BRAIN INSPIRED INTELLIGENCE FOR ENGINEERING AND HEALTHCARE APPLICATIONS

BRAIN INSPIRED INTELLIGENCE FOR ENGINEERING AND HEALTHCARE APPLICATIONS

By

Mahdi Naghshvarianjahromi

B.Sc. Iran University of Science and Technology, 2008

M.S. Amirkabir University of Technology, 2014

A Thesis

Submitted to the School of Graduate

Studies in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

McMaster University

Hamilton, Ontario, Canada

Doctor of Philosophy (2020) McMaster University
(Electrical and computer Engineering) Hamilton, Ontario

TITLE: Brain inspired intelligence for engineering and healthcare applications

AUTHOR: Mahdi Naghshvarianjahromi,
BSc. Iran University of Science and Technology, Tehran, Iran
MSc. Amirkabir University of Technology, Tehran, Iran

SUPERVISORS: Prof. Shiva Kumar and Prof. M. Jamal Deen

NUMBER OF PAGES: xvii, 209.

To provide change: Initiate out of box with less fear,

mistakes/failure, accepting hardships, educate your mind,

learning from mistakes, do it better this time, another mistake …

Do these cycles to achieve the goal.

Mistakes are the god teacher and precious gifts. *by Robert Kiyosaki*

Abstract

The cyber processing layer of smart systems based on a cognitive dynamic system (CDS) can be a good solution for better decision making and situation understanding in non-Gaussian and nonlinear environments (NGNLEs). The NGNLE situation understanding means deciding between certain known situations in NGNLE to understand the current state condition. Here, we report on a cognitive decision-making (CDM) system inspired by the human brain decision-making using prediction outcome of actions using a virtual NGNLE. The simple low-complexity algorithmic design of the proposed CDM system can make it suitable for real-time applications. The proposed system can be extended as a general software-based platform for brain-inspired decision making in smart systems in the presence of nonlinearity and non-Gaussian characteristics. Therefore, it can easily upgrade conventional systems to a smart one for autonomic CDM applications. Towards these objectives, CDS is applied to long-haul fiber optic link and a smart e-Health system as two examples of NGNLE.

In first case study, brain-inspired intelligence using the CDS concept is proposed to control the quality-of-service (QoS) over a long-haul fiber-optic link that is nonlinear and with non-Gaussian channel noise. Digital techniques such as digital-back-propagation (DBP) assume that the fiber optic link parameters such as loss, dispersion, and nonlinear coefficients, are known at the receiver. However, the proposed CDSs does not need to know about the fiber optic link physical parameters, and it can improve the bit error rate (BER) or enhance the data rate based on information extracted from the fiber optic link. The information extraction (Bayesian statistical modeling) using intelligent perception processing on the received data, or using the previously extracted models in the model library, is carried out to estimate the transmitted data in the receiver. Then, the BER is sent to the executive through the main feedback channel and the executive produces actions on the physical system/signal or internal commands for adaptive modeling configurations to ensure that the BER is continuously under the pre-defined forward-error-correction (FEC) threshold. Therefore, the proposed CDS is an intelligent and adaptive system that can mitigate disturbance in the fiber optic link (especially in an optical network) using prediction in the perceptor and/or doing proper actions in the executive based on BER

and the internal reward. Also, the earlier versions of CDS can achieve pre-defined goal faster using prediction outcome of actions in the executive. CDS was implemented for nonlinear fiber optic systems based on orthogonal frequency division multiplexing (OFDM) to show how the proposed CDS can bring noticeable improvement in the system's performance.

In second case study, CDS is applied for the autonomic computing layer of smart e-Health system for automatic diagnostic test and automatic screening process. In recent years, there has been a growing interest in smart e-Health systems to improve people's quality of life by enhancing healthcare accessibility and reducing healthcare costs. Continuous monitoring of health through a smart e-Health system may enable automatic diagnosis of diseases like Arrhythmia at its early onset that otherwise may become fatal if not detected on time. Towards this objective, we start from understanding the health situation by diagnosing healthy and unhealthy persons for automatic diagnostic test. For this, a decision-making system is developed that is inspired by medical doctors (MDs) decision-making processes. Our system is based on a CDS for CDM and it can create a decision tree automatically. Then, a CDS-based framework is developed for the smart e-Health system to realize an automatic screening process in the presence of a defective dataset. A defective dataset may have poor labeling and/or lack enough training patterns. To mitigate the adverse effect of such a defective dataset, we developed a decision-making system that is inspired by the decision-making processes in humans in case of conflict-of-opinions (CoO). The proposed CDS algorithm can thereby be incorporated in the autonomic computing layer of a smart-e-Health-home platform to achieve a pre-defined degree of screening accuracy in the presence of a defective dataset. The proposed platforms for automatic diagnostic test and automatic screening process can be extended for more healthcare applications such as disease class diagnosis, prevention, treatment or monitoring healing. As a result, the proposed CDS algorithms can be an example of the initial steps for designing the autonomic computing layer of a smart e-Health home platform.

Acknowledgments

I would like to thank my supervisors, Professors M. Jamal Deen and Shiva Kumar for the support and guidance during my PhD study. Without their guidance special care, mentorship, support and feedback, doing this thesis would have been impossible. I have learned many skills and knowledge from them during these years, which will help me in my further work as a researcher. These skills of research include critical thinking, planning of experiments, problem-solving, communications, data analysis and writing scientific articles. Also, I would like to thank Professors M. Jamal Deen and Shiva Kumar for the training they provided to me in all aspects and presentation to people with different education levels and backgrounds. It was a great honor for me to work under the supervision of such outstanding and knowledgeable professors.

I would like to thank the members of my supervisory committee Prof. Mohammed Bakr, and Prof. Steve Hranilovic for their constructive advice. I would also like to thank our group members Dr. Si Pan, Sumit Majumdar, Dr. Arif Alam, Wei Jiang, Yamn Chalich and Abu Ilius Faisal who all engaged me in constructive discussions during this work. I would like to thank Prof. Simon Haykin, Prof. Xun Li and Dr. Narjes Naghshvarianjahromi (Gynecologist) for helping me to better understand basic concepts such CDS, NGNLE and medical doctor decision making and healthcare policies.

Finally, I would like to thank all members of my family who have always supported me in all the aspects of my life.

List of Abbreviations

AC	Arrhythmia class
ADC	Analog-to-digital converter
ADMS	Autonomic decision-making system,
AF	Assurance factor
AI	Artificial intelligence
ASE	Amplified spontaneous emission
BER	Bit error rate
BP	Blood pressure
BR	Breath rate
CDM	Cognitive decision making
CDS	Cognitive dynamic system
CINC 2017	Computing in cardiology challenge 2017
CMA	Constant modulus algorithm
CNS	Central nervous system
CoO	Conflict of opinion
Covid-19	Corona virus disease 2019
CP	Cyclic prefix
CPS	Cyber-physical system
CSMDDMS	Cyber semi-medical doctors decision making system
CT	Computed tomography
CPU	Central processing unit
CVD	Cardiovascular disease
DAC	Digital-to-analog-converter
DBP	Digital back propagation
DE	Diagnosis error
DF	Discretization factor
ECG	Electrocardiogram
EDFA	Erbium-doped fiber amplifier
FA	False alarm
FEC	Forward error correction
FFT	Fast Fourier transform
FWM	Four-wave mixing
HCI	Human-computer interaction
HD	Hard decision
HR	Heart Rate
HRI	Human-robot interaction
LGE	Linear and Gaussian environment
ICI	Inter-carrier interference
IFFT	Inverse fast Fourier transform
IoT	Internet of things
IQ	In-phase quadrature
ISI	Inter-symbol-interference
LMS	Least mean squares
LO	Local oscillator

LOOCV	Leave-one-out cross validation
M2M	Machine to machine
MAP	Maximum a posteriori
NF	Noise figure
NLSE	Nonlinear Schrödinger equation
MD	Medical doctors
ML	Machine learning
NGNLE	Non-Gaussian and nonlinear environment
NGNLHE	Non-Gaussian and nonlinear health environment
OFDM	Orthogonal frequency division multiplexing
OH	Overhead
OHSCA	Out of hospital sudden cardiac arrest
OTDM	Optical time division multiplexing
P/S	Parallel to serial
PAC	Perception action cycle
PDF	Probability density function
PF	Precision factor
PMAC	Perception multiple actions cycle
QAM	Quadrature amplitude modulation
QoS	Quality of service
RAM	Random-access memory
RF	Random forest
RL	Reinforcement learning
rw	Reward
Rx	Receiver
S/P	Serial to parallel
SL	Supervised learning
SNR	Signal-to-noise ratio
SPM	Self-phase modulation
SSMF	Standard single mode fiber
SVM	Support vector machine
Tx	Transmitter
UK	United Kingdom
USA	United States of America
UUT	User under test
V2V	Vehicle to vehicle
WDM	Wave division multiplexing
XPM	Cross-phase modulation

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1. Smart system based on cyber-physical system architecture

Globally, Internet-of-Things (IoT) is attracting much attention from researchers, technology developers and providers [1.1][1.2][1.3]. IoT technology is based on connecting a variety of conventional devices and systems such as sensors, actuators, appliances, TV and cars with computing devices to have the capability to automatically transfer data over a network. Therefore, IoT creates a network of intelligent systems that can communicate with each other or with human Users [1.1][1.2][1.3]. In recent years, the considerable advances in computing, wireless and network communications, low cost and low power sensors, actuators, and electronic components have made many previous fictional applications of IoT now practical, an example being the smart e-Health home presented in [1.4]. This application can potentially make our lives more comfortable and safer, as well as dramatically reduce the healthcare system cost for elderly healthcare. For simplicity, we focused on the cyber-physical system (CPS) sub-area of IoT for the architecture of a smart system. Figure 1.1 shows the smart system architecture as a CPS. In this thesis, we focused on two main layers of a smart system based on CPS [1.4]:

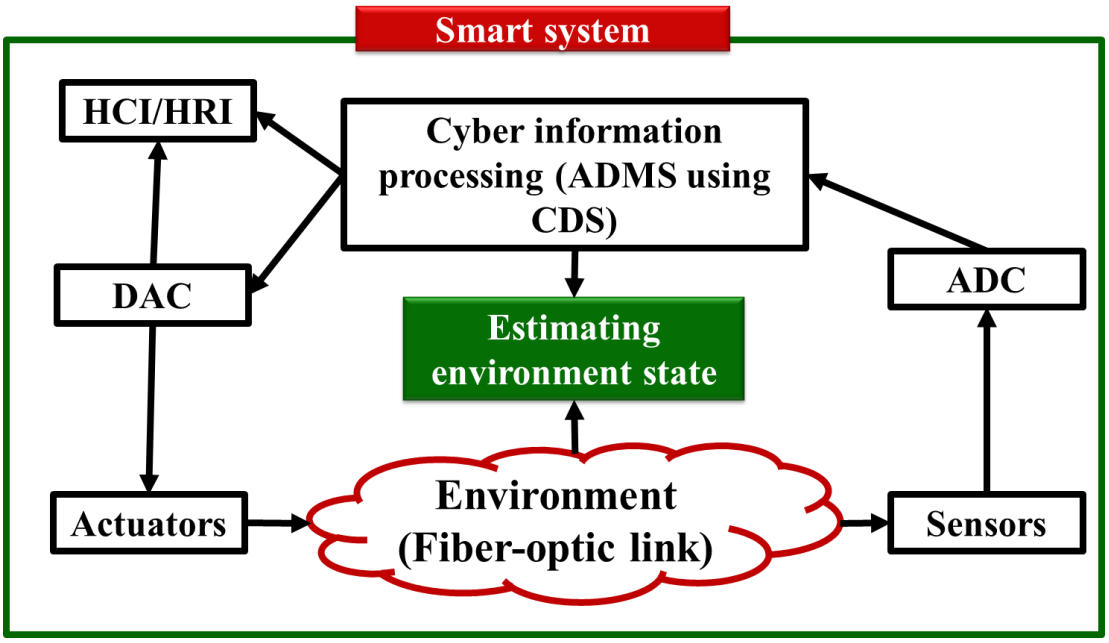


Figure 1.1: Smart system architecture with an autonomic decision-making system (ADMS) using cognitive dynamic system (CDS) as the cyber physical system (CPS). ADC: Analog-to-digital converter, DAC: Digital to analog converter, HCI: Human-computer interaction, HRI: Human-robot interaction.

- **Sensors and Actuators:** These are environmental sensors, environmental control units, or any device or methods providing information between Users (humans) and computers/robots in human-computer interaction (HCI) or human-robot interaction (HRI).
- **Autonomic computing:** This layer is responsible for knowledge management, environment situation understanding such as whether the User has a disease or not, intelligent reasoning and decision making. This cyber part of the smart system is termed as the autonomic decision making system (ADMS) for information processing of the measured signals from the sensors.

In this thesis, we focus on the *autonomic computing layer* of a smart system. As we mentioned before, the *autonomic computing layer* can be termed as the ADMS [1.5, 1.6].) ADMS is responsible for information processing that captures sensory data of the smart system environment as well as those related to estimation environment state. Typically, the ADMS can be implemented using artificial intelligence (AI) technique. However, in this thesis, we implemented the ADMS using the cognitive dynamic system (CDS) concept for two examples of non-Gaussian and nonlinear environments

(NGNLEs) i.e., fiber optic link and healthcare applications. Here, NGNLE means that the outputs of the environment are not linearly dependent on the inputs and do not have a Gaussian distribution.

1.2. Basic concepts

The functional block diagram of the typical CDS is built according to the principles of cognition, that is, perception-action cycle (PAC), memory, attention, intelligence, and language [1.7][1.8][1.9]. In the following, the role of each pillar of cognition is described in detail. The CDS can be considered as enhanced artificial intelligence (AI). A CDS creates internal rewards and uses it to take some actions (More details on enhanced-AI discussed in section 1.6) [1.8]. The CDS was proposed as an alternative to artificial intelligence (AI) in most AI applications [1.8][1. 9].

1.2.1. Perception-action cycle (PAC)

The PAC is a basic principle of cognition (Figure 1.2) [1.7]. Inspired by neuroscience and the human brain, the PAC is the cybernetic information-processing loop that helps the living organism to adapt dynamically to its environment (e.g., the environment can be the transmission medium in an acoustic channel or a fiber optic link or diagnosing diseases in a living organism) by aim-directed behavior or language [1.7]. In these activities, the CDS functions like the human brain and processes the measured information from sensors [1.8]. In Figure 1.3, the basic description of the CDS is shown. This figure gives us a better insight into how the PAC functions in the form of a global feedback loop. The PAC includes the perceptor and the executive. In addition, a feedback channel links them together, and the environment closes the PAC loop. The most important parts of PAC are the following.

1. The set of observables (which are the data obtained by measurement of the environment) are processed by the perceptor.
2. Based on the current and previous data and combined with intelligence, the perceptor predicts the states of the environment, which is passed to the executive through the feedback channel.

3. The executive produces actions on the environment to achieve a specific goal so that the set of observables in the following cycles may be different.
4. The results of each cycle of the PAC will be used for succeeding cycles.
5. When there is a specific goal, actions performed on the environment or physical system, the current PAC is guided by the derivation hypothesis from memory. As a result, the CDS will update the data in the current cycle and modify the hypothesis which will be used in the next cycle.

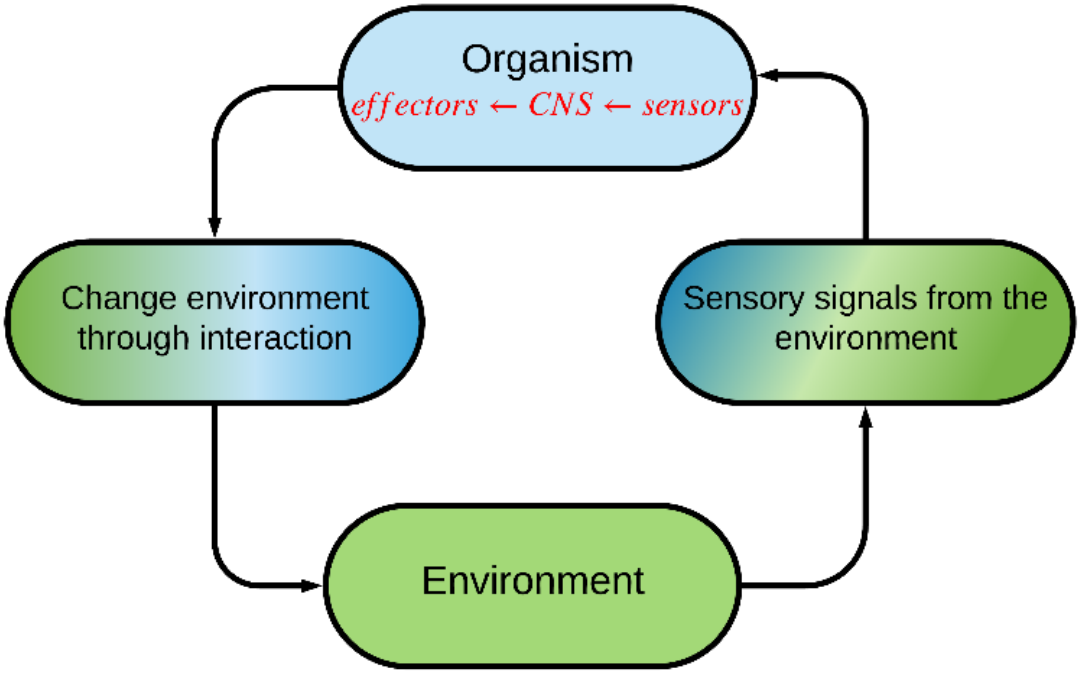


Figure 1.2: A simple diagram for perception-action cycle in the brain. (CNS: Central nervous system). [1.10]

The actions of the executive produce a change in the environment from one state to another. This procedure continues, cycle by cycle, with further actions until the desired goal that is determined by the systems policy, is achieved. The PAC may be viewed as coherent interaction and coordination between perception, prediction, action and outcome. For example, suppose someone wants to drink coffee (goal). If she sees a cup on the table (sense observables using the eyes), she perceives what it is and what she should do with it (perception). Even if this is the first time, she is seeing this specific

cup, she knows how it is different from a glass of cold water (prediction). This is because the extracted models for cups and glasses are stored in the brain and it can predict what is currently seen is a coffee cup or a glass of water based on previous experiences (extracted models of cups and glasses). The brain can also predict the hand movements before doing an action (prediction of outcome of actions), and she automatically picks up a coffee cup with her hand (action). Her eyes sense the cup and her brain measures the distance to the cup from her hand (perception that is the outcome of the action, "picking up the cup"). This process continues until the cup is close to her mouth (PAC).

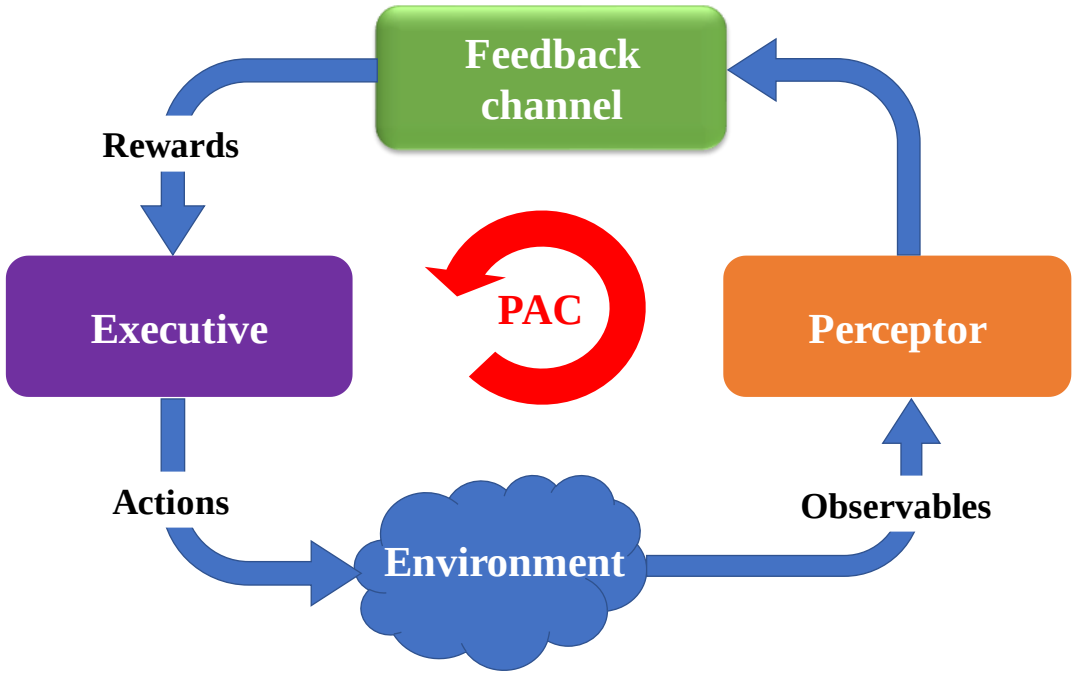


Figure 1.3: Block diagram of a basic cognitive dynamic system [1.11].

1.2.2. Memory

Figure 1.4 shows the functional block diagram of a brain-like memory in the CDS [1.9]. The memory is required in the perceptor (known as perceptual memory) as well as in the executive (known as executive memory). While the perceptual memory enables the perceptor to recognize the distinctive features of the observables and categorize the learned features accordingly in some statistical sense, the executive memory keeps track of the chosen actions in the past and their effectiveness (Figure 1.4) [1.8]. The

function of the memories is to learn from the environment, store the acquired knowledge, continually update the stored knowledge in the presence of environmental fluctuations, and predict the consequences of actions taken and/or selections made by the CDS [1.8].

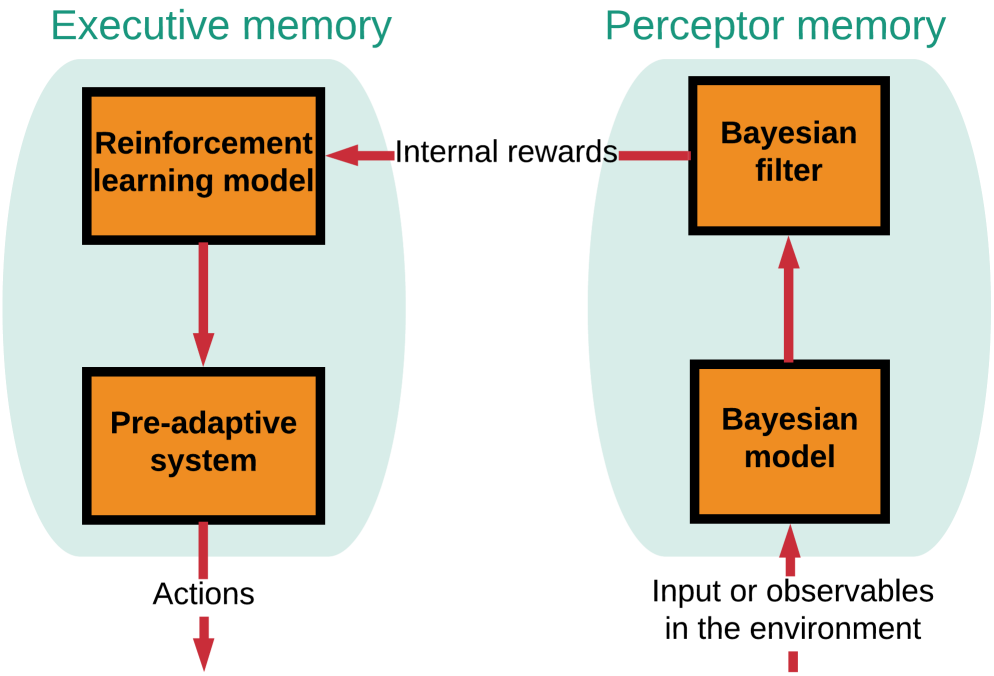


Figure 1.4: The functional brain-like block of the executive and perceptual memory in the cognitive dynamic system (CDS) [1.10].

1.2.3. Attention

Generally, perceptual attention and executive attention can be implemented in the perceptor and executive, respectively. While perceptual attention addresses the excessing information from the environment, executive attention can be used to preserve the CDS with minimum disturbance.

As shown in Figure 1.5, first, a statistical model (Bayesian model) of the environment is established, which is followed by the Bayesian filter for estimating the environment. In the conventional CDS [1.8], the Bayesian model/Bayesian generative model is responsible for feature extraction. Internal rewards (positive or negative) are sent from

the perceptor to the executive. The reinforcement-learning model in the executive exploits the internal reward attributed to imperfection in the perceptor. As a result, the best possible actions are chosen based on internal rewards [1.8].

