפרמטרים ממידע קודם לפעולה של אלגוריתם "CFSA". כל האלגוריתמים נבדקו לביצוע מיתוגים עבור התפלגויות להסתברות מטרה שונות.

הסימולציות נעשו עבור ארבע רמות שיתוף פעולה התחלתיות שונות . הביצועים הכוללים של המערכת חושבו כערך ממוצע של 200 סימולציות בלתי תלויות עבור כל התפלגויות של המטרות שנבדקו. המסקנות היו שלא כל המיתוגים האפשריים משפרים בהכרח את ביצועי המערכת אלא לעתים אף יכולים לפגוע בביצועים הכוללים . לכן, הכנסת מנגנון לא יעיל למיתוג דינאמי יכולה להוות פגיעה חמורה במערכות מסוג זה . לכן, על מנת לבצע מיתוג דינאמי במערכת אדם- רובוט להשגת שיפור הביצועים דרושים אלגוריתמים יעילים. פיתוח מפורט של אלגוריתמים כאלה מוצג בעבודה זו . התוצאות של הסימולציה הראו שמיתוג דינאמי השיג ביצועי מערכת טובים יותר . יתרה מכך, כל האלגוריתמים הראו שיפור בתוצאות תחת מקרים קיצוניים.

**מילות מפתח:** שיתוף פעולה בין אדם לרובוט, מיתוג דינאמי, רמות שיתוף פעולה, זיהוי מטרות.

<sup>4</sup> "action selection and automation of decision", Sheridan, 1992 [32].



אוניברסיטת בן-גוריון בנגב  
הפקולטה למדעי ההנדסה  
המחלקה להנדסת תעשייה וניהול

## מיתוג דינאמי בין רמות שיתוף פעולה במערכת אדם-רובוט למשימות של זיהוי מטרות

חיבור זה מהווה חלק מהדרישות לקבלת התואר "מגיסטר" בהנדסה

מאת: יצחק טקץ'

מנחים: פרופ' יעל אידן, ד"ר אביטל בכר

תאריך: .....

תאריך: .....

תאריך: .....

תאריך: .....

מחבר: יצחק טקץ'

מנחה: פרופ' יעל אידן

מנחה: ד"ר אביטל בכר

יו"ר ועדת מוסמכים: פרופ' עדנה שכטמן

אוקטובר 2008



אוניברסיטת בן-גוריון בנגב  
הפקולטה למדעי ההנדסה  
המחלקה להנדסת תעשייה וניהול

## **מיתוג דינאמי בין רמות שיתוף פעולה במערכת אדם-רובוט למשימות של זיהוי מטרות**

חיבור זה מהווה חלק מהדרישות לקבלת התואר "מגיסטר" בהנדסה

מאת: יצחק טקץ'