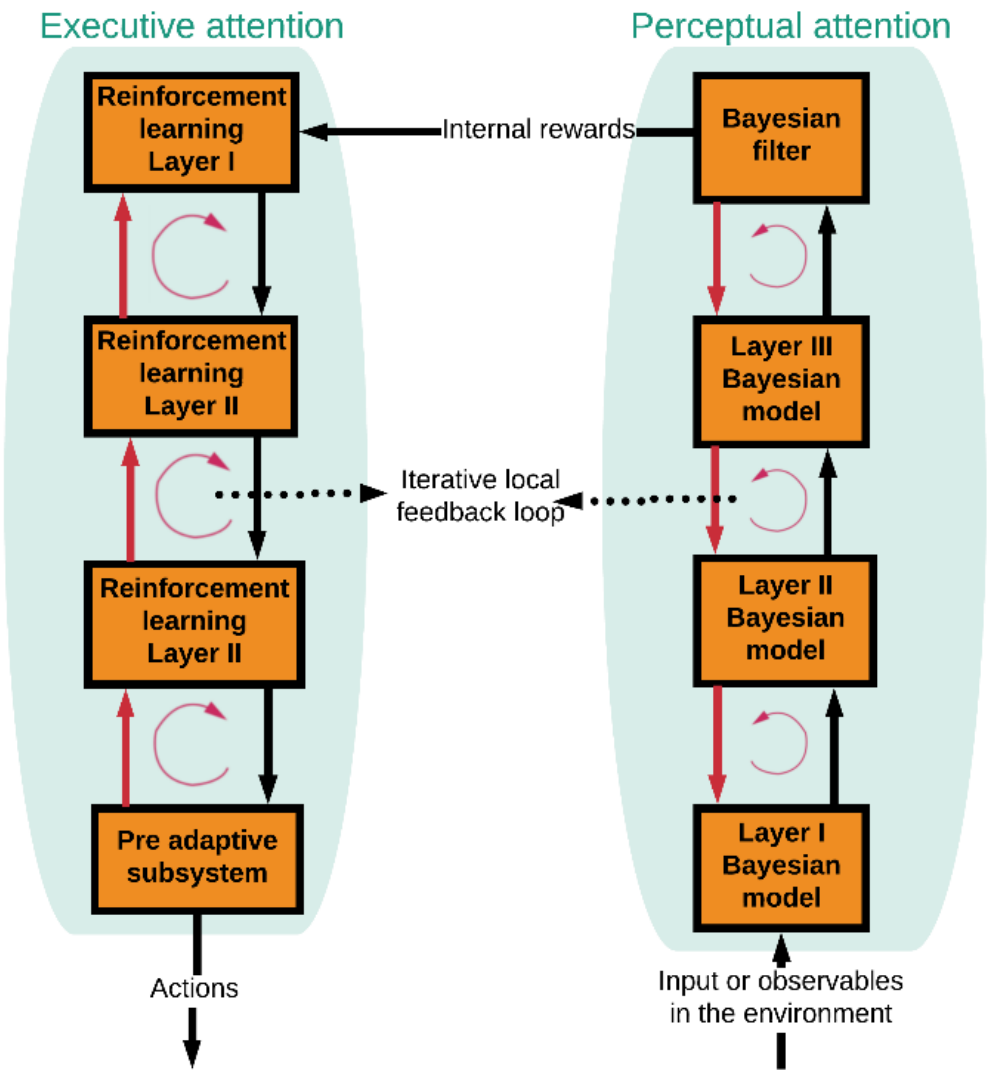


Figure 1.5: Attention (i.e., focusing) on the CDS [1.10].

1.2.4. Intelligence

There is no unified definition of intelligence. However, in this thesis, the intelligence of the CDS can be defined as reasoning and decision-making, perception, automatic control and problem solving. Using PAC concept and feedback channels, CDS can

make cognitive and intelligent decisions for situation understanding of unknown uncertainties in the environment.

1.2.5. Language

Language is a unique ability of the human brain. However, we can create it in a CDS, if we define language for a network of connected CDSs. Then, this network of CDSs can exchange information, models and experiences using machine-to-machine (M2M) communications protocols. However, implementing language for the CDS is out of the scope of this research.

1.3. Conventional CDS structure

Conventional CDS has three main sub-systems that are inspired by PAC: a perceptor, an executive, and a global feedback channel connecting the perceptor to the executive that can interact with a linear and Gaussian environment (LGE) such as a cognitive radio [1.8].

1.3.1. Perceptor

The conceptual schematic of a conventional perceptor is usually composed of the Bayesian generative model, Bayesian filter (Kalman filter), and entropic information-processor. In conventional CDS, the Bayesian generative model/Bayesian model is defined for feature extraction. However, in [1.12], it is stated that the Bayesian generative model is not required because observables are available. Also, according to [1.13], the Bayesian generative model is not a suitable choice and it is impractical due to dynamic nature of environment. Refs. [1.12] [1.13] did not use the Bayesian generative model in their algorithmic implementation. In this thesis, we propose a conventional perceptor without introducing the Bayesian generative model as a subsystem of perceptor, because, similar to [1.12], observables are available for applications in this thesis.

Bayesian filter (Kalman filter)

Typically, the Bayesian filter is the main sub-system of the conventional perceptor. The Bayesian filter should estimate the state of the observables [1.9] [1.12]. However, since a conventional CDS is applied only to the environments that have a linear model and a Gaussian distribution, the well-known Kalman filter is typically used in the perceptor [1.14].

Entropic Information-Processor

The entropy of the Kalman filter output can be calculated using Shannon's information theory [1.15][1.16]. However, in a generic CDS, the Shannon's equation for calculating the entropy is simplified to mean and covariance matrix for linear and Gaussian environments (LGE) [1.12].

1.3.2. Feedback channel

The feedback channel sends internal rewards to the executive. The internal reward is based on the calculated entropic state of the perceptor.

1.3.3. Executive of Generic CDS

The most important subsystems of the executive are reinforcement learning, planner, actions library and policy.

Reinforcement Learning

The objective of reinforcement learning is to optimize the received internal rewards from the perceptor using the cost-to-go function [1.17].

Planner and Action Library

Based on the current output of the cost-to-go function, the planner should extract a set of prospective actions from the action library to apply to the environment after filtering by the policy.

Policy

Based on [1.9], cognitive policy in conventional CDS can be defined as the goals of reinforcement learning and planning processes that should be improved by the CDS [1.9].

1.4. Literature survey on generic CDS

In [1.8, 1.18], the CDS was used for cognitive radar applications to provide optimal target detection and intelligent signal processing. In [1.8][1.18][1.19], the CDS was applied for cognitive radio applications, such as dynamic spectrum management in wireless communication networks [1.18]. The CDS was discussed for theory and applications of cognitive control in [1.8][1.9][1.18]. In [1.20], the general concept of CDS was introduced for risk control in physical systems. Specific applications of CDS for risk control were presented for vehicular radar transmit waveform selection [1.12], mitigating the cyber-attack in a smart grid [1.13], detecting the cyber-attack in smart grid [1.21], and mitigating vehicle-to-vehicle (V2V) jamming [1.22]. Also, CDS was applied as the brain of complex network [1.23]. A prospective block diagram of a CDS for smart homes was presented in [1.24]. In [1.25], the application of CDS for cybersecurity applications was discussed.

1.5. CDS as a research tool or a research objective?

We planned to use the conventional CDS as the research tool to apply on our desired nonlinear and non-Gaussian environments (NGNLEs) such as long-haul fiber optic link and healthcare applications. However, the conventional CDS was designed for only linear and Gaussian environments (LGEs) [1.8][1.9][1.12][1.13][1.18-1.25]. Here, LGE means that the outputs of the environment linearly depend on the inputs and they have a Gaussian distribution. Therefore, the perceptor of conventional CDS uses the Kalman filter, which can be applied only on LGEs and cannot be applied to NGNLEs. In addition, another issue regarding the Kalman filter is the computational complexity and concerns regarding implementation in hardware. Furthermore, the conventional

perceptor of the CDS cannot extract a model with finite memory, which is required for its implementation of CDS on a NGNLE. In addition, some equations in a conventional CDS can be simplified for LGEs such as entropic state and entropic processor can be simplified to mean and covariance matrices, respectively [1.12]. However, the complicated raw equations used for a NGNLE make the conventional CDS impractical for implementation in complex environments such as in healthcare applications or in a long-haul nonlinear fiber optic link. Thus, in this thesis, a CDS for NGNLE with simple, straightforward and fast PAC algorithms is developed.

In this thesis, using the PAC concept, the CDS is redesigned so that it can be applied to NGNLEs. It is desired that the redesigned CDS should be computationally efficient. Thus, we substitute subsystems such as the entropic state processor or Kalman filter with assurance factor and supervised learning for extracting the posteriori, respectively. These new subsystems can make the CDS applicable to NGNLE as well as lower the computational cost in comparison to the equivalent subsystem in the conventional CDS subsystems. Also, some subsystems such as the creation of a virtual NGNLE by the executive for the outcome of action prediction is newly added to the CDS. So, we created our research tool and evolve it version by version.

1.5.1. Examples of NGNLE application of AI

We should mention here that many AI applications [1.10][1.26-1.31] are for NGNLEs. For example, most data obtained or measured from health conditions, education and social sciences are often not normally distributed [1.26]. Some examples of non-Gaussian distribution for health conditions, education and social sciences are prostate cancer modeling [1.27][1.28], psychometrics [1.29] and labor income [1.30], respectively. The same holds true for a long-haul fiber-optic link as a NGNLE [1.10][1.31].

1.5.2. Proposed perceptor for NGNLE

Our proposed perceptor has three main subsystems: supervised learning (SL) for extracting the posterior, cognitive decision making based on the maximum of posteriori

(MAP) rule and internal reward calculation using assurance factor concept in the perceptor of the CDS.

Supervised learning (SL)

The proposed perceptor creates the automatic decision tree or a jungle of trees for extracting the posteriori of NGNLE. The decision tree approach is one of the common approaches in SL. The recent version of our proposed CDS can generate decision trees adaptively in the perceptor depending on the application.

Cognitive decision-making using MAP rule

The proposed CDS uses a MAP rule for cognitive decision-making using the extracted posteriori from the SL in versions 1 to 4. The detailed of the MAP rule details are discussed in sections 3.21, 4.2.1 and 5.2.1.

Internal reward calculation using the assurance factor concept

The internal reward is inspired by the fuzzy human decision-making approach with a lower computational cost, especially for making a decision in complex NGNLEs such as in healthcare applications. Fuzzy logic here means that the logic values of variables can be a real number between 0 and 1 [1.32][1.33] [1.37]. Fuzzy logic can be presented as the assurance about the decision. For example, we can make the wrong decision when the assurance is less than 1.

1.5.3. Feedback channels

In the proposed feedback channel for NGNLE, there are 3 types of feedback channels: Global feedback channel, Internal commands link (executive to perceptor), and internal feedback channel (perceptor to executive).

Global feedback channel

As in the conventional CDS, the global feedback channel sends the internal reward to the executive.

Internal feedback link (perceptor to executive)

The perceptor can send the extracted model or all the other required information from NGNLEs or current observables to the executive.

Internal commands link (Executive to perceptor)

The executive can send internal commands to perceptor for adaptive modeling configuration such as accuracy of discretization or level of decision trees.

1.5.4. Proposed executive for NGNLE

Similar to the conventional CDS, the executive is an essential part of the proposed CDS also. It is responsible for improving both the decision-making accuracy and the speed of achieving predefined goals such as diagnosis error better than 10%. This can be done by applying action on the NGNLE or sending internal commands to the perceptor. The executive provides non-monotonic reasoning (i.e. invalidate the previous decision offer achieving more evidence) to the CDS by using the internal reward and changing the decision trees level. The proposed executive for NGNLE has three parts: planner (it consists of the actions library also), policy, and learning using prediction of the outcome of the virtual action. Next, we describe the planner and policy followed by learning using prediction.

Planner and actions library

The planner extracts the set of prospective actions that are already saved in the CDS actions library. Here, actions library is a set of all possible actions that can be done by CDS in the action space. Also, the planner selects the first action in the 1^{st} PAC using pre-adaptive actions or randomly from the actions library. Actions type in action space can be environmental actions or internal commands. Environmental actions can be applied to NGNLE. In addition, in this thesis, the planner updates the actions type and sends an internal command to the perceptor for updating modeling configuration.

Policy

The policy determines the desired goals that the CDS should achieve using the PAC. The goal can be a trade-off between the desired CDM accuracy and the computational cost. For example, this goal can be a 10 percent acceptable decision-making error for diagnosing a patient's health state or providing a bit error rate (BER) less than 4.7×10^{-3} for the maximum possible data-rate.

Learning using prediction of virtual NGNLE outcome of actions

Similar to the conventional CDS, the objective of reinforcement learning is to optimize the incoming internal rewards computed in the perceptor and received through the global feedback channel by the cost-to-go function [1.17]. However, the proposed executive can take action on the virtual environment and optimize the internal rewards from the virtual environment, without the need apply the actions on real NGNLE. For example, the CDS without the outcome of action prediction may need 20 PACs to achieve the predefined goal. However, the CDS using a virtual environment for prediction outcomes of actions may reach a predefined goal in 9 PACs.

1.6. Why CDS?

In this section, we briefly discuss different machine learning methods. Then, we discuss why the CDS is selected among popular machine learning schemes. Typically, AI uses machine learning approaches to create intelligent machines. AI created by machine learning is different from the symbolic rule-based AI. Therefore, instead of programming predefined rules, AI using machine learning can learn from datasets, examples and experiences. However, machine learning-based AI requires access to large and reliable datasets, examples and experiences to provide reliable outcomes. This is an important weakness of machine learning based-AI approaches (including CDS) in comparison to rule-based AI. Also, this means that the training time is generally longer than that of the rule-based AI. In machine learning-based AI, a machine learning algorithm extracts the model from the dataset. Then, the model can be used for prediction. Also, the algorithm can learn to optimize models based on

datasets and policies, for example, in a special task such as providing pre-FEC BER under 0.01 with acceptable modeling complexity.

1.6.1. Machine-learning approaches

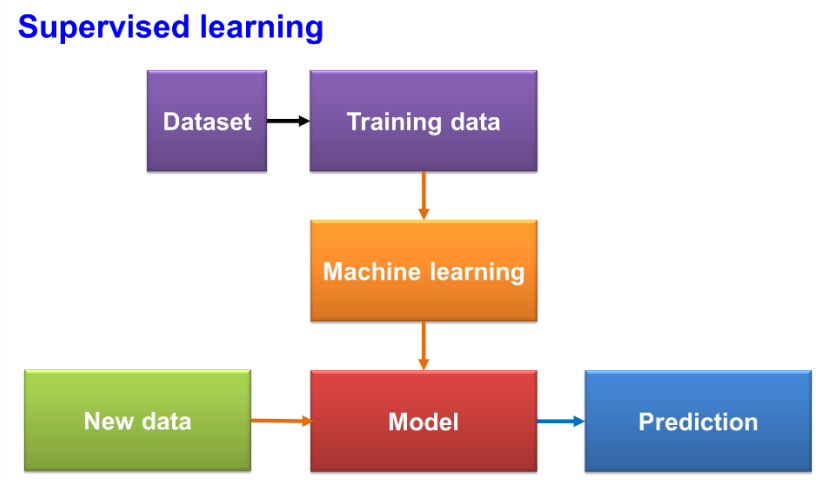
Machine learning is an interdisciplinarity field in computer science, mathematics and statistics. Generally, there are many machine-learning approaches such as supervised learning, reinforcement learning, semi-supervised learning, unsupervised learning and transfer learning. We will only focus on the first two main types-supervised learning and reinforcement learning.

Supervised learning (SL)

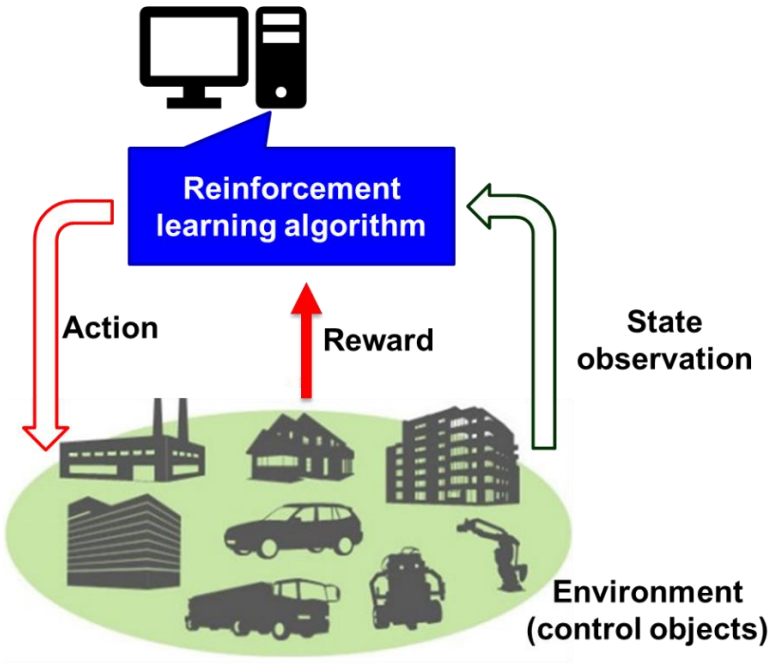
In machine learning, a very popular approach for practical applications such as predicting the length of stay in hospital, transmitted symbols in communication systems, radar target detection or health condition is supervised learning (SL). SL can find patterns inside data. In general, the SL algorithm can learn how to create a classifier for predicting the output variable y for a given input variable x (see Fig. 1.6a). The SL algorithm extracts a mapping function f which maps x to y, i.e., $y = f(x)$. An algorithm with a set of data $\{x_1, x_2, ..., x_n\}$ with the corresponding output label $\{y_1, y_2, ..., y_n\}$ builds the classifier.

Supervised learning can be divided into two main branches: (1) learning by type of prediction and (2) learning by type of modeling. The type of prediction problems can be divided into regression or classification. For predicting continuous output data, regression learning methods can be more suitable. For the prediction of output class, classification algorithms such as linear regression, naive Bayes, decision trees or support vector machines (SVM) are better choices [1.34]. For example, a child's height prediction is improved with linear regression. However, the decision tree or naive Bayes are better for binary diagnostic test predictions. This type of modeling can be the extraction of a discriminative model such as decision trees and SVM algorithms. These algorithms can extract the decision boundary within the data based on the learning goal. Also, machine learning methods such as naive Bayes or Bayesian approaches, can learn the probability distributions of the data.

In summary, SL trains an algorithm on a labeled database to predict the correct outputs for the unseen inputs (Figure 1.6a). SL can be applied to problems with input/output. In addition, SL can be used for prediction and classification, such as image recognition and filtering SPAM emails. In this thesis, we have used the Bayesian approach in CDS perceptor, which is a type of modeling that can extract the posteriori of the data.



(a)



(b)

Figure 1.6: Schematic of two popular machine learning approaches (a) supervised learning (SL) (b) reinforcement learning (RL) [1.35]

Reinforcement learning

Figure 1.6b shows a schematic diagram of reinforcement learning (RL). RL maps a decision-making problem into the interaction of a computer agent with a dynamic environment by trial and error [1.17]. The computer agent attempts to reach the best reward based on feedback received from the dynamic environment when it searches the state and action spaces. For example, in healthcare applications, the RL algorithm will try to improve the model parameters based on iteratively simulating the state (User health condition). Then, after applying the action (e.g., activating or deactivating sensors, amount of medication delivery and modeling accuracy), the agent obtains the feedback reward from the environment (healthy or unhealthy approval by MDs in the clinic). The RL algorithm can then converge to a model that may generate optimal actions [1.36].

In summary, RL learns through trial and error from interaction with a dynamic environment such as learning to play a game or a movie/video recommendation system. There are states and actions in RL. Typically, no database is required for RL and it can find the action for optimizing the reward. But RL requires that a model be extracted using examples and experiences. RL receives the reward/punishment from the dynamic environment. In this thesis, the executive uses RL for applying cognitive actions on a NGNLE.

1.6.2. Proposed cognitive dynamic system

Figure 1.7 shows the conceptual implementation of our proposed CDS for the NGNLE. By combining SL and RL as two main approaches of machine learning, CDS may be considered as an enhanced AI. The perceptor of the CDS can extract the model using SL algorithm. Using the extracted model, the perceptor can generate internal reward and predict the dynamic environment outcome [1.8]. The dynamic environment is the NGNLE with finite memory (e.g., a long haul OFDM fiber optic link). The perceptor sends the internal reward to the executive through the feedback channel. RL in the executive uses the internal reward in the current PAC to find an action that can decrease/increase the internal reward for the next PAC. However, the proposed CDS

can predict the outcome of the actions using a virtual NGNLE to make sure that the action can optimize the internal reward within predefined policy requirements.

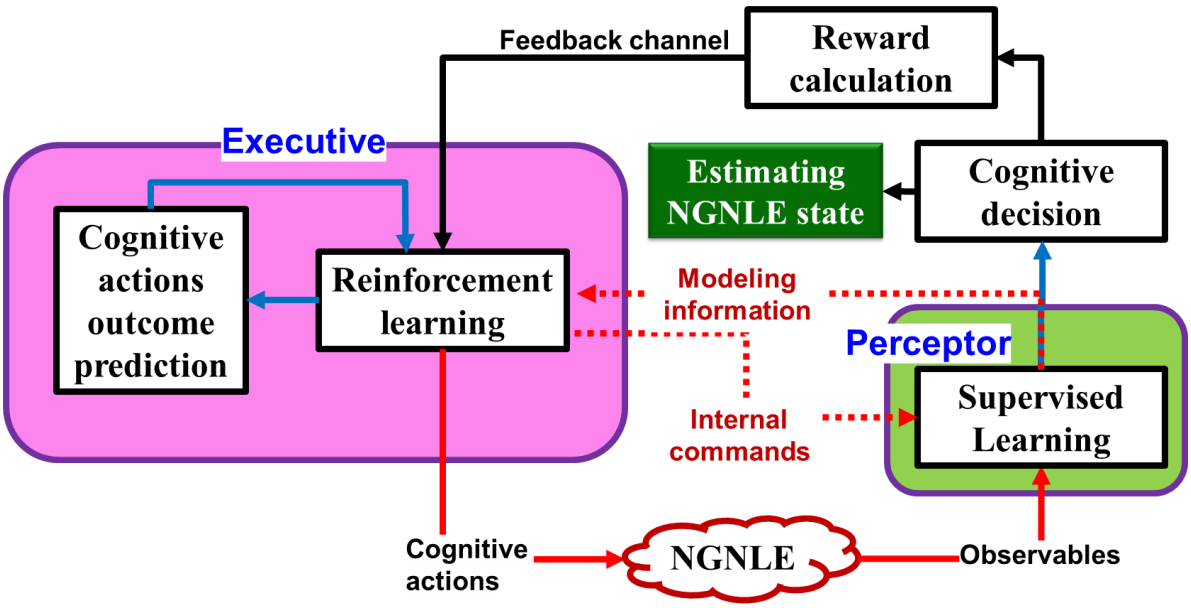


Figure 1.7: Conceptual implementation of proposed CDS [1.35]

In summary, unlike the typical RL, the executive of the CDS can use the extracted model by the perceptor to apply a cognitive action on the dynamic environment. The internal reward gives the CDS self-awareness, self-consciousness and independence from the dynamic environment. Briefly, the CDS has the "conscience" about the action. For extracting the internal reward, the SL algorithm should be applied in the perceptor. Also, the prediction of cognitive actions outcome is inspired from human brain imagination and risk assessment before doing any action in the real world. However, in a typical RL, the agent applies the actions blindly to receive feedback from the environment. Therefore, the CDS is a more appropriate choice rather than a typical RL in intelligent machine applications.

1.7. Contribution of the thesis

This research work has resulted in contributions to three main areas: CDS, long-haul fiber optic links and smart e-Health systems.

1.7.1. Contributions on CDS

Five versions of CDS are developed for NGNLE with simple, straightforward and faster PAC algorithms. Besides, the concept, design, and algorithms of CDS are presented for all five versions. Here, we provide the evolution of all five versions (see Table 1.1):

1. **CDS v1:** The concept of CDS v1 with a simple executive is used for a long-haul fiber optic link, which is an example of a NGNLE [1.10]. However, the simple executive of CDS v1 cannot predict the outcome of the actions before applying actions on the environment. Besides, the simple executive cannot control the modeling configuration of the perceptor.
2. **CDS v2:** CDS v2 is used as an alternative to AI in ADMS and for cognitive decision making on a NGNLE [1.37]. CDS v2 can control the modeling configuration of the perceptor by increasing or decreasing the decision tree level through internal commands. As a result, this version of CDS can model a NGNLE with finite memory. However, the executive cannot predict the outcome of the actions using a virtual NGNLE.
3. **CDS v3:** CDS v3 is presented as a general concept, design and algorithms of the CDS for the CDM in NGNLE. The CDS v3 uses the advanced executive [1.11]. The advanced executive can predict the outcome of multiple actions before applying an action to the environment using a virtual NGNLE. In addition, the advanced executive can change the modeling configuration of the perceptor through the internal commands. However, the perceptor of CDS v3 cannot model a NGNLE with finite memory.
4. **CDS v4:** The perceptor and executive of CDS v3 are upgraded as a general-purpose algorithm of the CDS for a CDM system in an NGNLE with finite memory [1.35]. Therefore, the CDS v4 may be considered as the more general form of CDS v3, and the perceptor can extract the model of the NGNLE with finite memory. The CDS v4 uses an advanced executive that can predict the outcome of multiple actions before applying an action to the environment with finite memory. In addition, the

advanced executive can change the modeling configuration of the perceptor through internal commands and increase or decrease the focus level by changing the decision tree level.

5. **CDS v5:** As mentioned earlier, the natural deficiency of ML-based AI in comparison to rule-based AI is a dependency of ML-based approaches to reliable datasets. Therefore, the performance of ML-based AI decreases significantly in the presence of a defective dataset. Here, a defective dataset means that the dataset does not have enough training patterns, has poorly labeled training patterns or both. A defective dataset can be a result of human errors or a hidden cyber-attack. CDS v5 uses conflict-of-opinion (CoO) decision making to mitigate the effect of a defective dataset and provide reliable results (see chapter 7). CDM based on CoO, unlike the earlier versions of CDS (CDS v1-v4), is placed in the executive instead of perceptor. Also, in CDS v5, the PAC concept is generalized to the PMAC concept. Also, the perceptor of CDS v5 can extract the NGNLE model as a jungle of decision trees.

Table 1.1: Evolution of five versions of CDS

CDS version	Virtual actions	Internal commands	Modeling with memory	Cognitive decision making	NGNLE for proof of concept case study
V1: Simple CDS [1.10]	✘	✘	✘	MAP rule	OFDM long haul fiber optic link
V2: ADMS [1.37]	✘	✓	✓	(100% selection)	Diagnostic test (Health)
V3: Advanced CDS [1.11]	✓	✓	✘	MAP rule	OFDM long haul fiber optic link
V4: Advanced CDS with focus level [1.35]	✓	✓	✓	MAP rule	OFDM long haul fiber optic link
V5, ADMS with non-monotonic reasoning*	✘	✓	✓	Conflicts of opinions (CoO)	Health screening

* Non-monotonic reasoning: decision can be invalidated by adding more evidence or knowledge

1.7.2. Contribution to fiber-optic communication

We applied CDS v1, v3, and v4 on a long haul OFDM fiber optic link as a example of NGNLE. The CDS can always keep the decision-making error under the system threshold, and this is a typical function of the CDS in a fiber optic communication system. Additionally, unlike digital back propagation (DBP), the proposed CDS does not require the fiber optic system parameters, such as the number of fiber spans, span length, dispersion and nonlinear coefficients of each fiber span. Intelligent perception processing by the CDS can extract a statistical model of a fiber-optic channel. Moreover, the CDS can learn and recognize a disturbance in the optical network, such as a variation in data rate, fiber length or power fluctuations. CDS v1 improves the Q-factor by 2.74 dB and the data rate is enhanced by 12.5% [1.10]. CDS v3 can provide 23.3% data-rate efficiency enhancement as well as 3.5 dB Q-factor improvement using the proposed fast algorithms [1.11]. Also, the CDS v3 reaches the predefined goal faster (in 8 PACs), while without action-outcome prediction, the CDS requires 12 PACs. CDS v4 can provide ~43% data-rate efficiency enhancement and 7 dB Q-factor improvement [1.35]. Besides, CDS v4 can reach the predefined goal in 9 PACs. However, a CDS without action-outcome prediction in the executive requires 20 PACs. A comparison between the CDS v1, CDS v3, and CDS v4 for the long-haul fiber optic communications case study and typical DBP method [1.40] is given in Table 1.2.

Table 1.2: Comparison between CDS v1, v3, and v4 and typical DBP method

References	Technique implemented	Q-factor improvement	Data rate enhancement
CDS v1 [1.10][1.38]	Simple CDS	2.7 dB	13%
CDS v3 [1.11][1.39]	CDS with action outcome prediction (Virtual actions)	3.5 dB	23%
CDS v4 [1.35]	CDS using finite memory modeling + Virtual actions	7 dB	43%
Typical nonlinearity mitigation method [1.40]	Digital back propagation (DBP) for OFDM fiber optic link	< 2 dB	-

1.7.3. Contribution to healthcare applications

The contribution of the CDS on healthcare applications are into two main areas as an automatic diagnostic test and an automatic screening process. The automatic diagnostic test implemented using the CDS v2 and an automatic screening process is implemented using the CDS v5. For both versions, concept, architecture structures, and algorithms are provided.

Automatic diagnostic test (CDS v2 [1.17])

ADMS based on CDS v2 is applied to Arrhythmia diagnosis. It is shown that the CDS v2 performs with 95.4% accuracy. In Table 1.3, a comparison between the proposed method and some related works on this Arrhythmia database decision-making accuracy is shown.

Table 1.3: Comparison CDS v2 and related published works

Technique implemented	Best reported accuracy (%)	Sensitivity (Diagnosis of abnormal rhythm accuracy) (%)	Specificity (Normal rhythms accuracy) (%)
Random forest (RF)+Support vector machine (SVM) [1.41]	77.4	59.9	91.4
Deep learning [1.42]	75.8	-	-
SVM for 2 classes and 11 features [1.43]	86	-	-
ADMS using CDS v2 [1.37]	95.4	90	100

Automatic Screening process (CDS v5)

Here, the screening process means screening between healthy and unhealthy with a predefined level of diagnosis error at an acceptably high false alarm rate. Defective datasets are so prevalent in healthcare applications. However, some defective datasets can be used for the screening process. A proof-of-concept case study using CDS v5 is applied in screening for Arrhythmia from a defective dataset. It is shown that the proposed CDS performs well with good agreement to the desired diagnosis errors of 25%, 10%, 5.9%, and 2.5%, achieving average final diagnosis errors of 13.2%, 9.9%, 6.6%, and 4.6%, respectively. These diagnosis errors with a defective dataset

correspond to acceptable high false alarm rates of 20.1%, 25%, 28.4%, and 54.7%, respectively.

1.7.4. Peer-Reviewed Publications:

1. M. NaghshvarianJahromi, S. Majumdar, S. Kumar, M. J. Deen, Natural Brain-Inspired Intelligence for Healthcare Screening Applications, Submitted to: *IEEE Access.* (CDS v5)
2. Naghshvarianjahromi, M.; Kumar, S.; Deen, M.J. "Natural Brain-Inspired Intelligence for Non-Gaussian and Nonlinear Environments with Finite Memory". *Appl. Sci.* 2020, 10, 1150. (CDS v4)
3. Naghshvarianjahromi, M.; Kumar, S.; Deen, M.J., "Brain-Inspired Cognitive Decision Making for Nonlinear and Non-Gaussian Environments". *IEEE Access* 2019, 7, 180910–180922. (CDS v3)
4. Naghshvarianjahromi, M.; Kumar, S.; Deen, M.J. "Brain-Inspired Intelligence for Real-Time Health Situation Understanding in Smart e-Health Home Applications". *IEEE Access* 2019, 7, 180106–180126. (CDS v2)
5. M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain-Inspired Dynamic System for the Quality of Service Control Over the Long-Haul Nonlinear Fiber-Optic Link", *Sensors,* 2019. (CDS v1)

1.7.5. Conference/Symposium participation:

1. M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain-Inspired Dynamic System for the Quality of Service Control Over the Long-Haul Nonlinear Fiber-Optic Link", *16th CWIT 2019,* June 2019.
2. M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Smart long-haul fiber optic communication systems using brain-like intelligence", *16th CWIT 2019,* June 2019.

1.8. Thesis Overview

The contents of this thesis are organized as follows. In Chapter 1, a brief background of human cognition, conventional CDS, and proposed CDS for NGNLE are introduced, and then motivations of this work are presented. Then, the contributions from this research are given.

In Chapter 2, the theoretical concepts of NGNLE are explained. Then, we provide background on two examples of NGNLE, i.e. long-haul fiber-optic link and healthcare application. Lastly, the objectives of the implementation of CDS on these two NGNLE examples are discussed.

In Chapter 3, the simple CDS, or CDS v1, is presented. It is shown how CDS v1 can improve the performance (even in the presence of disturbance) of a fiber optic link.

In Chapter 4, CDS v3 or CDS with the action-outcome prediction for a NGNLE is presented. In this chapter, it is demonstrated how the internal commands for the adaptive modeling configuration and action-outcome prediction provide more data-rate enhancement and make the CDS faster to achieve predefined goals.

In Chapter 5, CDS v4 is presented for NGNLE with finite memory. Both executive and perceptor of CDS v4 are upgraded for extracting adaptive modeling of NGNLE with finite memory. Implementing CDS v4.0 on the long-haul fiber-optic link shows that adaptive modeling with finite memory can provide significant data rate enhancement in comparison to earlier versions of CDS for long haul fiber-optic links. Similar to CDS v3, CDS v4 can use action-outcome prediction applied on a virtual NGNLE in the executive. Prediction outcomes of actions make CDS v4 fast in achieving predefined goals for the CDS.

In Chapter 6, CDS v2 based on decision tree for a diagnostic test in a smart e-Health home is presented. Concepts, algorithms, design and the simulation results for an arrhythmia case study are proposed in this chapter. It is shown that an ADMS using CDS can provide noticeable improvement in diagnosis accuracy compared to related published works.

In Chapter 7, CDS v5 based on the jungle of decision trees and CoO is presented for the screening process in a smart e-Health system. First, a generalized form of the PAC

is presented as PMAC. In addition, the motivation for using CoO in the presence of a defective dataset in healthcare applications is provided. Then, the detailed equations and algorithms for the screening process using CDS v5 are discussed. After that, simulation results for diagnosis error and false alarm are presented.

In Chapter 8, the conclusions of this research are presented. Recommendations for future work on CDS for NGNLE are briefly discussed with the main focus on the potential for implementation in a practical and commercial fiber optic link and disease class diagnosis in healthcare applications.

Chapter 2

CDS FOR NGNLES WITH FINITE MEMORY

In this chapter, the relevant fundamentals of NGNLEs problem with finite memory are described. A nonlinear barrier in long-haul fiber optic for improving the data-rate is discussed. Then, two examples for the NGNLE are provided: (i) a fiber optic communication system based on OFDM, and (ii) a smart eHealth system. Then, the similarities and differences between these two examples for implementing the CDS are discussed.

2.1. Non-Gaussian and nonlinear environment (NGNLE) with finite memory

As mentioned before, a NGNLE means that most of the observables (outputs) from the measurement are not linear functions of unknown NGNLE inputs. Also, the observables do not have a Gaussian distribution. In most NGNLE applications, we can measure the observables as the output. However, since the current situation of a NGNLE is unknown, we need to decide between *M* discrete known situations.

Suppose $\mathbf{X}^n = [X_1^n, X_2^n, \ldots, X_S^n]$ is a vector of variables as a function of discrete time *n* corresponding to *S* known discrete states or situations (See Figure 2.1). For example, in communication systems that uses QAM-4, there are four possible symbols, $X_1^n, X_2^n, X_3^n \text{ and } X_4^n$ (00, 01, 10, 11) which vary as a function of time. The receiver may select X_2 out of four possible symbols as the symbol that was transmitted at time *n*. The relation between observables and a NGNLE situation can be defined as follows:

$$\mathbf{Y}_n^m = g_n(X_1^n, X_2^n, \ldots, X_S^n, w_n), \tag{2.1}$$

$$\mathbf{Y}_n^m = [Y_{n-m}, Y_{n-m+1}, \ldots, Y_n]. \tag{2.2}$$

Here, $\mathbf{Y}_n^m$ is a set of observables that is extracted from measured signals from the sensors (e.g., [2.1][2.2]) from discrete time $(n-m)$ to n, m is the current focus level that can be used for the reasoning with acceptable complexity, and w_n represents noise with an arbitrary probability density function (PDF), which could be non-Gaussian. As mentioned before, X_s^n corresponds to the current NGNLE condition or situation, which we want to estimate. In (2.1), $\mathbf{Y}_n^m$ is known by measurements or another information gathering technique, but, X_s^n is unknown. The $g_n(\cdot)$ is a nonlinear mapping of $\mathbf{X}^n$ and w_n to $\mathbf{Y}_n^m$. One example of such a system can be seen in a long-haul orthogonal frequency division multiplexing (OFDM) fiber optic link. In the next section, we briefly describe the OFDM fiber optic link.

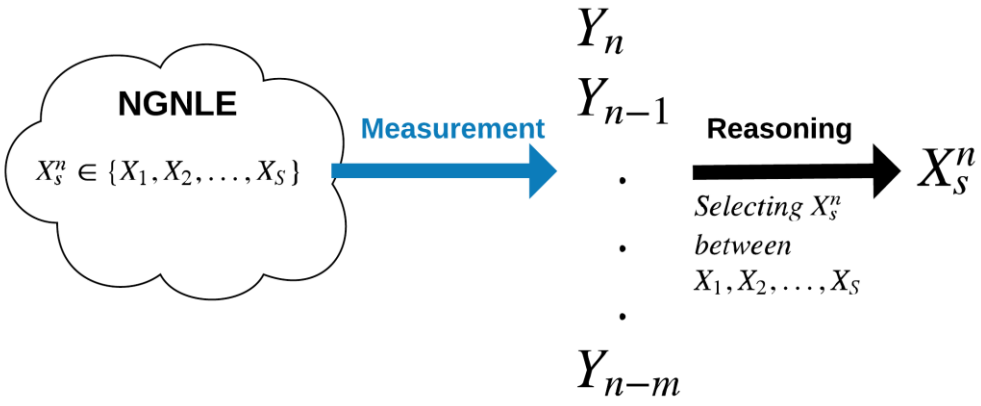


Figure 2.1: Decision making between *S* known situations using *m* focus levels

2.2. Fiber optic bottleneck and nonlinearity barrier

2.2.1. Fiber-optic communication systems

Wired fiber-optic communication systems can transmit information from one place to another place by sending electromagnetic lightwave signals through a fiber optic-link [2.3][2.4]. A typical fiber-optic link has three main subsystems, an optical transmitter, an optical fiber, and an optical receiver. The transmitter of the fiber optic communication system using an optical modulator converts an electrical signal to an optical signal. After the signal passing through the optical fiber, the receiver of fiber optic communication system converts the optical signal into an electrical signal. The key advantage of communication systems using optical fiber is the huge available bandwidth, which results in high data-rate transmission. This is mainly because the carrier frequency of the optical communication system is a high-frequency laser with a center frequency of ~200 THz. This center frequency is significantly higher than that of the microwave or mm-wave systems (300 MHz-300 GHz).

2.2.2. Nonlinearity distortion: source and effect

During propagation of optical signals in an optical fiber, due to fiber loss, dispersion and nonlinearity, the receiver will receive a distorted signal. The loss of silica optical fiber optic is about 0.2 dB/km. In the long-haul fiber-optic link, attenuation of the signal is significant and should be compensated using erbium-doped fiber amplifiers (EDFAs). However, the EDFAs add amplified spontaneous emission (ASE) Gaussian noise. Another issue in the transmission of a signal by fiber optic link is the frequency dependence of the refractive index that results in a dispersive fiber-optic medium. Therefore, different frequency components of transmitted signals have different speeds, which results in the broadening of the received signal and inter-symbol interference (ISI). These linear distortions can be significantly mitigated in the fiber-optic link. However, fiber-optic link non-linear impairments are the main source of distortions [2.4]. Kerr effect is the dominant reason for the nonlinear effect in a long-haul fiber-optic link. Kerr effect results from the dependence of the refractive index on the intensity of the optical transmitted pulse.

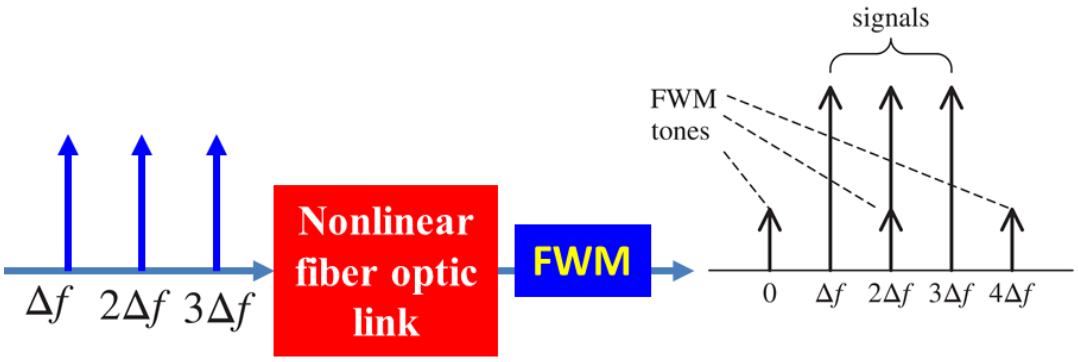


Figure 2.2: Conceptual nonlinear interaction in an optical fiber, Four-wave mixing (FWM), and cross-phase modulation (XPM).

In an orthogonal frequency division multiplexing (OFDM) based fiber-optic system, nonlinear distortion due to the Kerr effect can be deterministic such as self-phase modulation (SPM), cross-phase modulation (XPM) and four-wave mixing (FWM) between subcarriers or stochastic nonlinear impairments such as the interaction between amplified spontaneous emission (ASE) and Kerr effect leading to Gordon-Mollenauer phase noise [2.5]. As shown in Figure 2.2, when signals using multiple channels transmitted in a single optical fiber, the XPM effect leads to a phase shift in the signal channel, which results from another signal channel power fluctuations [2.4]. Scattering of the incident photons results in the FWM effect. FWM is a nonlinear interaction between signals at frequencies Δf , $2\Delta f$ and $3\Delta f$ resulting in FWM tones at $0, 2\Delta f$ and $4\Delta f$ [2.4] (see Fig. 2.2) If there is a channel at the location of a FWM tone, it acts as noise leading to performance degradations.

Currently, the nonlinear distortions in the long-haul fiber-optic link are the main reason for limiting the performance of an optical communications system to reach the potential data-rate based on optical available bandwidth [2.6]. Potential data-rate means achievable data-rate when nonlinear distortion is adequately mitigated. Figure 2.3 shows the spectral efficiency of fiber optic link for different distances without using nonlinear compensation methods. Here, for a given bandwidth, spectrum efficiency means the number of bits that can be transmitted per symbol in an optical communications system [2.6]. In addition, Figure 2.3 shows that after the optimum point, by increasing SNR, the spectral efficiency of fiber decreases significantly.

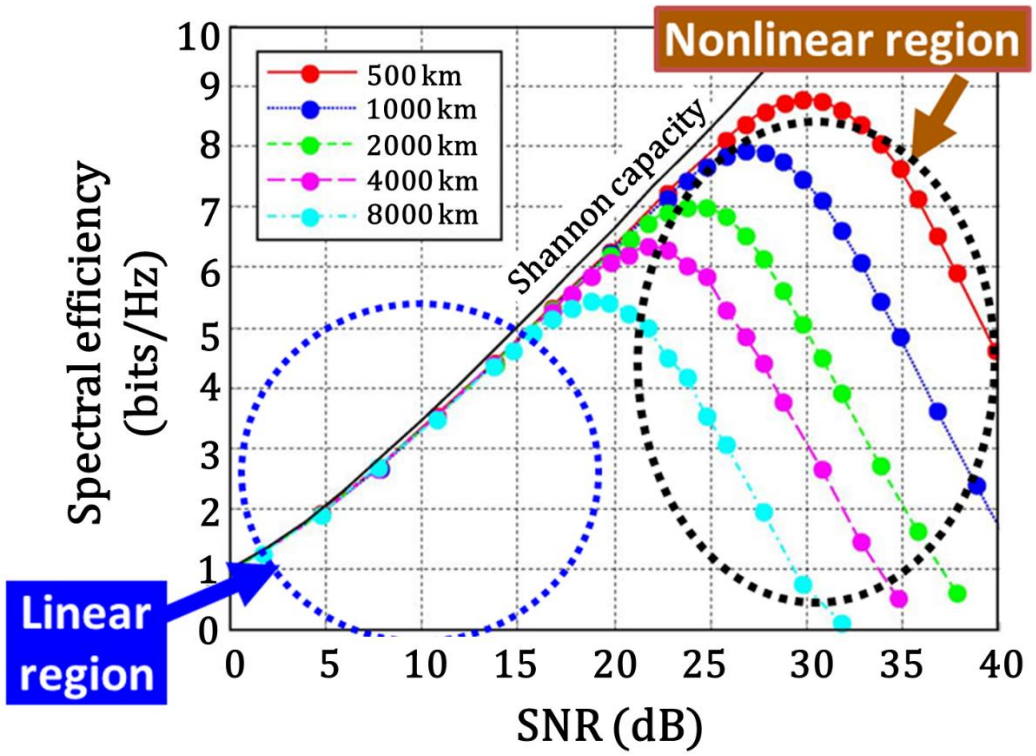


Figure 2.3: Spectral efficiency of fiber optic link for different distances without using nonlinear compensation methods [2.6] (This figure is from [2.6] under IEEE Thesis/Dissertation Reuse permission)

2.2.3. The motivation for upgrading fiber optic communication with CDS

Rapid advances in fiber-optic communication technologies have been continuing for several decades. Capacity of data transmission by a fiber-optic link is enhanced by a factor of 10 every 7.5 years [2.7] (see Figure 2.4). However, the data rate of the wireless communications is increasing by a factor of 10 every 4 years [2.7]. As shown in Figure 2.4, it is expected that indoor wireless communications data-rate can match the wired fiber-optic communications data-rate in ~2030.

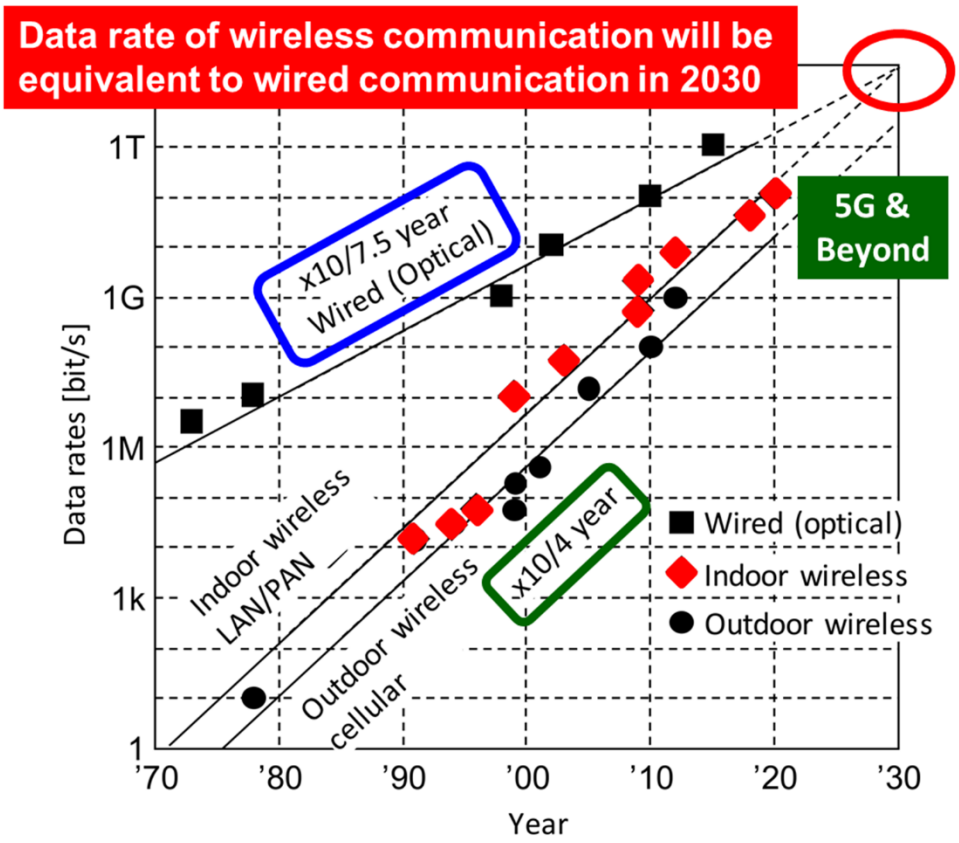


Figure 2.4: Evolution of data rate of wired optical communication and wireless communication [2.7]

The fiber-optic communications networks are the backbone of current modern communications such as internet, TV, and tele-medicine. However, current data-rate improvement of fiber-optic communications cannot handle future communications requirements. Therefore, there is need for a method that significantly increase the fiber-optic communications data-rate, is software-defined (cheap) with acceptable computation cost.

Currently, the linear distortion of the fiber-optic link can be compensated adequately. However, nonlinear distortions in a fiber optic link cannot be efficiently removed by conventional nonlinear mitigation methods in practical applications. These conventional methods can be classified into three categories (i) optical techniques, such as optical phase conjugation [2.8][2.9] and optical backpropagation [2.10][2.11][2.12][2.13] (ii) Optoelectronic techniques, such as compensation using a phase modulator [2.14] (iii) Digital techniques, such as digital backpropagation (DBP)

[2.15][2.16], perturbation techniques [2.17][2.18], nonlinear Fourier transform [2.19][2.20][2.21], and Volterra series approach [2.22][2.23][2.24]. The optical/optoelectronic techniques require additional photonic/optoelectronic components, while the digital techniques such as DBP, require computational architecture with higher cost. In addition, these techniques require the knowledge of fiber optic system parameters such as the number of fiber spans, span length, dispersion, and nonlinear coefficients of each fiber span. In addition, each time a data sequence is sent, extensive signal processing is done by these techniques. The signal processing is data dependent due to the nonlinear nature of the system. Although there have been recent efforts to make the DBP adaptive [2.25][2.26], the computational cost is high. As a result, we need a technique that is adaptive, intelligent and works without any previous knowledge about the photonics/optoelectronic components in the network. It should have a low cost and be easy to install (software-defined). In addition, the technique should adapt to any fluctuations in the network such as power fluctuation or link length variations. Thus, we used CDS to upgrade a wired optical communications system that is designed to meet these requirements.

2.3. OFDM-based fiber optic link with CDS

In this Section, application of a CDS to the fiber optic link is presented. Figure 2.5 shows the block diagram of a fiber-optic system with CDS. The executive, the perceptor, and the feedback channel are placed in the receiver (Rx), and the executive sends actions such as new launch power to the fiber optic link by a low data rate link to the transmitter (Tx). This schematic is valid for CDS v1, v3 and v4. However, due to the assumption of the fiber parameters being unknown to the receiver, for CDS v1 linear distortions are partially compensated. However, CDS v3 and v4 are implemented on fully mitigated linear distortion received signals. It may be noted that the blind equalization of fiber dispersion using least mean squares (LMS) or constant modulus algorithm (CMA) is possible, although its computational cost is higher than the case when the fiber dispersion is known to the receiver [2.4]. In Figure 2.5, Y_n^k and X_n^k are received and transmitted symbols, respectively, and $\bar{X}_n^k$ are symbols after cognitive

decision making by the perceptor. Here, *k* is the current PAC index and *n* is discrete-time.

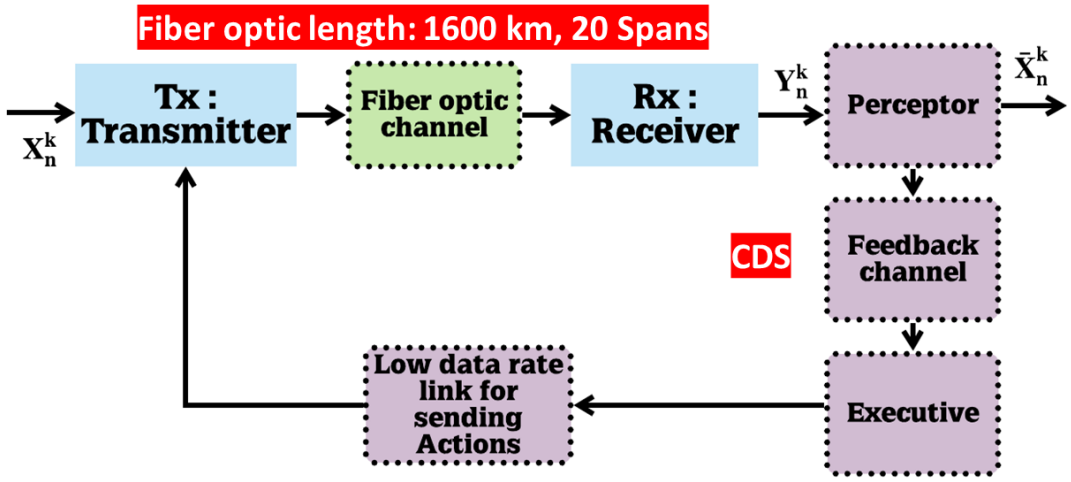


Figure 2.5: The basic design for quality of service (QoS) control and bit error rate (BER) improvement by CDS for the fiber-optic link.

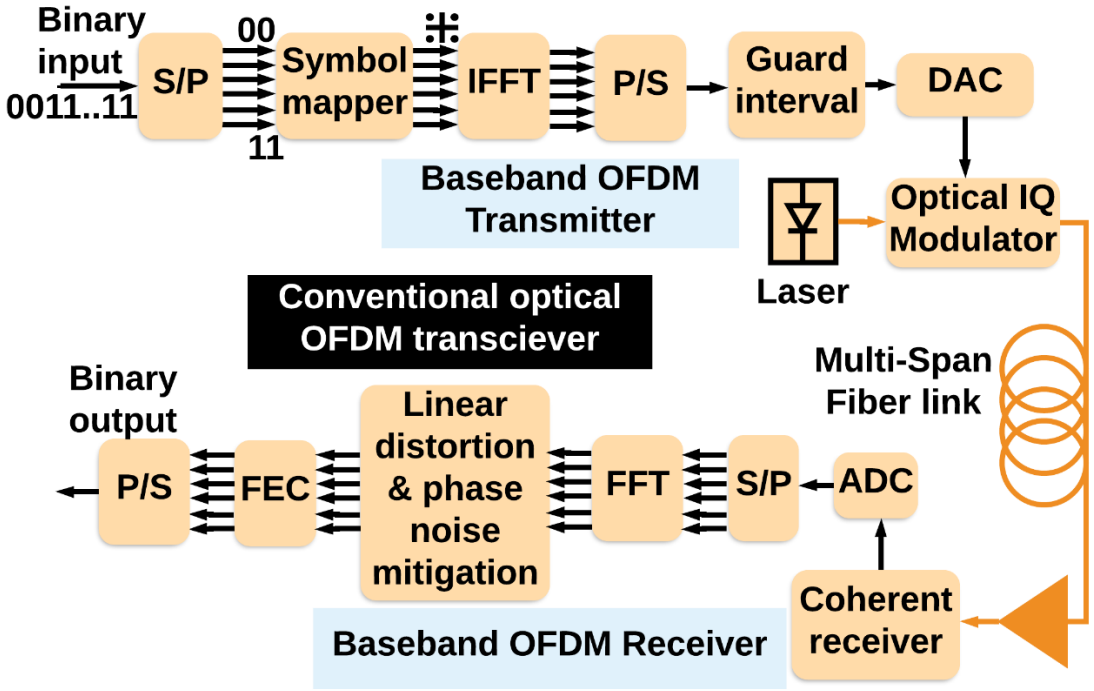


Figure 2.6: The conventional OFDM-based fiber optic system [1.11].

(ADC: Analog to Digital Converter, DAC: Digital to Analog Converter, FEC: Forward Error Correction, FFT: Fast Fourier Transform, IFFT: Inverse Fast Fourier Transform, P/S: Parallel to Serial and, S/P: Serial to Parallel)

2.3.1. OFDM-based Fiber Optic System

OFDM has received significant attention from the researchers in optical communication systems [1.40][2.2][2.26]. The detailed description of OFDM-based fiber optic system are presented in[1.40][2.4]. Figure 2.6 shows the conventional OFDM-based fiber optic system and Figure 2.7 shows the system with CDS (it is the same for CDS v1, v2 and v3). In Figure 2.7, $\hat{X}_n^k$ are the training symbols for fiber optic model extraction. Also, in Figure 2.7, we assumed that the CDS is placed in the receiver, but the CDS can also send commands to the transmitter, such as changing the data-rate or the transmitted signal power. The binary data is mapped to symbols using a symbol mapper. For example, if the symbol is '11', the corresponding amplitude X_n is (1+1j) and if the symbol is '00', the amplitude is (-1-1j). At the transmitter, N subcarriers modulated by low data rate QAM-16 data are multiplexed using the inverse fast-Fourier transform (IFFT). A cyclic prefix (CP) is used in the guard interval between OFDM frames to preserve the orthogonality of the subcarriers and also to avoid inter-carrier interference (ICI) due to dispersive effects in the fiber [2.2]. After digital-to-analog conversion (DAC), the OFDM data modulates the laser output using an in-phase quadrature (IQ) modulator. The output of the IQ modulator passes through a fiber optic link consisting of several spans of standard single-mode fiber (SSMF) and each fiber span is followed by an inline EDFA. The signal propagation in the fiber optic link is governed by the Nonlinear Schrödinger equation (NLSE) which includes fiber dispersive effects, loss and third-order nonlinear susceptibility (also known as Kerr effect). The NLSE models the complex envelope of the signal and by solving it, we find the complex envelope of the signal at the output of the link given that the complex envelope of the signal at the transmitter is known.

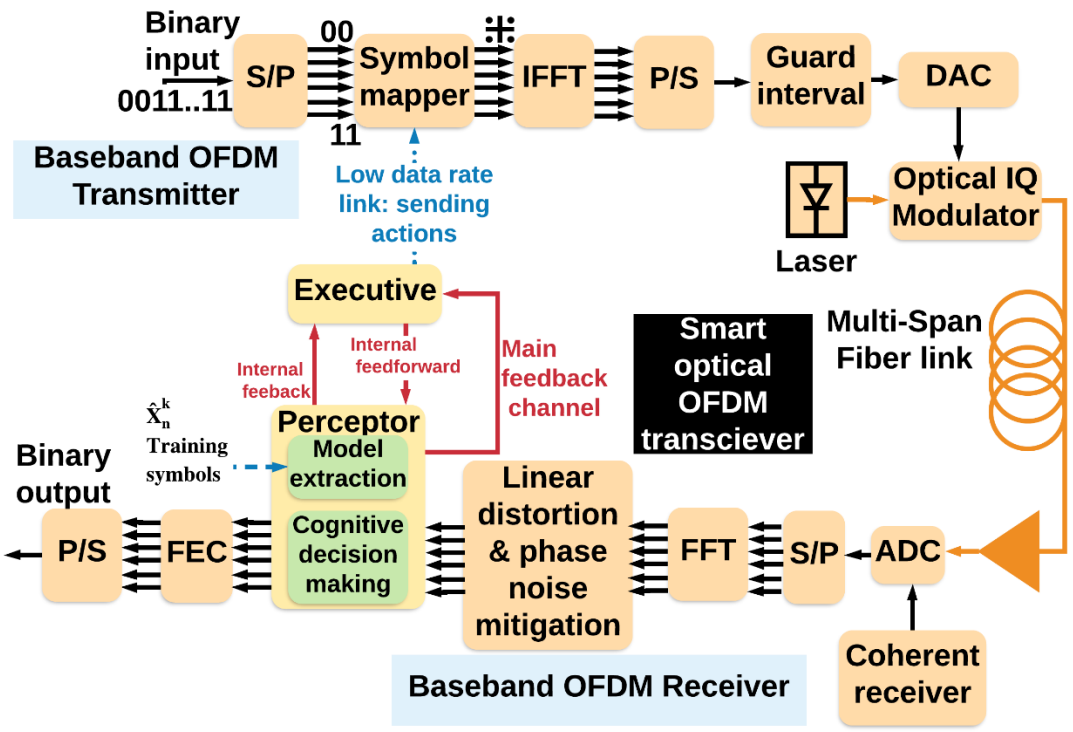


Figure 2.7: The OFDM-based fiber optic system enhanced with the cognitive dynamic system (CDS) [1.11].

The output of the fiber optic link passes through a coherent receiver in which the OFDM signal is down-converted to baseband. After analog-to-digital conversion (ADC), the subcarriers are demultiplexed using the fast Fourier transform (FFT). The adaptive linear equalizer mitigates the linear distortions of the data in each subcarrier in parallel. The phase noise of the transmitter laser and that of local oscillator (LO) are compensated using the standard technique [1.40][2.4]. But, nonlinear distortions in a fiber optic link cannot be efficiently removed by conventional nonlinear mitigation methods in practical applications. The QAM symbols modulating each subcarrier pass through the perceptor for statistical modeling and cognitive decision-making (see Figure 2.7).

2.3.2. Long-haul Fiber Optic OFDM Link as an Example of NGNLE

Typically, in the analysis of long-haul fiber optic systems, the fiber optic channel is assumed to be non-linear and the noise is assumed to be Gaussian to simplify the

problem [1.31]. Under the Gaussian noise assumption, it follows that the variance of nonlinear noise should be proportional to the total distance L. However, in [1.31], it was shown that the variance of nonlinear noise scales as L^x where $x \in (1, 2)$ depending on the dispersion map. Therefore, the Gaussian noise assumption is not accurate for OFDM systems in the presence of nonlinearity. In this thesis, we investigate the performance of a long-haul fiber-optic OFDM link that is an example of a NGNLE.

2.4. CDS for Smart e-Health system/home as the second example of NGNLE

2.4.1. Motivation

The modern healthcare system highly advanced, but it is not cost-effective. Also, for people who live alone, there is no witness to describe their symptoms or they forget to describe it. Especially during the Covid-19 pandemic, when people stay home more than before, similar to the USA and UK there are some excess deaths in Canada [2.27], Two questions can be asked by experts: (i) are they missing Covid-19 deaths? (ii) are they dying because of heart attacks? It seems some people afraid to go to the hospital because of fear about Covid-19. Another issue regarding the current healthcare system is the time-consuming procedure of diagnosis diseases. Ref. [2.28] provides information a man died in New Zealand because an expert missed to read his CT scan results showing cancer for more than one year. In [2.29], it is mentioned that medical doctors' errors are the third cause of death in the USA after heart diseases and cancer. Therefore, we need a system to take care of patients automatically. This system should be cost-effective for real-time monitoring and recording users' health state history.

The solution for that is using CDS as the brain of a smart e-Health system/home for information processing and ADMS. ADMS using CDS can be used for different situations or recommendations with minimal human intervention. Note that ADMS is not intended to replace medical doctors, but the ADMS using CDS can be inspired by MDs decision-making approaches. Also, it can perform time-consuming tasks, and

handle emergency issues in which time is a critical element in saving the life of the User using real-time monitoring at home. Therefore, Users can have health monitoring as needed in a comfortable environment (their home) without interrupting their daily activities.

2.4.2. Smart e-Health system using CDS (home)

Similar to concept we mentioned in section 1.1, a smart e-Health system is the sub-area of smart systems. Therefore, Figure 2.8 shows the smart e-Health system architecture as a cyber-physical system (CPS) based on the smart systems concept. A specific example of a smart e-Health system is smart e-Health home. A smart e-Health home can be considered to have four main layers [1.4]:

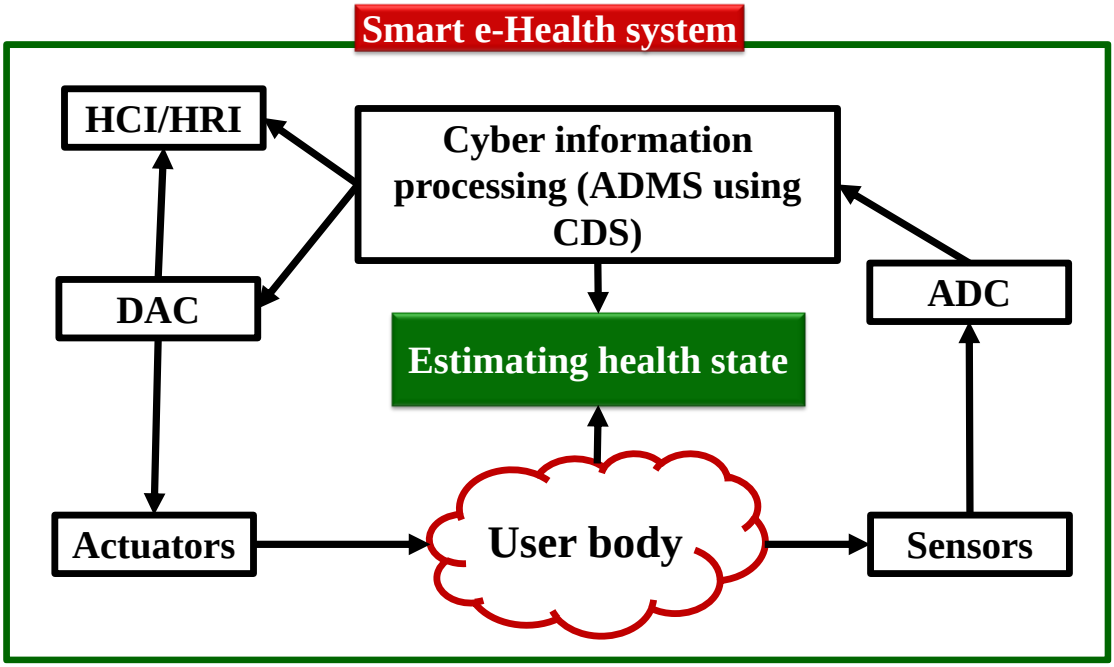


Figure 2.8: Smart e-Health home architecture with an autonomic decision-making system (ADMS) using cognitive dynamic system (CDS) as the cyber physical system (CPS) [1.37]. ADC: Analog-to-digital converter, DAC: Digital to analog converter, HCI: Human-computer interaction, HRI: Human-robot interaction.

- **Sensors and Actuators:** These are environmental and medical sensors, appliance and home control units, or any device or methods providing

medical information between Users (humans) and computers/robots in human-computer interaction (HCI) or human-robot interaction (HRI).

- **Home communication network:** This layer is responsible for information gathering and management, feature extracted from signals and discovery of appliances or sensors.
- **Autonomic computing:** This layer is responsible for knowledge management, situation understanding of the inhabitants such as whether the User has a disease or not, intelligent reasoning and decision making. This cyber part of the smart e-Health home is termed as the ADMS for information processing of the measured signals from the sensors.
- **Safety and healthcare services:** This can include health-care services (hospital/doctors), safety services (police, fireguards), remote support, telemedicine, e-Health networks and other smart e-Health services.

As we mentioned in section 1.1, we focus on the autonomic computing layer of a smart e-Health home. Smart e-Health home is an example of NGNLE. The CDS-based ADMS of smart e-Health home is responsible for information processing of sensory data that are related to the health of the inhabitants (Users).

2.4.3. Related works

Here, we provide a very brief review of related published methods using machine learning or AI techniques. These techniques are applied for diagnostic testing or decision making in healthcare applications. Recently, because of the increasing popularity of wearable and portable health sensors, health records and data have rapidly grown. As a result, a large number of healthcare datasets have been generated [2.30]. To predict clinical outcomes or clinical problems, clinicians and researchers apply machine learning and AI algorithms using available datasets [2.31]. Also, machine learning is being applied to the diagnosis of various diseases [2.32][2.33][2.34][2.35]. Besides, these techniques can also be used for disease treatment and optimal decision making [2.36][2.37][2.38].

In [2.39], machine learning was used to reduce false Arrhythmia alarm from electrocardiogram (ECG) signals. In [2.40, 2.41], it was shown that machine learning

can even be used to generate reports from medical images [2.40, 2.41]. Here, the generalizable prediction models can be extracted by machine learning-based approaches. Then, the patterns of the measured data can be extracted. Thus, extracted patterns help MDs to perform more personalized clinical prediction in patients [2.42].

2.4.4. CDS for an automatic diagnostic test in a smart e-Health system

Here, we implemented the proof-of-concept of ADMS using the proposed CDS for diagnostic tests or low false alarm policy. We focus on a low false alarm policy (someone has a disease or no disease) as the first stage of our work. For the implementation of a low false alarm policy, the CDS is inspired by the decision-making approaches of medical doctors (MDs). On the other hand, if a person with an unknown or a new disease visits an MD, then the MD can still make the diagnosis that the person is not healthy. The MD can diagnose the condition based on measured abnormal symptoms or features. Therefore, we choose the binary decision making of a User's health state as the first step in designing an ADMS using CDS.

The binary decision making between healthy and unhealthy persons has a key role in designing the ADMS of a smart e-Health home. This is because binary decision-making results can be used as the base for developing a future comprehensive ADMS of a smart e-Health home such as disease class diagnosis. Therefore, the proposed CDS for ADMS can be considered as the first step in designing a simple "cyber semi-medical doctor's decision-making system (CSMDDMS)" for situation understanding between a healthy and unhealthy User in the *autonomic computing layer* of a smart e-Health home.

2.4.5. CDS for automatic screening process using defective dataset

Datasets for healthcare applications is generally limited. In addition, the labeling of the healthcare dataset can be erroneous owing to its dependence on human skills. Moreover, a dataset can be inherently defective or manipulated by hidden cyber-attack. These shortcomings in the available data may result in an overfitted model, potentially causing the test accuracy of the model to drop significantly compared to the training accuracy. In such a case, one can infer that the used dataset for model extraction is

badly labeled and/or lacks enough training patterns required to extract a reliable, accurate and converged model. However, it is shown that some defective datasets can be used for the automatic screening process. Therefore, we focus on a clinically acceptable high false alarm policy (screening someone has a disease or no disease) as the second stage of our work for developing CDS for ADMS.

2.5. Fiber optic communications vs Healthcare applications

The principles of PAC for long-haul fiber-optic link and smart e-Health systems are identical. For implementing CDS, long-haul fiber optic link and smart e-Health systems have some similarities and differences. Both examples are non-gaussian and nonlinear environments and they require decision-making based on measured signals for unknown transmitted signals/health status. Also, the conventional structure of CDS for both are the same, but the details needs to be changed based on the differences discussed next.

2.5.1. Differences, and research challenges

The signal time period in fiber-optic communications is short (in picoseconds) while that in health applications is long (milliseconds). Because of the faster nature of optical signals in time-domain, there is no time for real-time features extractions in time/frequency domains in practical applications. However, time-domain healthcare signals are slow enough for real-time practical systems that can provide an advantage for health care applications to extract signals features in time/frequency domains. In fiber-optic communications, enough training data for extracting converged models can be easily provided in a tiny fraction of a second. However, in healthcare application, datasets with enough training patterns to provide a converged model is rare. Another difference is that there are no ethical problems for implementation or accessing data in optical communications, but there are ethical problems for the implementation of CDS or accessing data in healthcare applications. Also, the training data in fiber-optic communications are reliably labeled. However, the labeling of the healthcare dataset can be erroneous owing to its dependence on human skills.

2.5.2. AI aspects for fiber communications and healthcare applications

AI aspects implemented using the PAC concept for fiber optic are automatic control, problem-solving, decision making and reasoning, and perceptions. In this thesis, these AI aspects are implemented as the initial steps for achieving CDS main objectives for long-haul fiber optic communications. Here, CDS objectives for fiber-optic communications are controlling QoS and providing reliable communications over the fiber optic link as well as guarantee maximum achievable data rate or length. AI aspects implemented using the PAC concept for healthcare applications are problem-solving, decision making and reasoning, and perceptions. CDS applied for two objectives of automatic diagnostic test and automatic screening process.

.

Chapter 3

CDS V1 (SIMPLE CDS) FOR LONG-HAUL FIBER OPTIC LINK[1]

In this chapter, brain-inspired intelligence using the cognitive dynamic system (CDS) concept is proposed to control the quality-of-service (QoS) over a long-haul fiber-optic link that is nonlinear and with non-Gaussian channel noise. Digital techniques such as digital-back-propagation (DBP) assume that the fiber optic link parameters, such as loss, dispersion, and nonlinear coefficients, are known at the receiver. However, the proposed CDS in this chapter does not need to know about the fiber optic link physical parameters, and it can improve the bit error rate (BER) or enhance the data rate based on information extracted from the fiber optic link. The information extraction (Bayesian statistical modeling) using intelligent perception processing on the received data, or using the previously extracted models in the model library, is carried out to estimate the transmitted data in the receiver. Then, the BER is sent to the executive through the main feedback channel and the executive produces actions on the physical system/signal to ensure that the BER is continuously under the pre-defined forward-error-correction (FEC) threshold. Therefore, the proposed CDS is an intelligent and adaptive system that can mitigate disturbance in the fiber optic link (especially in an optical network) using prediction in the perceptor and/or doing proper actions in the executive based on the BER and the internal reward. A simplified CDS was implemented for nonlinear fiber optic systems based on orthogonal frequency division multiplexing (OFDM) to show how the proposed CDS can bring noticeable improvement in the system's performance. As a result, enhancement of the data rate by

[1] Most of this chapter was published as: M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain Inspired Dynamic System for the Quality of Service Control Over the LongHaul Nonlinear Fiber-Optic Link," *Sensors*, vol. 19, no. 9, pp. 2175-2195, 2019.

12.5% and the Q-factor improvement of 2.74 dB were achieved in comparison to the conventional system (i.e., the system without smart brain).

This chapter is organized as follows. In section 3.1, the introduction on CDS v1 as simple CDS is provided. Then, In Section 3.2, the concept of CDS is applied to a fiber optic system, which is nonlinear and the noise is non-Gaussian. The design of the perceptor and the executive for fiber optic applications are discussed in detail. In Section 3.3, the detailed algorithm for the proposed CDS is discussed. In Section 3.4, the proposed CDS is implemented for a long-haul fiber optic system based on OFDM. Numerical simulations of the fiber optic system with and without CDS are carried out and the improvements resulting from the use of CDS are discussed. Additionally, the computational complexity associated with CDS is compared with other digital techniques used for fiber optic communication systems. Finally, the conclusions are presented in Section 3.5.

3.1. CDS v1

In chapter, a simple cognitive dynamic system (CDS v1) based on ADMS concept is presented for the fiber optic communication systems. CDS can be used as the brain of fiber optic transceiver. As the first step, the principles of CDS are applied to a nonlinear fiber optic communication system for the bit error rate (BER) improvement and the data-rate enhancement. The typical CDS has three main subsystems: (i) perceptor, (ii) main feedback channel, and (iii) executive. The main contribution of this chapter is the design of the perceptor and executive for a long-haul nonlinear fiber optic system. The executive, the preceptor, and the feedback channel are placed in the receiver (Rx), and the executive sends action, such as new data rate by the low data rate link, to the transmitter (Tx) (See section 2.3). The basic principles of CDS are discussed in detail in chapter 1. The perceptor presented in this chapter operates in two modes: (i) the prediction mode and (ii) the BER improvement mode. If the BER is less than the FEC threshold, the CDS extracts a statistical model of the fiber optic channel and saves it in the model library. Under this condition, the CDS operates in the BER improvement mode. If there is a disturbance in the fiber optic channel, the BER could increase and

exceed the FEC threshold. In this case, the CDS cannot extract a model and hence, it switches to the prediction mode in which the preceptor selects one of the models in the model library that is closest to the current condition. Then, the executive, in conjunction with the preceptor, attempts to bring the BER below the threshold and the CDS will switch back to the BER improvement mode. These will be discussed in more detail in Sections 3.2–3.4.

The proposed CDS can improve the BER and/or enhance the data rate based on the intelligent processing of the received data, which includes the extraction of a statistical model of the fiber optic channel or the use of the previously extracted models in the model library. Then, the perceptor sends the BER to the executive through the main feedback channel to caculate the internal reward. The executive produces actions on the fiber optic system/signal to ensure that the BER is continuously under the FEC threshold. The proposed technique based on the CDS concept has low computational cost, is a software-defined technique, and provides a significant improvement in the BER and/or the data rate. For example, in the example calculations done in Section 3.4, we found that the Q-factor improves by 2.74 dB and the data rate is enhanced by 12.5% using the proposed CDS system as compared to the conventional system. Additionally, unlike DBP, the proposed CDS does not require the fiber optic system parameters, such as the number of fiber spans, span length, dispersion, and nonlinear coefficients of each fiber span. Intelligent perception processing by CDS can extract a statistical model of a fiber optic channel. Moreover, the CDS can learn and recognize a disturbance in the optical network, such as a variation in data rate, fiber length or power. Then, the CDS using proper actions such as data-rate tuning and/or constellation size adjustment can automatically tackle network fluctuations.

3.2. Applications of the CDS for the Fiber Optic Link

3.2.1. Perceptor

Figure 3.1 shows the proposed preceptor for fiber optic applications. The purpose of the preceptor is to sense the environment and characterize it. In this example, it makes a statistical model of the fiber optic channel whenever it is feasible. If it is not possible

to make a model using the current data, it selects a model from the model library that is closest to the current state of the channel.

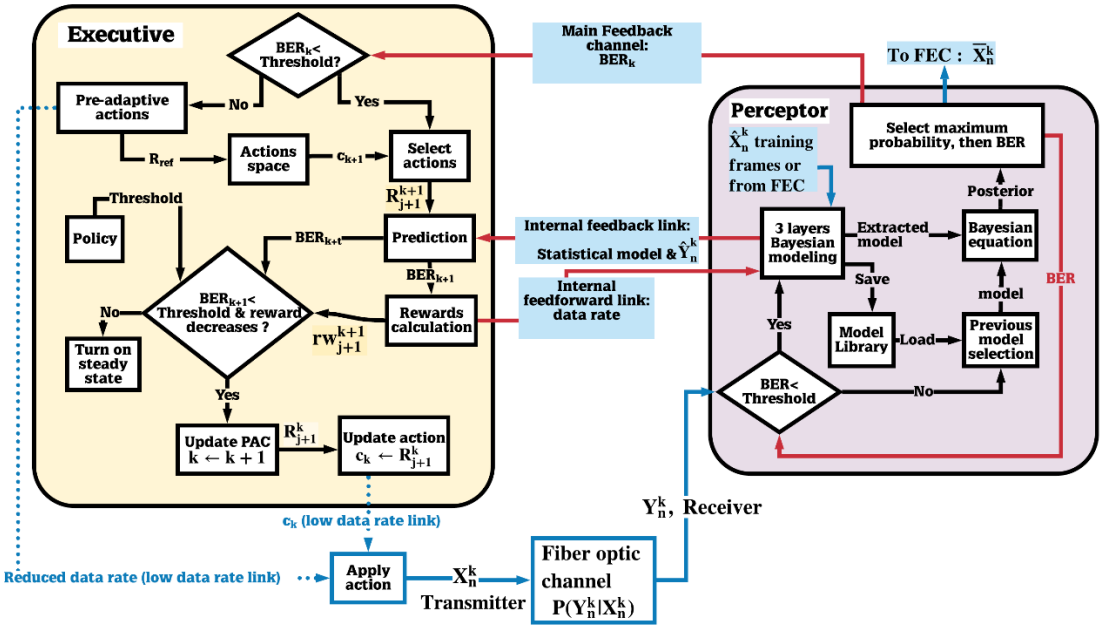


Figure 3.1: The proposed CDS for the long-haul fiber optic link [1.10].

As shown in Figures 2.5 and 3.1 (see section 2.3 also), the input of the perceptor block is the received symbol Y_n^k. The index *k* denotes the cycle number of PAC. The main sub-blocks of the proposed perceptor are: (i) three layered Bayesian generative model, (ii) previous model selection, and (iii) Bayesian equation, which are discussed below.

Three-Layered Bayesian Model

Figure 3.2 shows the three-layered Bayesian modeling for a long-haul fiber optic system inspired by the human brain. Suppose that, initially, there exists no statistical model of the system in a model library. Then, the three-layered Bayesian modeling block extracts the statistical model of the system. We illustrate the modeling process by performing the numerical simulation of the OFDM system. The simulation parameters are presented in Table 3.1.

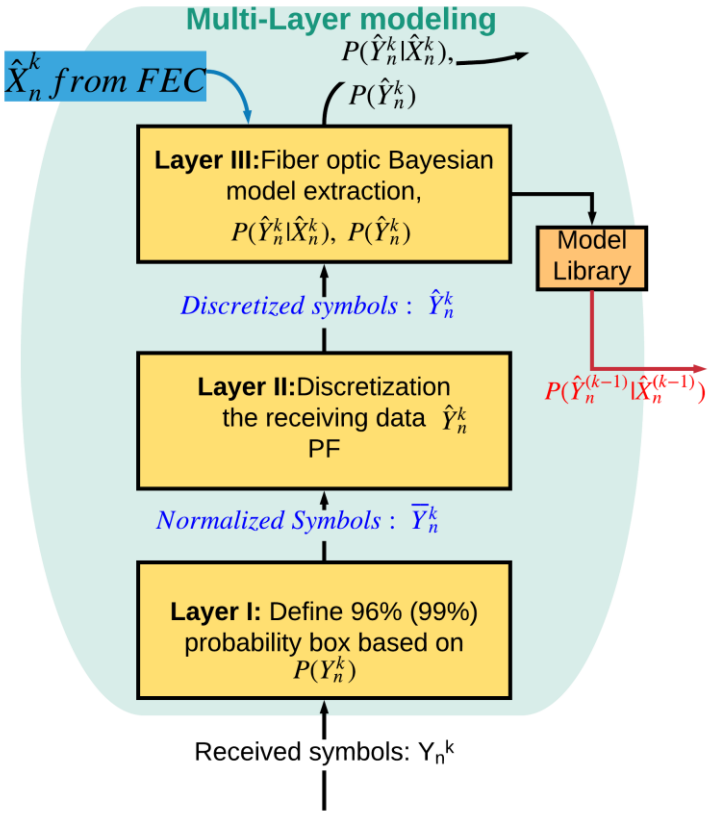


Figure 3.2: Three layered Bayesian modeling inspired by the brain for modeling fiber optic link [1.10].

The signal propagation in an optical fiber is described by the nonlinear Schrodinger equation (NLSE), which can be solved using the standard split-step Fourier scheme [2.3, 2.4]. The output of the fiber optic link passes through a coherent receiver and then the Fast-Fourier transform (FFT) is applied to demultiplex the subcarriers of the OFDM data (See Section 3.4 for more details). The channel estimation and compensation of linear impairments is done using the one tap linear equalizer that uses 16 training frames to mitigate linear distortion [2.4]. The output of the linear equalizer is passed to the preceptor. X_n^k represents the transmitted QAM-16 symbols which takes any one of the values X_t from the following set $\{\pm 3 \pm 3\text{j}, \pm 1 \pm 1\text{j}, \pm 1 \pm 3\text{j}, \pm 3 \pm 1\text{j}\}$, with equal probability at the *k*th cycle and at discrete time *n*, and *t* is the index denoting one of the 16 possible symbols. Y_n^k represents the corresponding received symbols after linear equalization. The three-layered Bayesian modeling consists of three layers, discussed next.

Table 3.1: The numerical simulation parameters of the orthogonal frequency-division multiplexing (OFDM) system [1.10].

Simulation Parameters	Value
Fiber dispersion coefficient (β_2)	$-22.1\ \mathrm{ps}^2/\mathrm{km}$
Fiber nonlinear coefficient (γ)	$1.1\ \mathrm{W}^{-1}\mathrm{km}^{-1}$
Fiber loss coefficient (α)	$0.2\ \mathrm{dB/km}$
Number of fiber spans (N)	20
Span length (L_{span})	80 km
Noise figure (NF)	4.77 dB
Number of OFDM subcarriers per frame	256
Subcarrier modulation	QAM-16
Guard intervals (cyclic prefix)	1.86 ns
Number of data frames	1024
Line width of transmitter laser/LO	22 kHz
Data carrying subcarriers per frame	126
Over sampling factor	2
Number of pilot subcarrier	2
Wavelength (λ)	1550 nm
Length of PRBS	$2^{16}-1$
Fiber type	Standard single mode fiber (SSMF)

Layer I

Figure 3.3 shows the in-phase (I) and quadrature (Q) of the received data Y_n^k. We define 96% probability box (or 99% probability box depending on the application), which means that 96% of the received data is inside this box when QAM-16 data is sent. In addition, the system maps and normalizes all receiving symbols outside of this box to its borders. For example, if the received symbols Y_n^k is $8 + 9j$, then it is mapped to $6+6j$ when the 96% box is used. The normalized data $\bar{Y}_n^k$ (i.e., $6 + 6j$ in this example) is sent to Layer II (Figure 3.2) for further processing. The BER is used to adaptively select the borders of the probabilistic box. For lower computational cost and faster modeling, a probability box with smaller area is better. For example, the computational cost

associated with the 90% probabilistic box is a lower than the 99% probabilistic box. However, this leads to inaccurate results and higher BER. Therefore, the BER is used as a feedback to choose the optimum probability box for which the computational cost is lower with acceptable high accuracy (in this example, it is 96% probabilistic box, see Figure 3.3).

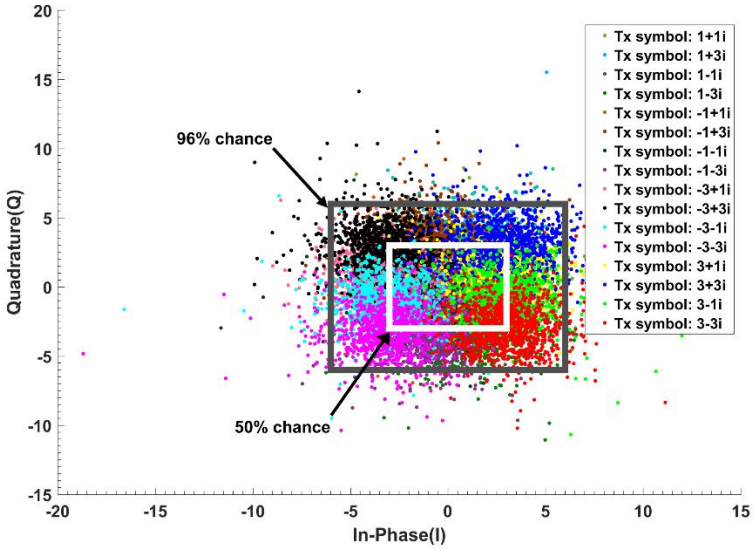


Figure 3.3: Received symbols (Y_n^k) after linear equalization from the simulation results [1.10].

Layer II

In Layer II, the received symbols are discretized with discretization steps Δx and Δy. For simplicity, we have assumed $\Delta x = \Delta y$ and we define precision factor (PF) = $10\Delta x = 10\Delta y$. The normalized received symbols, $\bar{Y}_n^k$ that are inside the shaded box (see Figure 3.4) are mapped to the center of the square as $(x_{l,}, y_m)$, where l is the index for in-phase and m is the index for quadrature. For example, if $x_l = 1$, $y_m = 2$ and $\bar{Y}_n^k = 1.1 + 2.05j$, it is mapped to $\hat{Y}_n^k = 1 + 2j$. The discretized symbols, $\hat{Y}_n^k$ are sent to Layer III for probability estimation. This discretization is inspired by the brain. Our brain monitors the temperature, light, and other environmental conditions and maps this low-level representation to our initial perception, which is a high-level representation such as day or night, cold or warm [3.1]. The low-level representation requires a huge amount of data, which is mapped to an important set of data (initial perception) that takes significantly less memory. This discretization can be more

complicated if we use non-uniform discretization. Figure 3.5 shows the comparison between the CDS system with PF = 0.5, 1, 2, 5 and the conventional system for 52 Gb/s data rate. Here, the conventional system refers to the fiber optic system without CDS. A lower PF means a larger BER improvement and higher model accuracy. However, it leads to higher computational cost**.** The Layer II also receives the BER as top-bottom attention and it can choose the PF based on the BER. Layer II will send the discretized received symbols as $\hat{Y}_n^k$ to Layer III.

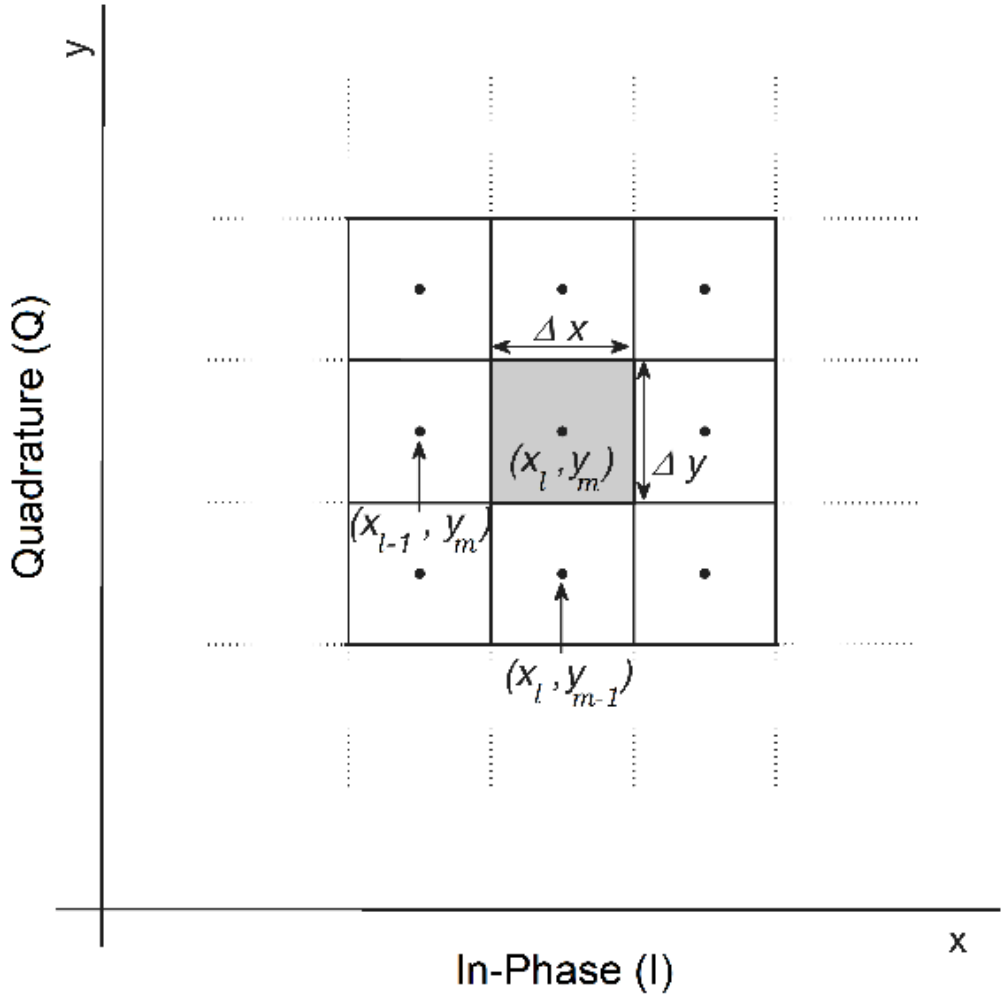


Figure 3.4: Discretization of received symbols [1.10].

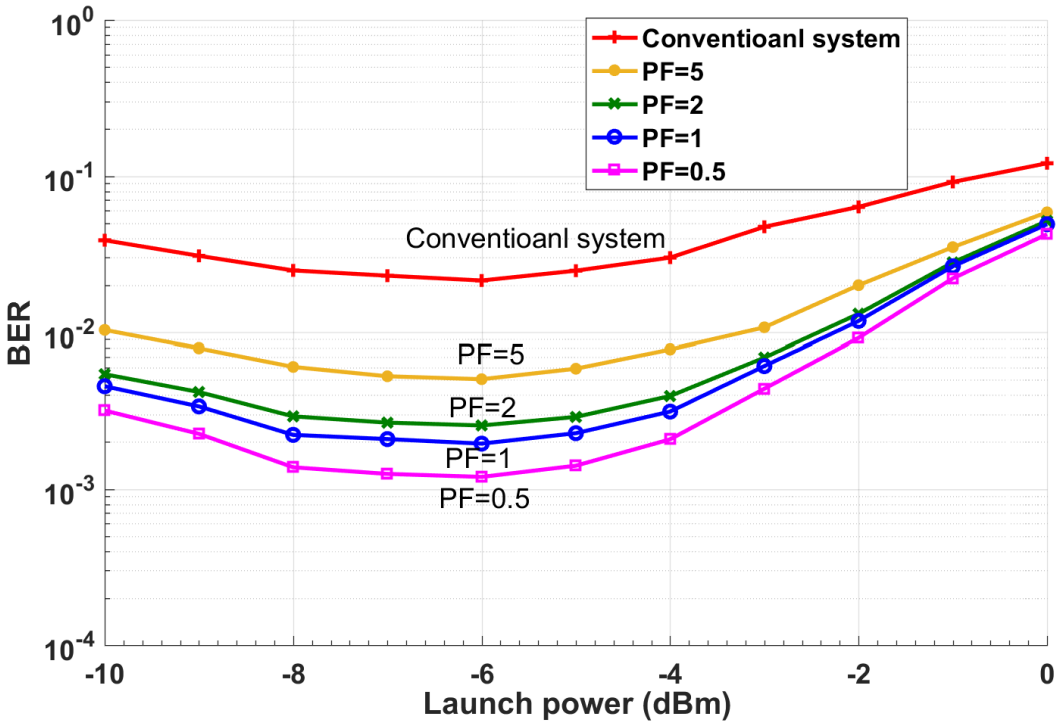


Figure 3.5: The simulation of BER improvement for various precision factor (data rate = 52 Gb/s and transmission distance, L = 1600 km) [1.10].

Layer III

In this layer, the system estimates the probability of receiving Y_n^k for the given transmitted symbol X_n^k, i.e., $P(Y_n^k|X_n^k)$ by approximating it as the probability of receiving $\hat{Y}_n^k$ for the given $\hat{X}_n^k$ (i.e., $P(\hat{Y}_n^k|\hat{X}_n^k)$) using the Monte-Carlo method. The transmitted symbols X_n^k could be known to the receiver by sending training symbol sequences during the statistical model extraction process as $\hat{X}_n^k$ (or might be the post-FEC data usable for training, if post-FEC BER is between 10^{-15}-10^{-9}). We assume that the BER at the FEC threshold is 1.03×10^{-2} [3.2]. The BER after FEC for such a threshold is between 10^{-15} and 10^{-12} [3.2]. We need a certain number of data frames (each frame includes 256 subcarriers) for accurate estimation of $P(\hat{Y}_n^k|\hat{X}_n^k)$. Figure 3.6 shows the BER as a function of the number of received OFDM frames used in the calculation of $P(\hat{Y}_n^k|\hat{X}_n^k)$. As shown in Figure 3.6, the BER converges to 2.7×10^{-3} after 640 OFDM frames for the CDS system when PF = 2. Thus, it can automatically find the sufficient number of OFDM frames to keep the BER under the FEC threshold reliably by the converged model. For example, suppose that the number of transmitted

symbols, $X_n^k = 1-3j$ in a subcarrier is 100,000. Due to channel noise and distortion in the fiber optic link, they could be mapped to any point in Figure 3.3. Next, suppose that the number of received symbols (after normalization and discretization in Layers I and II) with $\hat{Y}_n^k = 2.2-1.5j$ is 10,000 when the transmitted symbol is $X_n^k = 1-3j$. In Layer III, $P\left(\hat{Y}_n^k = 2.2-1.5j \mid \hat{X}_n^k = 1-3j\right) = 0.1$ is calculated. In other words, in Layer III, the number of hits in each of the cells of Figure 3.4 is calculated and the calculated probability $P(\hat{Y}_n^k \mid \hat{X}_n^k)$ is stored in a tensor whose dimension is shown in the second column of Table 3.2. The probability calculation is done for each type of the transmitted symbol. For example, QAM-16 has 16 constellation points or 16 different types of transmitted symbols. In Table 3.2, first, second, and third tensor dimensions correspond to discretized cells for In-phase (I) axis, quadrature (Q) axis (after 2D discretization in Layer II, see Figures 3.3 and 3.4), and number of constellation points of QAM, respectively. For example, if we consider PF = 5 in the first row of Table 3.2, then 33 × 29 × 16 corresponds to 33 columns (I-axis), 29 rows (Q-axis), and QAM-16, respectively. In addition, PF = 5 means the discretization step is 0.5.

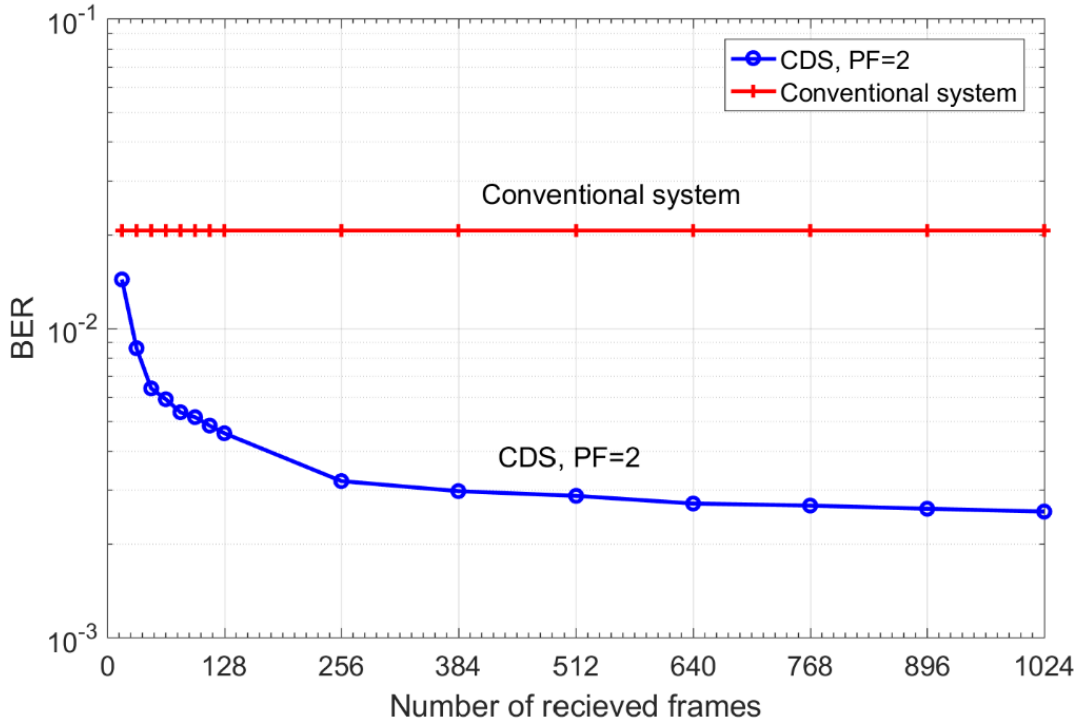


Figure 3.6: Model accuracy improvement versus a number of the received frames from the simulation results. Data rate = 52 Gb/s, P_{tx} = −7 dBm, and L = 1600 km [1.10].

Table 3.2: $P(\hat{Y}_n^k \mid \hat{X}_n^k)$ system model dimension versus precision factor (PF) extracted by simulation (number of OFDM frames = 1024) [1.10].

Precision factor (PF)	$\boldsymbol{P(\hat{Y}_n^k \mid \hat{X}_n^k)}$ System Model Tensor Dimensions
5	$33 \times 29 \times 16$
2	$81 \times 71 \times 16$
1	$161 \times 141 \times 16$
0.5	$321 \times 281 \times 16$

From Table 3.2, we see that the required memory for PF = 0.5 is 16 times more than that for PF = 2. As can be seen from Figure 3.5, the BER improves as PF decreases, but a lower PF implies a larger memory (see Table 3.2). Hence, there is a trade-off between the BER improvement and memory requirements. The extracted model in this layer can be called the statistical model for a fixed fiber network with specific signal and system parameters. This model is saved in the model library for future use. This model is not data dependent, although it has labels for various data rates, fiber lengths, launch power, or other system parameters such as nonlinear coefficient or span length. Unlike the DBP method, the proposed CDS does not need to perform extensive data dependent signal processing if the fiber optic channel does not change appreciably. The extracted statistical model will be used for prediction if there is a disturbance in a fiber optic network, such as a power fluctuation or a change in transmission length due to re-routing. The executive controls the data rate to ensure that the input BER of FEC is always less than the FEC threshold and the data can be transmitted continuously without interruption. For example, when users watch live video on YouTube, the video quality automatically varies (which corresponds to variable data rate) for good quality-of-service (QoS) and minimal interruptions.

Previous Model Selection

If the BER exceeds the FEC threshold, then $P(\hat{Y}_n^k \mid \hat{X}_n^k)$ cannot be extracted by the three-layer modeling in BER improvement mode, immediately. In this case, the system, like a human brain, needs prediction. Thus, it is necessary to use previous experiences for predicting the future. As a result, this subsystem (see Figure 3.1) will choose the closest model to the current system state (state number *k*). Suppose we have selected

the model $P(\hat{Y}_n^{k-1} | \hat{X}_n^{k-1})$. The model $(k-1)$ could correspond to one of the previous data rates, input launch power, previous transmission distance, etc. For example, if the transmission distance is changed from 1600 to 1800 km due to re-routing in fiber optic network, but the data rate, and launch power are fixed at 52 Gb/s, and –6 dBm respectively, then this subsystem will select the closest model to the current system state for prediction (closest in reach, launch power, and data rate). For more accurate prediction, the perturbations to the system should be small, in other words, the changes in transmission reach ΔL $(L_k - L_{k-1})$ and/or the change in data rate ΔR $(R_k - R_{k-1})$ should be small enough (e.g., ΔR = 2 Gb/s when the data rate is 50 Gb/s.) The CDS system will find the closest reach to the current reach and/or closest data rate to the current data from the model library for prediction. We considered *k* to be scalar for simplicity. In general, *k* could be a vector, [*l,m,n,p,* …]. For example, *l, m, n, p* could correspond to data rate, launch power, fiber nonlinear coefficient, and transmission reach, respectively.

Bayesian Equation

The Bayesian equation is another subsystem that is shown in Figure 3.1. If the BER is under the FEC threshold, $P(X_n^k | \hat{Y}_n^k)$ model extracted in Layer III will be used and the CDS operates in the BER improvement mode. In this mode, the estimation is as follows,

$$P(X_n^k | Y_n^k) \cong P(\hat{X}_n^k | \hat{Y}_n^k) = \frac{P(\hat{Y}_n^k | \hat{X}_n^k) P(\hat{X}_n^k)}{P(\hat{Y}_n^k)}, \qquad (3.1)$$
$$b_k(\hat{X}_n^k, \hat{Y}_n^k) = P(\hat{X}_n^k | \hat{Y}_n^k),$$

Here, the index *k* is PAC number in CDS. $b_k(\cdot)$ is the extracted posterior at the *k*th PAC. If the BER exceeds the FEC threshold, the CDS will use the model $b_{k-1}(\hat{X}_n^k, \hat{Y}_n^k)$ stored in the model library to approximate $b_k(\hat{X}_n^k, \hat{Y}_n^k)$ as predictive Bayesian model:

$$b_k(\hat{X}_n^k, \hat{Y}_n^k) \cong b_{k-1}(\hat{X}_n^k, \hat{Y}_n^k). \qquad (3.2)$$

The model $b_{k-1}(\hat{X}_n^k, \hat{Y}_n^k)$ is the closest model to the current system state. Hence, the CDS can guarantee maximum possible data rates using the prediction based on previous models and improve the BER. In statistical processing using the Bayesian

approach, it is known that the evidence $P(\hat{Y}_n^k)$ can be considered as the scaling factor. We accomplish the task of model extraction by cycling through the prediction mode and BER improvement mode, as will be explained in Section 3.4.

Selecting Maximum Probability and BER Calculation

The CDS can save the posterior probability $P(X_n^k | Y_n^k)$ in the perceptor library after one-time calculation. The symbol that has the maximum probability to be transmitted is selected using

$$\bar{X}_n^k = \arg\max_{t=1,2,\ldots,M} \{P(X_n^k = X_t \mid \hat{Y}_n^k)\} \tag{3.3}$$

where, $t = 1, 2, \ldots, M$, and M is the constellation size. For example, for QAM-16, M is 16, and $P(X_n^k = X_t \mid \hat{Y}_n^k)$ represents the probability that the transmitted symbol X_n^k is X_t which is one of the values from the symbol alphabet $\{\pm 3 \pm 3\text{j}, \pm 1 \pm 1\text{j}, \pm 1 \pm 3\text{j}, \pm 3 \pm 1\text{j}\}$ for the given $\hat{Y}_n^k$. The symbol $\bar{X}_n^k$ that has the highest chance to be transmitted for the given $\hat{Y}_n^k$ is selected. Comparing $\bar{X}_n^k$ and $\hat{X}_n^k$, the BER is computed by error counting and the BER is sent to executive. As top-bottom attention, this sub-system also sends the BER to the other subsystems in the perceptor. If the BER exceeds the FEC threshold, the FEC output will not converge and the post-FEC BER will be high and, hence, the system would infer that the pre-FEC BER has exceeded the FEC threshold. Therefore, the executive will take some actions such as decreasing the data rate. In addition, the local cycles in the preceptor will examine a different type of model for prediction and try to decrease the BER as much as possible.

3.2.2. Main Feedback, Internal Feedback, and Feedforward Channels

As shown in Figure 3.1, the main feedback and the internal feedback channels were used to exchange the information between perceptor and executive. The perceptor sends the BER for calculating the internal reward to the executive through the main feedback channel. In addition, the perceptor sends the probabilistic model ($P(\hat{Y}_n^k | \hat{X}_n^k)$), evidence ($P(\hat{Y}_n^k)$) and current perceptor mode to the executive through the internal feedback channel. The executive asks the transmitter to send information to the

perceptor through feedforward channel such as fiber launch power P_{tx}, data rate, and $P(X_n^k)$ (if probabilistic shaping is used).

3.2.3. The Executive

The executive is placed in the receiver (see Figures 2.5, 2.7 and 3.1), but it can send the action to the transmitter, such as new data rate by low data rate link. Based on the BER sent through the main feedback channel as raw reward, together with the probabilistic model ($P(\hat{Y}_n^k|\hat{X}_n^k)$) and evidence ($P(\hat{Y}_n^k)$) sent through the internal feedback, the executive may decide to increase or decrease the data rate. It could adjust the launch power and data-rate. The proposed executive in this chapter can do basic actions based on "actions space" block in Figure 3.1, which are discussed below.

These actions include adjusting launch power and data rates in fixed or variable discrete steps. The executive will decide what actions to take, based on the BER sent through the main feedback channel. When there is a disturbance in a fiber optics communication system such as power fluctuation or reach change, and if there is no proper model in the perceptor library that is closest to the perturbed system, then the executive will become aware of it through the BER sent through internal feedback channel (in this case, the BER will be high). The executive will take basic actions such as lowering the data rate so that the BER is below the FEC threshold. When the desired goals are met, the executive would save the corresponding actions in the actions library and will load them for the similar situations in future.

Policy

The policy determines the goals that the CDS should achieve. The first goal is that the BER should be under the FEC threshold. The second one is that the computational cost should be lower than a pre-defined cost. An acceptable computational cost. The storage and/or computational cost depends on PF and ΔBER (Figure 3.6). We define ΔBER as the difference in BER between current received frames and previous received frames with a step of 128 frames. For example, we can set our policy that the desired ΔBER should be less than 5×10^{-5}. From Figure 3.6, we find that if the current received frame is 768 and the previous received frame is 640, then $\Delta BER = 4 \times 10^{-5}$, which is less

than the desired ΔBER. Therefore, the CDS stops the model extraction process after receiving 768 frames and saves the model and the posterior probability in the CDS model library.

The CDS with a Simple executive

In this section, we present the CDS with a simplified executive that does only basic actions. The simple executive cannot predict the outcome of the actions before applying actions on the environment and after applying action on environment could find out the outcome of action by received internal rewards from perceptor. Figure 3.1 shows the block diagram of the proposed executive for the long-haul fiber optic link. In Figure 3.1, the executive receives the BER_k through the main feedback channel at the kth cycle. We could define the internal reward as a function of the BER_k only. However, in general, as the data rate increases, the BER increases too, and higher data rates are desirable while the BER is relatively low, (i.e., under the FEC threshold). Hence, a better definition of internal reward should be a function of both BER and the current data rate. Therefore, we define internal reward (cost-to go function) as follows:

$$rw^k = \frac{BER_k}{\left(R_j^k - R_{ref}\right)^4} \tag{3.4}$$

$$R_j^k = R_{ref}(Gb/s) + j \times d(Gb/s) \tag{3.5}$$

where the BER_k is the BER at kth PAC, d is the discretization step for changing the data rate which is set equal to 2 Gb/s in this chapter, and j is equal to k before a disturbance to the fiber optic system and it is reset to 1 after the disturbance and pre-adaptive actions. The pre-adaptive actions refer to those actions that reduce the data rates until the BER is under the FEC threshold. R_{ref} is a convenient reference data rate that is set equal to $R^1 - d$. The executive updates the R_{ref} after any disturbance using pre-adaptive actions. The internal reward rw^k and data rate R_j^k are further explained in Section 3.4.1 with an example.

The action space for this specific case study can be defined for the fiber optic link as follows:

$$C = \{c_k \mid data\ rate\ R_j^k > R_{ref}\} \tag{3.6}$$

where C is set of all possible actions in the actions space, and c_k is the action at kth PAC such that the data rate R_j^k exceeds the reference data rate R_{ref}.

Suppose that the current action is c_k .The executive can calculate the outcome of the action c_{k+1}, after applying it on the fiber optic channel using the following equations:

$$P(\hat{X}_n^{(k+1)} \mid \hat{Y}_n^{(k+1)}) \cong b_k(\hat{X}_n^k, \hat{Y}_n^k), \tag{3.7}$$

$$\bar{X}_n^{(k+1)} = \arg\max_{t=1,2,\ldots,M}\{P(X_n^{(k+1)} = X_t \mid \hat{Y}_n^{(k+1)})\}, \tag{3.8}$$

where $X_n^{(k+1)}$ represents the input symbols at (k+1)th PAC at the data rate R_{j+1}^{k+1}. The BER is computed by comparing $\bar{X}_n^{(k+1)}$ and $X_n^{(k+1)}$. In equation 3.7, the previous model $b_k(\hat{X}_n^{k+1}, \hat{Y}_n^{k+1})$ is used to calculate the posteriori. So, if the BER using previous model is under FEC threshold and rw^k is decreased (see eq. 3.4), then CDS keep the c_{k+1} as the new action for the CDS. Note when the BER exceeds the FEC threshold, the preceptor switches to prediction mode and selects the previous model closest to the current situation. When the BER is under the FEC threshold using the previous model, the executive enhances the data rate to R_{j+1}^{k+1} as new fiber optic communications system data-rate. However, when the BER is more than FEC-threshold using the previous model, R_j^k will be used by the executive and the steady-state mode will turn on.

The subsystems of the executive shown in Figure 3.1 are explained as follows. If the BER exceeds the FEC threshold, the executive runs pre-adaptive actions such as data rate reduction and sending the updated reference data rate R_{ref} to the actions space. If the BER is less than the FEC threshold, the executive selects an action from the actions space. Here, the action is the data rate enhancement. The block “prediction” in the executive predicts the BER_{k+1} based on the statistical model received from the

perceptor through the internal feedback link (see eq. (3.7) and eq. (3.8)). The block "rewards calculation" computes the internal reward using the predicted BER and a prospective data rate of R_{j+1}^{k+1} and eq. (3.4). If the predicted BER is below the FEC threshold and a lower value of the internal reward is achieved, then the executive interprets that the system is performing well with R_{j+1}^{k+1}, and the data rate can be enhanced further. The new (prospective) data rate is communicated to the transmitter using a low data rate link. Otherwise, the executive turns on the steady state mode, and assumes R_j^k is the best possible data rate that can be achieved for the current fiber optic link.

3.3. The Algorithm of CDS with a Simplified Executive

In this section, we describe the proposed algorithm for the simplified CDS applied on a fiber optic link. Algorithm 3.1 shows the outline of the complete one PAC procedure of a simplified CDS. In addition, Table 3.3 lists all the notations used in this thesis.

Table 3.3: The important notation list used in this chapter [1.10].

Notations	Value
K	The total number of environmental perception action cycles (PACs)
k	The current PAC number during PAC running
k'	The PAC number before the current CDS steady state mode
C	The set of all possible actions in actions library
X	The set of all possible modulation state
X_n^k	The current transmitted symbols, at time n
$\widehat{X}_n^k$	The known data using for modeling
$\overline{X}_n^k$	The estimated decision by the system
Y_n^k	The received symbols at time n and kth PAC
$\overline{Y}_n^k$	The normalized received symbols at time n and kth PAC
$\widehat{Y}_n^k$	The discretized received symbols at time n and kth PAC

rw^k	The internal reward (cost-to-go function) at time *n* and kth PAC
$P(\hat{X}_n^k \mid \hat{Y}_n^k)$	The estimated posterior at time *n* and kth PAC
c_k	Current action applied on fiber optic link
$FEC-threshold$	The predefined threshold
$b_{k-1}(\hat{X}_n^k, \hat{Y}_n^k)$	The prediction using model at *k*th PAC
rw^{k+1}	The internal reward for (*k*+1)th PAC
c_{k+1}	The final action for applying in (*k*+1)th PAC

The algorithm is described briefly as follows:

1. Prediction of rewards using available models or reduction of the data rate until the BER is less than the FEC-threshold.
2. Extract the exact model of the system and minimize the BER.
3. Use this model to predict the BER and internal rewards before increasing the data rate.
4. If prediction shows lower internal reward and the BER is less than FEC-threshold, then the CDS increases the data rate.

Algorithm 3.1: Environmental actions through the global feedback

Input: the observables Y_n^k

Output: c_{k+1} as the final action in environmental actions mode

Initialization:
Pre-adaptive actions:
Reduce data rate till BER<FEC-threshold
Make steady state mode off, prediction mode on
Probability box% = 96, threshold = 0.01, $PF = 2$, j = 1
$c_{k'+1} \leftarrow$ *first action from C*

Apply $c_{k'+1}$ to the environment (fiber optic channel)

1: **for** $k = (k'+1)\,to\,K$ (see Table 3.3)
2: Take the observable Y_n^k
3: **if** the model is not available **then**
4: **if** BER > FEC-threshold **then**
5: Estimate $P(\hat{X}_n^k \mid \hat{Y}_n^k)$ using $b_{k-1}(\hat{X}_n^k, \hat{Y}_n^k)$

6: Estimate $\bar{X}_n^k$ by decision making
7: **else**
8: Extract the $P(\hat{Y}_n^k \mid \hat{X}_n^k)$ and $P(\hat{Y}_n^k)$
9: Calculate the posterior $P(\hat{X}_n^k \mid \hat{Y}_n^k)$
10: Estimate $\bar{X}_n^k$ by decision making
11: **end if**
12: **else if** model is available **then**
13: Load model, evidence and posterior from preceptor library
14: Estimate $\bar{X}_n^k$ by decision making
15: **end if**
16: Send *PF1*, $b_k(\hat{Y}_n^k \mid \hat{X}_n^k)$ and $P(\hat{Y}_n^k)$ to the executive
Internal reward
17: Calculate BER_k, *and* rw_j^k and send it to executive
Planning
18: Localize the set of all closest action to c_k
Learning
19: Apply c_{k+1} ($c_{k+1} \in C$)
20: Estimate $b_k(\hat{X}_n^{k+1}, \hat{Y}_n^{k+1})$
21: Predict $\bar{X}_n^{(k+1)}$
22: Calculate BER_{k+1} *and* rw_{j+1}^{k+1}
23: **if** $BER^{(k+1)} \le threshold$ and $rw^{k+1} \le rw^k$ **then**
24: apply c_{k+1} to the fiber link
25: $j \leftarrow j+1$
26: $k' \leftarrow k+1$
27: **else**
28: Turn steady state on (Stay on c_k)
29: **end if**
30: **end for**

3.4. The CDS Case Study for the OFDM-Based Long-Haul Standard Single-Mode Fiber System

To illustrate how CDS can control QoS (i.e., increase the data rate/improve the BER), we implemented the CDS with the simplified executive for the OFDM-based long-haul standard single mode fiber link. This is a proof-of-concept case study for the presented Algorithm 3.1, whose validity to other scenarios will be further studied. The fiber optic

system simulation parameters are the same as Table 3.1, and the parameters of the CDS are as follows: precision factor, PF = 2 and 96% probability box is used. All the figures shown in this chapter are obtained by simulating the fiber optic system (with or without CDS) using the parameters listed in Table 3.1.

The BER versus launch power of a conventional fiber optic communications system (i.e., without CDS) is shown in Figure 3.7. The BER is calculated based on the geometric boundaries between the symbols for the conventional system. At low launch powers, the BER decreases with the launch power since the SNR improves. However, at higher launch power (greater than −4 dBm), the BER increases with launch power due to nonlinear impairments. If we consider 1.03×10^{-2} (with 14.3% overhead) as the FEC threshold [3.3], we can see from Figure 3.7 that the maximum achievable data rate is 48 Gb/s. The adaptive linear equalizer (See Figure 2.6 and 2.7) mitigates the linear distortion, and hence, the nonlinear distortion is dominant for data rates greater than 48 Gb/s. The CDS now extracts the model for $k = 1$ (which corresponds to the data rate of 48 Gb/s) in the preceptor while operating in the BER improvement mode (see Figure 3.8) and saves the posterior for 48 Gb/s as $b_1(\hat{X}_n^k, \hat{Y}_n^k)$ in the model library. It may be noted that the BER for the case of 48 Gb/s with CDS (Figure 3.8, $k = 1$) is lower than the BER of the conventional system at the same data rate (Figure 3.7) because of the Bayesian approach (see Section 3.2). The CDS uses 2 Gb/s discretization step for data rate and a step of 1 dB for the launch power. In the next cycle ($k = 2$), the executive applies the data rate to 50 Gb/s (see eqs. (3.4) and (3.5), Algorithm 3.1). Since there is no model in the model library for the case of the data rate of 50 Gb/s, the perceptor will switch to the prediction mode also, and predicts $b_2(\hat{X}_n^2, \hat{Y}_n^2)$ based on the posterior for 48 Gb/s (i.e., $b_1(\hat{X}_n^2, \hat{Y}_n^2)$) and the BER is computed. It is found that the BER is under the FEC threshold (See Figure 3.9, $k = 2$). As can be seen in Figure 3.9, the predicted BER is lower than the FEC threshold and $rw^2 \leq rw^1$ (see Fig. 3.11) the action of enhancing the data rate is communicated to the transmitter using a low data rate link and transmitter enhances the data rate. Since the BER is now below the FEC threshold, the preceptor will operate in BER improvement mode and it will extract the posterior for $k = 2$, $b_2(\hat{X}_n^2, \hat{Y}_n^2)$). Therefore, the BER can be improved further (Figure 3.9), for

example, if we compare the BERs in Figures 3.8 and 3.9, when k = 2, we find that the BER is lower in the BER improvement mode (See Figure 3.8) as compared to that in the prediction by the perceptor (see Figure 3.9). Since the objective is to enhance the data rate (with BER below the FEC threshold), in the next cycle (k = 3), the executive will apply the enhanced data rate of 52 Gb/s and the predicted BER is shown in Figure 3.9 (k = 3) and this process continues. This process of cycling through the BER improvement mode and prediction mode will continue and the executive will increase the data rate up to 56 Gb/s (k = 5 in Figure 3.9). At this point, the executive finds that the predictive model cannot bring the BER under the FEC threshold (see Figure 3.9). Therefore, it will decrease the data rate and the steady-state data rate will be 54 Gb/s. The system will continue to operate at 54 Gb/s until there is a disruption to the system. When the disturbance arises, the BER increases and the executive will decrease the data rate, and the perceptor will operate in the prediction mode and choose the closest posterior from the model library to predict the transmitted data.

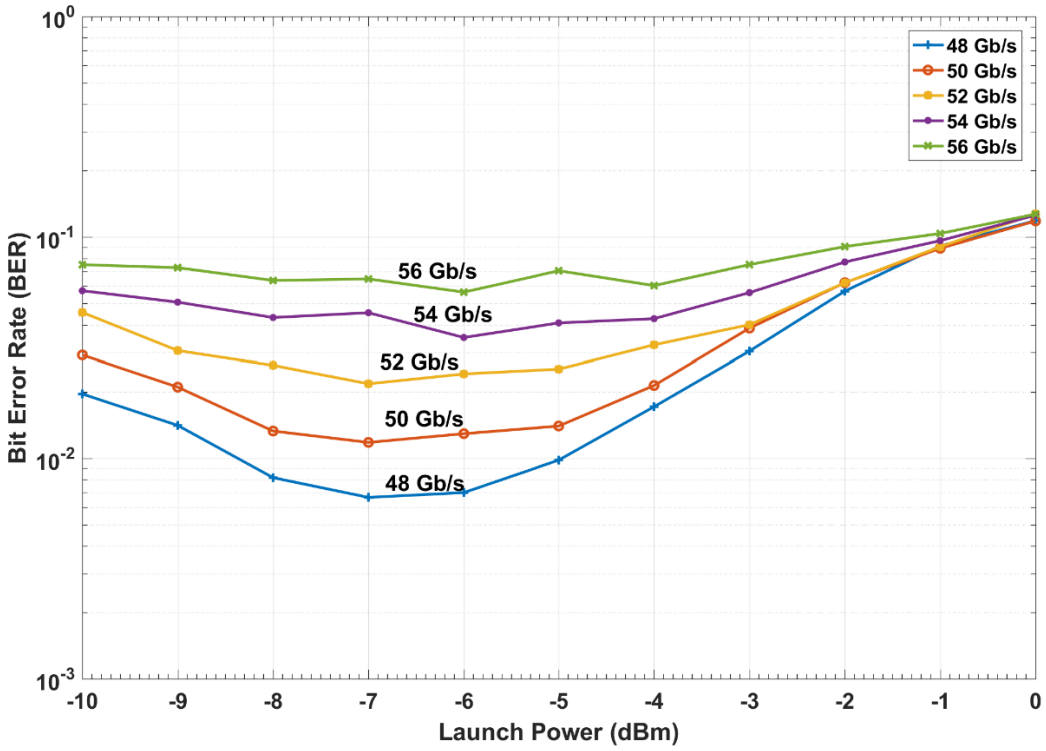


Figure 3.7: The simulation of a conventional system with a linear equalizer [1.10].

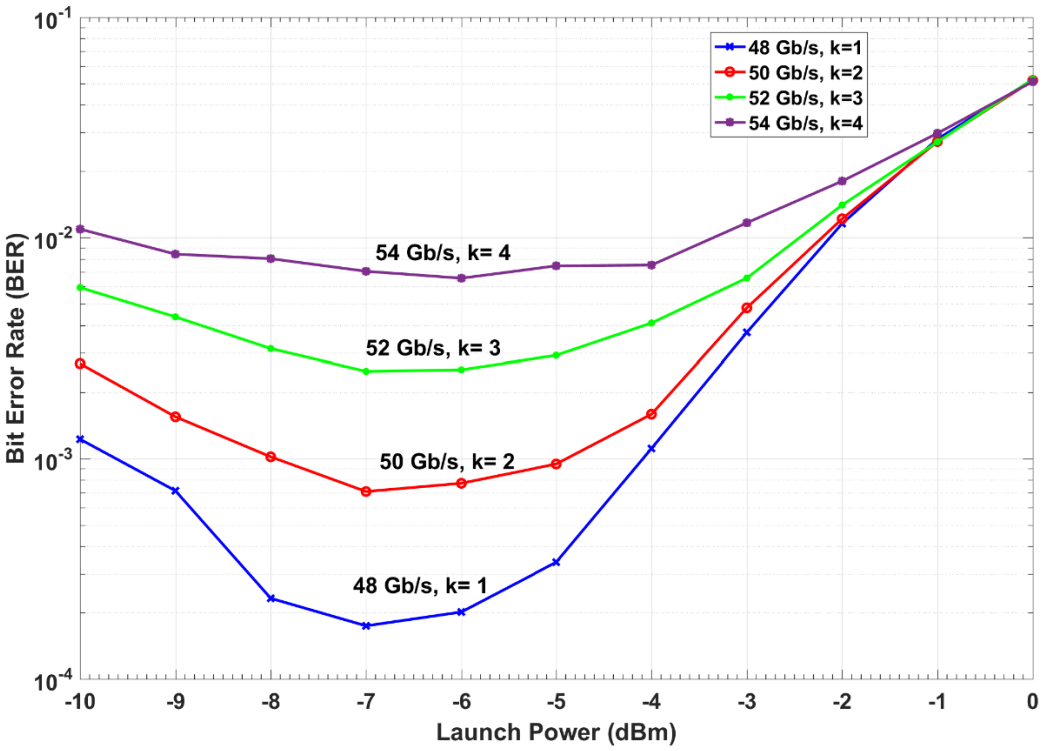


Figure 3.8: The simulation of a fiber optic system with CDS with the known model (BER improvement mode) [1.10].

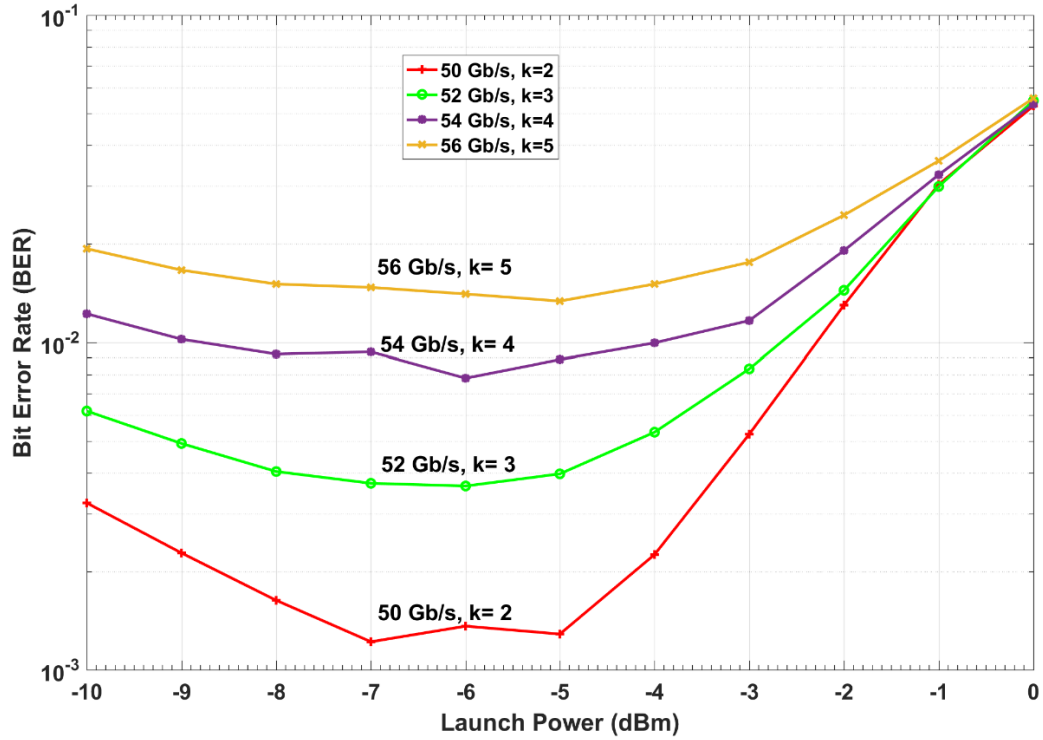


Figure 3.9: The simulation results for the prospective BER predicted in executive (CDS in prediction mode) [1.10].

For example, if the gain of an inline amplifier of a fiber optic link drops from 18 to 10 dB, the BER would increase. The BER is sent from the preceptor to the executive that takes action to lower the data rate. The CDS cycles through the prediction mode and BER improvement mode (see Figures 3.8 and 3.9) and finally settles down at a data rate that is achievable for the current state of the channel without interruption of the service. The CDS monitors the BER continuously, and if the amplifier is repaired, then the CDS automatically will retrieve the original data rate. Additionally, if there is a cut in a fiber-optic network, the data will be re-routed over a fiber optic link that may have different characteristics such as fiber length, dispersion, and nonlinear coefficient. The CDS senses this change through the BER fluctuations and again adapts itself to the new environment.

Now, we show the improvement of the BER for the OFDM fiber optic link upgraded with the CDS. It is customary to project the Q-factor based on BER and Gaussian noise statistics as [2.4]:

$$Q = 20log10(\sqrt{2}erfcinv(2.BER)) \tag{3.9}$$

In Figure 18, the Q-factors of the conventional system, CDS in BER improvement mode, and CDS in prediction mode for the data rate of 54 Gb/s (k = 4) as the function of the launch power are compared. As can be seen, the Q-factor improves by 1.75 and 2.74 dB when the CDS operates in prediction mode and BER improvement mode, respectively. The improvement in BER using the CDS principle is due to the Bayesian approach, while the data rate enhancement is due to the cycling through the prediction mode and BER improvement mode.

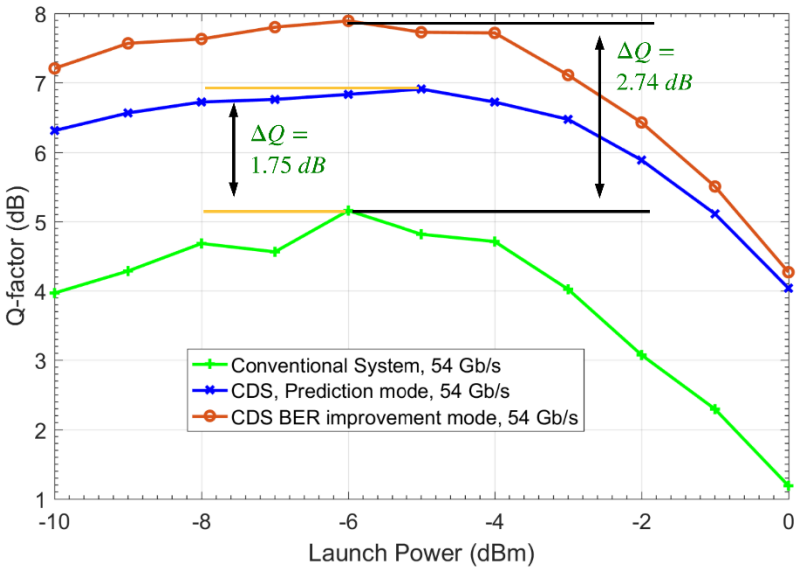


Figure 3.10: The simulation results for the quality-factor versus launch power for the conventional system, CDS prediction mode, and CDS improvement mode [1.10].

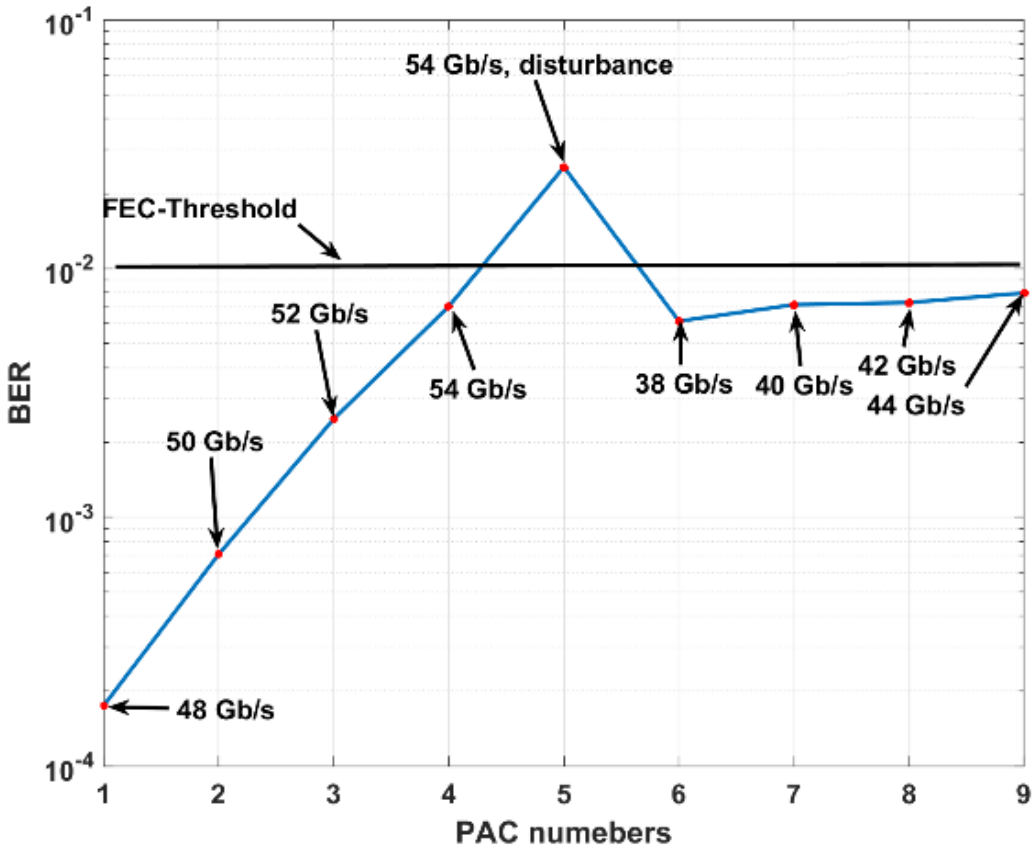
From Figure 3.10, it may be noted that the optimum launch power does not change for the case of CDS in BER improvement mode as compared to the conventional system, although it increases slightly in the prediction mode (blue curve). However, if DBP is used in a conventional system, then the optimum launch power increases. This is because the DBP compensates for nonlinear impairments to some extent. The simplified CDS in this example should not be considered as a substitute for a nonlinear compensation scheme such as DBP. The difference between the DBP or any nonlinear compensation scheme and the proposed approach is that the DBP makes use of the deterministic channel model (such as nonlinear Schrodinger equation) whereas the proposed approach utilizes the statistical channel model. The Q-factor gain shown in Figure 3.10 using the simplified CDS is due to the information extracted by the perceptor that helps the CDS to approximate the optimal decision-making boundaries even if the symbol likelihood is non-Gaussian and asymmetric. Therefore, the final data rate is fixed at 54 Gb/s. This Q factor improvement makes the system less sensitive to any small disturbance, and the system will have better QoS with less interruptions.

Using the proposed CDS, the BER can be improved further by operating the preceptor in the BER improvement mode with a lower PF (see Figure 3.5). However,

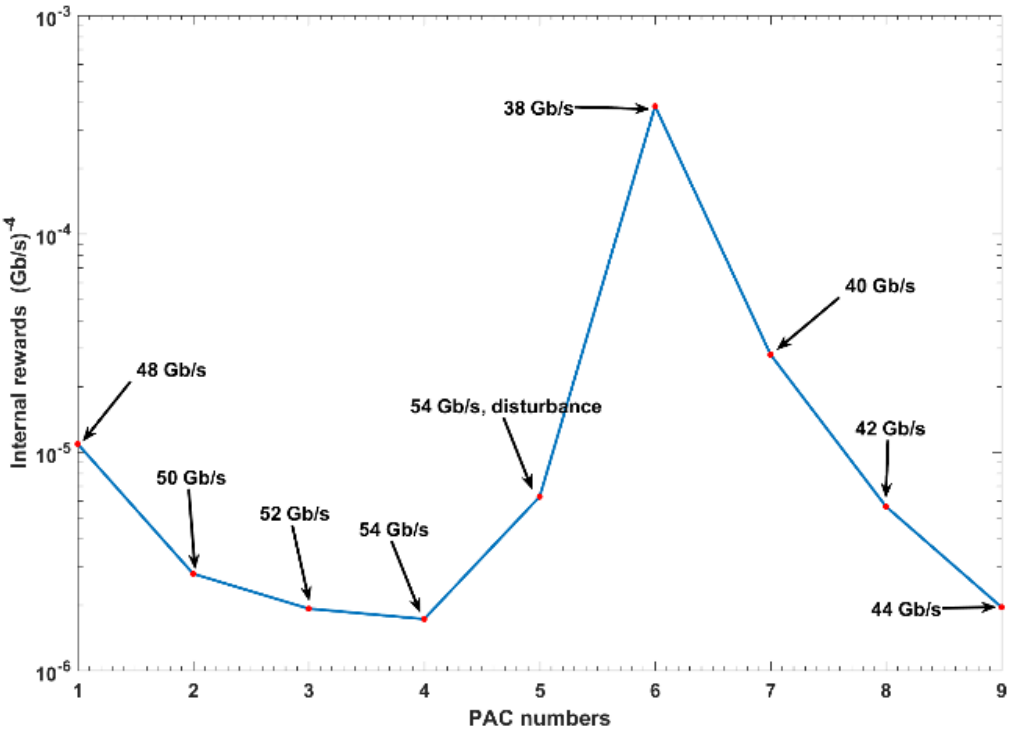
this is not implemented in the example described here. We have provided an example of how a CDS with a simple executive CDS can still improve the BER, increase the data rate and handle changes in fiber optic channel with less interruption. From Figures 3.8–10, it may be noted that as the signal power increases, the BER gets larger due to fiber nonlinear impairments.

3.4.1. Simulation Results in the Presence of Disturbance

If the fiber optic system is disturbed, then the CDS can sense it and adapt to the new environment. We simulate the CDS for the disturbed system by assuming that the tenth inline amplifier of the fiber optic link is partially damaged. The noise figure (NF) of this amplifier is increased from 4.77 to 23 dB, while the rest of the inline amplifiers have the normal NF of 4.77 dB. As a result, the received signal becomes noisier. The simulation results using the CDS in the presence of disturbance are shown in Figure 19.



(a)



(b)

Figure 3.11: The simulation results for the cognitive dynamic system (CDS) in presence of disturbance (a) bit error rate (BER) vs perception action cycle (PAC) numbers, (b) CDS learning curve [1.10].

Figure 3.11a shows that in the PAC numbers 1 to 4, the BER is under the FEC-threshold (see Figures 3.7–3.10), and the CDS achieves a data rate of 54 Gb/s at fourth PAC. Hence, the steady state data rate is 54 Gb/s, and the CDS operates in s steady state mode. At the fifth PAC, the system is disturbed because of the partial damage to the tenth amplifier. The current statistical model for the system operating at 54 Gb/s cannot bring the BER under FEC-threshold. As a result, the CDS turns the steady state mode off and activates the pre-adaptive actions (see Figure 3.1). Therefore, the CDS lowers the data rate with a step of 2 Gb/s until the BER is under the FEC-threshold. At $k = 6$, j is reset to 1, data rate is 38 Gb/s, and the BER is under the FEC-threshold (see Figure 3.11a). Since $R_1^6 = 38$ Gb/s, R_{ref} is updated to $R_1^6 - d = 36$ Gb/s. The procedure for PAC numbers 6 to 9 is similar to that for PAC numbers 1 to 4 (see Algorithm 3.1). Additionally, the rewards are calculated using eq. (3.4), and Figure 3.11b shows the

internal reward rw^k as a function of the PAC number *k* calculated. Before disturbance (i.e., $k < 5$) we see that the internal reward decreases as *k* increases. This is because the weight of the data-rate is more than the weight of the BER (see eq. (3.4)). The executive would increase the data rate if the internal reward is decreasing (provided that the BER is less than the FEC-threshold). At $k = 5$, there is a disturbance leading to an increase in the BER and hence, internal reward also increases. The internal reward reaches its peak at $k = 6$ since the difference between R_1^6 and R_{ref} is the lowest (see eq. (4)). Thereafter, the internal reward decreases at $k = 7$, the CDS applies the same procedure used at $k = 2$. At ninth PAC number ($k = 9$), the new steady-state data rate in the presence of disturbance is the 44 Gb/s (similar to fourth PAC number). If the amplifier is repaired in future, the CDS follows the same procedure as in PAC numbers 1 to 4 to increase the data rate again.

3.4.2. CDS Complexity

In order to calculate the computational cost, we distinguish three types of CDS modes: (i) BER improvement mode, (ii) prediction mode, and (iii) steady state. The complexity of a CDS varies depending on the perceptor mode. When the CDS operates in the BER improvement mode, the BER converges after *N* OFDM frames (in Figure 3.6, it is about $N = 512$) and each frame has *K* data-carrying subcarriers (From Table 3.1, it is 126). To create the histogram (i.e., to calculate $P(Y_n^k \mid X_n^k)$) based on the discretized data, we need $N \times K$ simple additions (just counting the samples within the tiny surface). To calculate $P(X_n^k \mid Y_n^k)$, we need *M* real multiplications where *M* is the order of QAM (in our example, $M = 16$). The storage requirement depends on the precision factor (see Table 3.2). For example, with a PF of 2, we need to store a matrix of dimension $81 \times 71 \times 16$. In addition, we need to store the matrix corresponding to $P(X_n^k \mid Y_n^k)$ which has a dimension of $16 \times 81 \times 71$. When the CDS operates in the prediction mode, there is no cost associated with model extraction (i.e., to calculate $P(Y_n^k \mid X_n^k)$), but we need *M* real multiplications to calculate $P(X_n^k \mid Y_n^k)$. In this mode, we need to store the matrix corresponding to $P(X_n^k \mid Y_n^k)$, which has a dimension of $16 \times 81 \times 71$. In steady state, the

CDS uses the saved $P(X_n^k \mid Y_n^k)$, and makes a decision symbol by symbol (see eq. (3)). These matrices can be stored in hard disk and can be loaded to RAM when needed. No multiplication or addition needs to be done in steady state.

The computational cost associated with DBP is as follows. In each propagation step, $J\log_2(J)$ complex multiplications are required where *J* is the number of samples in time domain. If there are *B* propagation steps, the complex multiplication is $O(B\times J\log_2 J)$ [2.14][2.15]. For example, for a 20-span system, assuming 10 steps per span and with *J* = 16,384, we need roughly 45,875,200 complex multiplications! Besides, this computation needs to be done continuously for each block of data. Three complex vectors of length *J* (about the total size of 98,304) needs to be stored. In contrast, most of the computational complexity of the CDS is present only when there is a fluctuation in the fiber optic channel; under steady state, the computational cost is minimal. It is useful to compare the complexity of the proposed method in this chapter with the digital backpropagation (DBP) for Q-factor enhancement using the simulation parameters shown in Table 3.1. We run both algorithms on a Microsoft Surface-Pro with Intel® Core™ i706650U CPU @ 2.20GHz 2.21 GHz, 16 GB RAM, system type 64-bit Operating System x64-Based processor using MATLAB. For the DBP algorithm [2.15][3.4], the runtime is 1225.4 s (~20 min and 25 s). However, the runtime for the proposed CDS is 14.03 s for PF = *2*. In addition, the CDS runtime reduces to 7.71 s for PF = *2* in steady state mode. However, the DBP running time always remain similar for any continuous data stream. In addition, the reported Q-factor improvement for DBP in OFDM systems is between 1 and 2.2 dB [1.40], while it is 2.74 dB (see Figure 3.10) for the fiber optic link upgraded with the CDS. Additionally, the main features of the proposed CDS are the following:

1. The CDS perceptor can extract the model (likelihood) dynamically and does not need the transmitter to send training sequences.
2. We assume that the fiber optic link parameters are unknown in an optical network.
3. The CDS can be applied to any type of fiber optic system such as single carrier, wavelength division multiplexing (WDM), single channel OFDM, or a WDM system with individual channels carrying the OFDM signal.

4. In our approach, we do not use training sequences from the transmitter to extract the channel model $P(Y_n \mid X_n)$ since it would disrupt the service. It can improve the BER by finding the optimum decision boundaries.

3.5. Conclusions

The cognitive dynamic system (CDS) is inspired by the human brain and can be regarded as the brain-like intelligence. The principles of CDS are applied to a nonlinear fiber optic communication system for the bit error rate (BER) improvement and the data rate enhancement. The executive and preceptor of CDS are implemented in the optical receiver, but executive sends actions to the transmitter such as new data rate. The block diagram of the CDS consisting of the preceptor, the main feedback channel, and executive is presented. The preceptor operates in two modes: prediction mode and BER improvement mode. When there is a fluctuation in the fiber optic network parameters, such as reach or amplifier gain, the BER may exceed the forward error correction (FEC) threshold. In this case, the preceptor operates in prediction mode and, working closely with the executive, it lowers the BER to meet FEC threshold. Once the BER comes below the FEC threshold, the perceptor operates in the BER improvement mode. In this mode, it extracts a statistical model for the fiber optic channel for the current fiber optic link parameters and saves it in the model library for future use. By cycling through the prediction mode and BER improvement mode with appropriate actions by the executive, we have found that the data rate and Q-factor can be enhanced by 12.5% and 2.74 dB, respectively, as compared to the conventional fiber optic system. The key advantage of CDS is that it is intelligent, and software-defined, and it can automatically recognize and extract information about the fiber optic channel. In addition, it can enhance the data rate and improve the BER for different fiber optic link characteristics such as the span length, input power, and other system/signal parameters. These characteristics may change during the data transmission in a fiber optic network, but the CDS can still adapt to these changes. The computational cost of the proposed technique is much lower than the methods used to mitigate nonlinear impairments, such as digital back propagation (DBP). It should be noted that the simplified CDS

implemented in this chapter is not a substitute for nonlinear compensation techniques such as DBP, although it provides performance benefits comparable to that obtained by DBP.

Chapter 4

CDS V3 FOR COGNITIVE DECISION MAKING OVER NGNLEs[2]

The autonomic-computing layer of the smart systems based on a cognitive dynamic system (CDS) is proposed as a solution for better decision making and situation understanding in non-Gaussian and nonlinear environments (NGNLE). A cognitive decision-making (CDM) system inspired by the human brain decision-making process is discussed. The simple low complexity algorithmic design of the proposed system can make it suitable for real-time applications. A case study of the implementation of the CDS was done on a long-haul fiber-optic orthogonal frequency division multiplexing (OFDM) link. An improvement in Q-factor of 3.5 dB as well as 23.3% data rate efficiency enhancement are achieved using the proposed algorithms to keep CDM error automatically under the system threshold. The proposed system can be extended as a general software-based platform for brain-inspired decision making in smart systems in the presence of nonlinearity and non-Gaussian characteristics. Therefore, it can easily upgrade a conventional system to a smart one for autonomic CDM applications.

The chapter is organized as follows. In section 4.1, the introduction on CDS v3 as CDS with advanced executive is provided. Section 4.2 presents the architectural structure of CDS and the proposed algorithms CDM on general NGNLE. In section 4.3, we discuss the simulation results for the case study of a long-haul fiber-optic OFDM link. Finally, in section 4.4, we summarize the main contribution of our work.

[2] Most of chis chapter published as: M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain-Inspired Cognitive Decision Making for Nonlinear and Non-Gaussian Environments," *IEEE Access*, *7*, 180910–180922, 2019.

4.1. CDS v3

In [1.10] and chapter 3, the CDS was proposed for smart fiber optic communication systems as an example of smart systems using cognitive decision-making (CDM). The CDS v1 is used to control the quality of service in long-haul fiber-optic links. The CDS can take some actions such as changing the data rate so that the bit error rate (BER) is under the predefined threshold. In Chapter 3, the CDS with a simple executive was presented. The simple executive cannot predict the outcome of the actions before applying actions on the environment and after applying action on environment could find out the outcome of action by received internal rewards from perceptor. Besides, the simple executive cannot control the modeling configuration of the perceptor.

In this chapter, the algorithms for the CDS v3 is presented for the CDM in NGNLE. The proposed algorithms for the CDS uses the advanced executive. The advanced executive can predict the outcome of the multiple actions before applying action to the environment making use of virtual environment. In addition, the advanced executive can change the modeling configuration of perceptor through the internal commands. At the end of this chapter, the case study CDS is presented for NGNLE long-haul fiber-optic link. It is demonstrated that this new design can provide 23.3% data-rate efficiency enhancement as well as 3.5 dB Q-factor improvement using the proposed fast algorithms. The CDS can always keep the decision-making error under the system threshold, and this is a typical function of the CDS. A comparison between the proposed method and the related work on long-haul fiber-optic communications case study chapter 3 is given in Table 4.1.

Table 4.1: Comparison between proposed work and CDS v1 [1.11]

References	Technique implemented	Q-factor improvement	Data rate enhancement
CDS v1 ([1.10])	Simple CDS	2.7 dB	13%
CDS v3 ([1.11])	CDS with action outcome prediction (Virtual actions)	3.5 dB	23.0%

4.2. Design of CDS Architecture

The detailed CDS architecture for decision-making is shown in Figure 4.1. The proposed CDS in this chapter can operate in four modes (i) pre-adaptive actions mode (ii) environmental actions mode, (iii) post-adaptive actions mode, and (iv) steady-state. The executive functions will be different depending on the CDS mode. The steady-state mode means all is well and no more actions are required. Pre-adaptive mode actions are related to finding the first action to apply to the environment. The environmental actions correspond to actions that can be done on the environment for better decision making or total system performance enhancement. When the CDS operates in post-adaptive action mode, internal commands are sent to the perceptor from the executive using internal feedforward link for improving the decision making accuracy.

Figure 4.1 shows that the CDS uses a database for training and Bayesian model extraction. The perceptor is also responsible for knowledge extraction (see section 4.2.1). The knowledge is used to reason and predict the unknown situations for decision-making. The knowledge in the perceptor is represented as a set of concepts and the relationships among them, together with the environment situation estimations. Therefore, in a specific situation or on user request, the CDS switches from steady-state to the pre-adaptive mode. In the next sub-sections, the detailed algorithmic descriptions of preceptor and executive and those of mode (ii) and mode (iii) are presented.

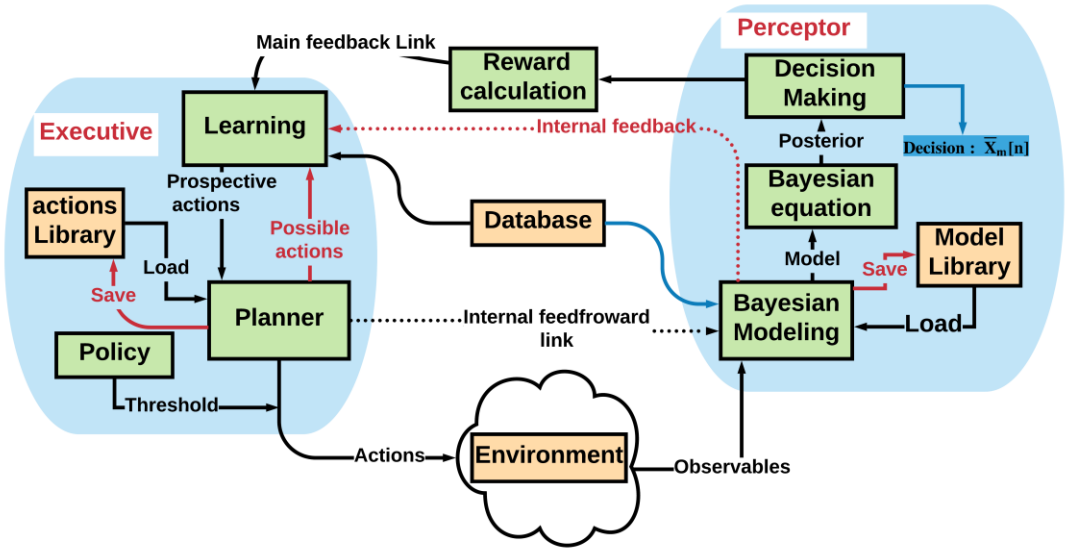


Figure 4.1: Architecture of CDS for CDM in NGNLE [1.11].

4.2.1. Perceptor

The main sub-blocks of the perceptor are (i) a three-layered Bayesian generative model, (ii) previous model selection, and (iii) *Bayesian equation* and decision-making. The multilayer Bayesian modeling is the same as that discussed in Chapter 3. The extracted model is passed to the block 'Bayesian equation' to calculate the posterior.

Bayesian equation and maximum posteriori selection

The *Bayesian equation* approach is used for extracting the posterior in decision-making, which is simpler than the *Bayesian filtering* approach. This is because we need to design low-complexity and real-time algorithms in NGNLE applications. The posterior can be calculated using the Bayesian equation as

$$P(\hat{X}_n^k \mid \hat{Y}_n^k) = \frac{P(\hat{Y}_n^k \mid \hat{X}_n^k) P(\hat{X}_n^k)}{P(\hat{Y}_n^k)}, \qquad (4.2)$$
$$b_k(\hat{X}_n^k, \hat{Y}_n^k) = P(\hat{X}_n^k \mid \hat{Y}_n^k),$$

Here, k denotes the PAC number, and n means CDM at time n. The estimated *Bayesian generative model* $P(\hat{Y}_n^k \mid \hat{X}_n^k)$ and the evidence $P(\hat{Y}_n^k)$ are extracted in Layer III. The

$P(\hat{X}_n^k)$ can be estimated using the database. $b_k(\cdot)$ is the extracted posterior at the kth PAC.

Maximum posteriori selection and assurance factor

The CDS can save the estimated posterior $P(\hat{X}_n^k \mid \hat{Y}_n^k)$ in the perceptor library after one-time calculation, and it is not necessary to calculate eq. (4.2) for each symbol. We can use the posterior $P(\hat{X}_n^k \mid \hat{Y}_n^k)$ to select the $\bar{X}_n^k$ that has the maximum probability in each discretized cell as:

$$\bar{X}_n^k = \arg\max_{l=1,2,...,M} \{P(X_n^l = X_m \mid \hat{Y}_n^k)\}, \tag{4.3}$$

Here l =1,2,…, M, where M is the total number of decision making situations.

4.2.2. Feedback channel, assurance factor and internal rewards

As we mentioned in chapter 1, typically, the CDS uses the Kalman filter and Shannon theory [1.8] to calculate the entropic state at time n. The above technique is applied on the linear and Gaussian environment (LGE). However, in a NGNLE, the complexity of using above algorithm is higher than that in LGE. We would like to prevent using extra logarithms, complicated functions, or the integrals required in the entropic state calculation with Kalman filter and use simple, straightforward and faster algorithm. Most of the time, we make a wrong decision when we do something with an assurance <100%, such as gambling. Similarly, the CDS makes a decision using eqs. (4.2) and (4.3), and selects the maximum probability using maximum a posteriori probability (MAP) rule. Therefore, the CDM error is proportional to assurance (when the assurance is less than 100%). Consequently, we define the average assurance factor (AF) as

$$AF_n^k = \frac{\sum_{o=n-L}^{n} P(\bar{X}_o^k \mid \hat{Y}_o^k)}{L}, \tag{4.4}$$

where L is an arbitrary time-discrete interval to calculate the assurance factor and $P(\bar{X}_o^k \mid \hat{Y}_o^k)$ refers to the maximum posterior at discrete time *o* and PAC *k*. The increment of the assurance factor in *k*th PAC is given by

$$\Delta_k^{AF_n} = AF_n^k - AF_n^{k-1}, \tag{4.5}$$

where the AF_n^{k-1} and AF_n^k are the assurance factors that can be calculated at (*k*-1)th and *k*th PAC, respectively. The preceptor sends the AF_n^k and ΔF_n^k to the executive by the global feedback channel as the raw internal rewards. The internal reward (cost to go function), denoted as the rw_n^k, may be defined as an arbitrary function of AF_n^k and $\Delta_k^{AF_n}$ as:

$$rw_n^{\ k} = f(AF_n^k,\ \Delta_k^{AF_n}, c_k, k), \tag{4.6}$$

where $f(\cdot)$ is an arbitrary funcation. The $f(\cdot)$ can be defined, based on the environment type (e.g. human body or fiber-optic channel, etc) or the preceptor configuration such as desired computational cost or accuracy. For example, in fiber-optic communications, the function $f(\cdot)$ should be wisely chosen to optimize between data-rate (action) and BER (internal reward) for the desired quality-of-service. The rw_n^k is calculated in the executive by the received raw internal rewards through the global feedback channel. In addition, the perceptor sends the estimated *Bayesian model* and *evidence* through the internal feedback channel. Also, the perceptor receives the modeling configurations such as PF_k and PB_k through the internal feedforward channel that are internal commands from the executive. Here, PF_k and PB_k are the precision factor and probability box percentage at the *k*th *PAC*, respectively. The executive uses this information to find the proper action for applying on NGNLE. Table 4.2 shows the important notations that are used in this chapter.

Algorithm 4.1: Environmental actions through the global feedback

---

Input: the observables $Y_n^k, k = 1, 2, ..., K$

Output: c_{k+1} as the final action in environmental actions mode

Initialization:

Pre-adaptive actions:

Make steady state mode off, environmental actions on

$c_1 \leftarrow$ *first enviromental action from C*

Apply c_1 to the environment (e.g., fiber optic channel)

Probability box% = PB1, threshold, $PF = PF_1$

for k=1 to K **then**

Take the observable Y_n^k

if the model is not available **then**

if k>1 **then**

Predict $P(\hat{X}_n^k \mid \hat{Y}_n^k)$ using $P(\hat{Y}_n^{k-1} \mid \hat{X}_n^{k-1})$

Estimate $\bar{X}_n^k$ by MAP rule

End if

Extract the $P(\hat{Y}_n^k \mid \hat{X}_n^k)$ and $P(\hat{Y}_n^k)$

Calculate the posterior $P(\hat{X}_n^k \mid \hat{Y}_n^k)$

Estimate $\bar{X}_n^k$ by decision making

Else if model is available **then**

Load model, evidence and posterior from preceptor library

Estimate $\bar{X}_n^k$ by MAP rule

End if

Send PF_1, $P(\hat{Y}_n^k \mid \hat{X}_n^k)$ and $P(\hat{Y}_n^k)$ to the Executive

Internal reward

Calculate rw_n^k and send it to executive

Planning

$rw_n^{(k+1),0} \leftarrow rw_n^k$

Localize the set of all close actions to c_k

Learning

If actions belong to environmental set **then**

for t=1 to *T* **then**

Apply c_{k+1}^t **virtually (**$c_{k+1}^t \in C$**)**

Calculate $P(\hat{X}_n^{(k+1),t} \mid \hat{Y}_n^{(k+1),t})$

Estimate $\bar{X}_n^{(k+1),t}$

Calculate $rw_n^{(k+1),t}$

if $rw_n^{(k+1),t} \geq rw_n^{(k+1),t-1}$ **or** $AF_n^{(k+1),t} \geq threshold$ **then**

$t' \leftarrow \arg\min_{t''=0,1,2,\ldots,(t-1)} location(rw_n^{(k+1),t''})$

$c_{k+1} \leftarrow c_{k+1}^{t'}$

break for

End if

End for

if $t' \geq 1$ **then**

32: apply c_{k+1} to the environment
33: **else**

Policy updating

34: Update *threshold*

Planning update

35: Switch actions type to *internal commands*
36: Start post adaptive actions
37: **End if**
38: **End for**

Table 4.2: The important notation list used in this chapter [1.11]

Notations	Value
K	The total number of environmental PACs
K	The current PAC number (environmental)
G	The total number of probability box PACs
G	The current PAC number (probability box)
F	The total number of precision factor PACs
F	The current PAC number (precision factor)
$\mathbf{C}$	The set of all possible actions in actions library
$\mathbf{X}$	The set of all discrete situations of NGNLE
M	Maximum number of the situations for CDM
L	The discrete time interval for AF calculation
X_n^k	The current NGNLE situation, at time n
$\hat{X}_n^k$	The training data from database using for modeling
$\bar{X}_n^k$	The estimated decision by the system
Y_n^k	The observables at time n and kth PAC
$\bar{Y}_n^k$	The normalized observables at time n and kth PAC
$\hat{Y}_n^k$	The discretized observables at time n and kth PAC
AF_n^k	The assurance factor at time n and kth PAC
rw_n^k	The internal reward (Cost-to-go function) at time n and kth PAC
$b_k(\hat{X}_n^k, \hat{Y}_n^k) = P(\hat{X}_n^k \mid \hat{Y}_n^k)$	The estimated posterior at time n and kth PAC

c_k	Current action applied on NGNLE
threshold	The predefined threshold
T	The number of virtual actions can predict by current available model at time *n* and *k*th PAC
PF_f	The precision factor in *f*th PAC (shunt cycle)
PB_g	The probability box in *g*th PAC (shunt cycle)
$b_k(\hat{X}_n^{(k+1),t}, \hat{Y}_n^{(k+1),t})$	The *t*th prediction using model at *k*th PAC
$AF_n^{(k+1),t}$	The *t*th assurance factor prediction in executive
$rw_n^{(k+1),t}$	The *t*th internal reward (cost-to-go function) prediction in executive
c_{k+1}^t	The *t*th virtual action at time *n* and *k*th PAC
c_{k+1}	The final action for applying in *(k+1)*th PAC

4.2.3. Executive

The executive is the most important part of the CDS. It is responsible for improving decision-making accuracy. Decision-making accuracy improvement can be achieved by applying action on NGNLE. For example, the executive can activate the actuators in the smart home or send internal commands to the perceptor for changing the modeling configurations. The executive brings non-monotonic reasoning to the CDS by using the internal reward. The executive that we design here has three parts: planner (it consists of the actions library also), policy, and learning using prediction.

Planner and policy

The planner extracts the set of prospective actions that are already saved in the CDS actions library. In addition, the planner selects the first action in 1st PAC using pre-adaptive actions. In addition, the planner applies the post-adaptive actions through internal feedback and feedforward channels (i.e., shunt cycle). In this chapter, the policy determines the desired goals that the CDS should achieve using PAC. The goal can be the trade-off between desired CDM accuracy and computational cost. Here, we define the CDS goal as the *threshold*, for simplicity.

Learning using prediction

a) Executive actions

The actions in the executive can be classified as the environmental actions and internal CDS commands. Therefore, we define the actions space as

$$C=\left\{ c_k \middle| c_k = \begin{array}{c} Environmental\ actions, or \\ Internal\ CDS\ commands = \left\{ \begin{array}{c} PF_f, or \\ PB_g \end{array} \right. \end{array} \right\} \tag{4.7}$$

where c_k is the action in *PAC* number k and $\boldsymbol{C}$ is set of all possible actions in action space (see Table 4.2 also).

Since the physical action to change the environment could be expensive, the executive performs virtual actions and predicts the outputs of the sensors based on the statistical model of the NGNLE. The virtual action means that the executive can virtually apply action to the virtual environment and predict the action outcome as prospective internal rewards or assurance factor. We first calculate the maximum ratio of the standard deviation of virtual sensor outputs at kth and (k-1)th PAC as (see Table 4.2 for the notation)

$$ratio^{k+1,t} = \max_{m=1,2,...,M} \{ \frac{std(\hat{Y}_n^k \mid \hat{X}_n^k = X_m)}{std(\hat{Y}_n^{k-1} \mid \hat{X}_n^{k-1} = X_m)} \}, \tag{4.8}$$

where $t \in \{0,1,2,...,T\}$ is the index for the virtual action and T is a total number of desired virtual actions for which the model at kth PAC is still valid. The ratio of standard deviations in eq. (4.8) gives insight to the executive how the observables are changed by applied action in current kth PAC in comparison the applied action in previous (k-1)th PAC. The virtual sensor output at (k+1)th PAC and tth virtual action cycle index is

$$\hat{\mathbf{Y}}_n^{(k+1),t} = h\left(\hat{Y}_n^k, ratio^{k+1,t}, c_k, c_{k-1}, t \right), \tag{4.9}$$

where $h(\cdot)$ is an arbitrary funcation. The $h(\cdot)$ can be defined based on the environment type (e.g. human body or fiber-optic channel, etc). The posteriori due to the virtual action c_{k+1}^t is predicted using

$$P(\hat{X}_n^{(k+1),t} \mid \hat{Y}_n^{(k+1),t}) \simeq b_k(\hat{X}_n^{(k+1),t}, \hat{Y}_n^{(k+1),t}), \tag{4.10}$$

In eq. (4.10), the $b_k(\cdot)$ is received through the internal feedback from the perceptor (See eq. (4.2)). The predicted assurance factor may be calculated as

$$AF_n^{(k+1),t} = \frac{\sum_{o=n-L}^{n} b_k(\bar{X}_o^{(k+1),t}, \hat{Y}_o^{(k+1),t})}{L}, \tag{4.11}$$

and

$$\Delta_{k+1}^{AF_n^t} = AF_n^{(k+1),t} - AF_n^k, \tag{4.12}$$

Then, the predicted internal rewards for the desired virtual action c_{k+1}^t is

$$rw_n^{(k+1),t} = \begin{cases} f(AF_n^{(k+1),t}, \Delta_{(k+1)}^{AF_n^t}, k+1), & t \in \{1,2,...,T\} \\ rw_n^k & t = 0 \end{cases} \tag{4.13}$$

The virtual action index *t* bar that leads to the minimum internal reward is

$$t_{opt} = \arg\min_{t \in \{0,1,2,..,T\}} (rw_n^{(k+1),t}) \tag{4.14}$$

If $t_{opt} = 0$ CDS switch to send internal commands instead of applying environmental actions. But, for $t_{opt} \geq 1$, the actual actions to be applied on the environment is

$$c_{k+1} = c_{k+1}^{t_{opt}}, \tag{4.15}$$

c_{k+1} is the best action to apply on the environment to improve the decision-making performance. Algorithm 4.1 shows outline of the main algorithmic process for the global PAC of the CDS. In some applications, the policy may update the *threshold* to change the accuracy for a lower complexity with acceptable accuracy.

b) Post adaptive actions: Internal commands

As mentioned before, some actions apply on the environment and the CDS receives the internal reward through the global PAC using the main feedback channel. However, the actions can be internal commands that the CDS can use for self-tuning i.e. the trade-off between computational cost and accuracy according to the policy. Based on eq. (4.7), internal commands are a precision factor $PF_f \in \{PF_1, PF_2, ..., PF_F\}$, and probability box, $PB_g \in \{PB_1, PB_2, ..., PB_G\}$. Here, *F* and *G* are maximum PAC indices related to *PF*

and *PB* actions, respectively. The process is similar for both *PF* and *PB* actions and it is presented in detail through Algorithm 4.2.

Algorithm 4.2: Post adaptive actions (shunt cyle actions)

Input: *threshold,* c_k , *PB₁*, *PF*

Output: the $c_{(k+f+g-1)}$ as the final actions

1: **for** *f = 2 to F* **then**
2: Action $c_{K+f} \leftarrow PF_f$
3: Send command c_{K+f} to reduce *PF*
4: Receive $P(\hat{Y}_n^{k+f} \mid \hat{X}_n^{k+f})$ and $P(\hat{Y}_n^{k+f})$
5: Calculate the $P(\hat{X}_n^{k+f} \mid \hat{Y}_n^{k+f})$ then $\bar{X}_n^{k+f}$
6: Estimate the rw_n^{k+f}
7: **if** the $rw_n^{k+f} < threshold$ and $rw_n^{k+f} > 0$ **then**
8: Applying c_{k+1}, and c_{k+f}
9: Turn Steady state mode on
10: **Break for**
11: **End if**
12: **End for**
13: **if** steady state mode off **then**
14: **for** *g=2 to G*
15: Action $c_{k+f+g-1} \leftarrow PB_g$
16: Send command $c_{k+f+g-1}$ to change *PB*
17: Receive $P(\hat{Y}_n^{k+f+g-1} \mid \hat{X}_n^{k+f+g-1})$ and $P(\hat{Y}_n^{k+f+g-1})$
18: Calculate the $P(X_n^{k+f+g-1} \mid \hat{Y}_n^{k+f+g-1})$
19: Estimate $\bar{X}_n^{k+f+g-1}$, then $rw_n^{k+f+g-1}$
20: **if** $rw_n^{k+f+g-1} < threshold$ and $rw_n^{k+f+g-1} > 0$ **then**
21: Apply c_{k+1}, c_{k+f} and $c_{k+f+g-1}$
22: Turn Steady state mode on
23: **Break for**
24: **End if**
25: **End for**
26: **End if**

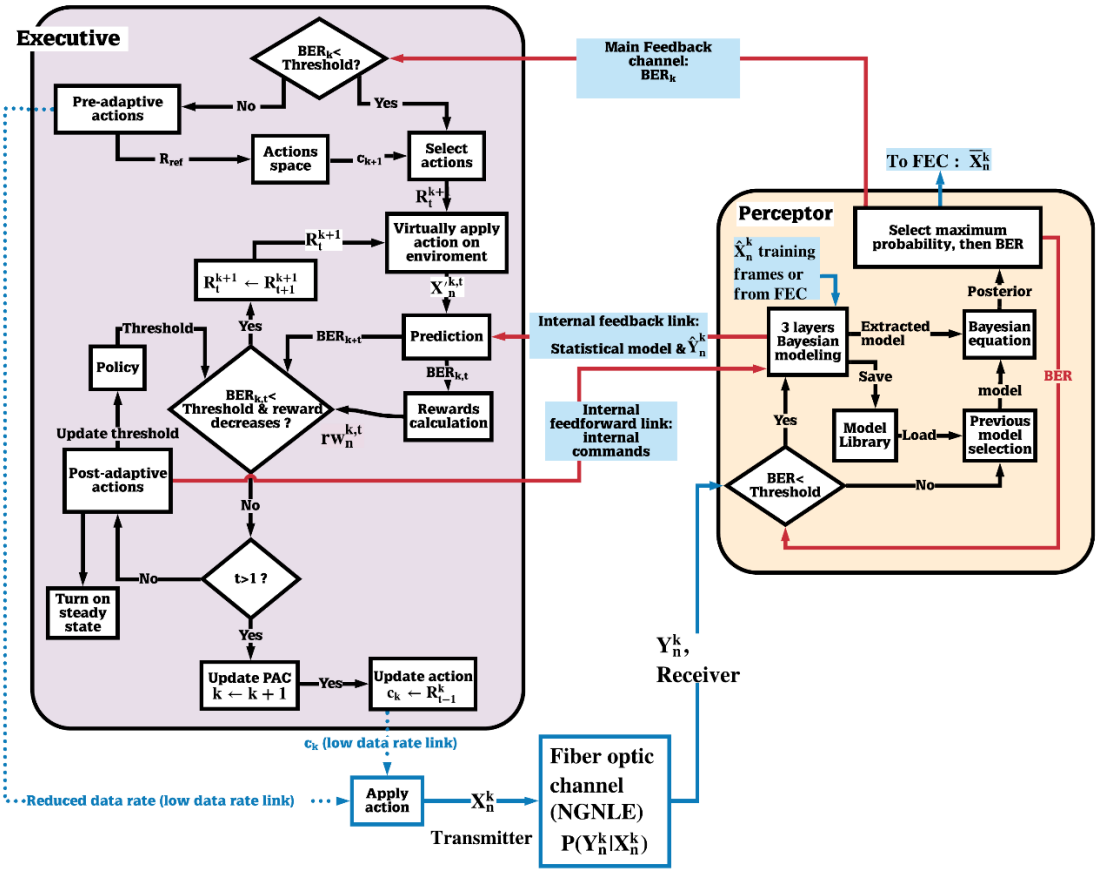


Figure 4.4: CDS architecture for long-haul NGNLE OFDM-based Fiber Optic link [1.11].

4.3. CDS implementation case study: OFDM long-haul fiber optic communications

Figure 4.4 shows the CDS architecture block diagram for the OFDM long-haul fiber optic link. As mentioned before, we select the fiber optic link as an example of NGNLE. In general, $\hat{X}_n^k$ can be supplied by the database shown in Figure 4.4, i.e. the transmitter may send a known sequence during the training period. Alternatively, the estimated $\hat{X}_n^k$ after the FEC may be used instead of the sequence from the database for model extraction. The bit error rate (BER) after the forward error correction (FEC) is between 10^{-15} and 10^{-9} [3.2] when the BER before the FEC is less than the FEC-threshold. In addition, a good measure of the decision error is the BER. The BER can be estimated using 16 or 32 training frames that are used for linear distortion mitigation. The channel

estimation and compensation of linear impairments are done using the linear adaptive channel equalizer that uses 16 training frames to mitigate linear distortion [2.4]. Therefore, the prospective internal reward can be calculated as follows (see eq. (4.13)):

$$rw_n^{k,t} = \frac{BER_n^{k+t}}{\left(R_t^k - R_{ref}\right)^4}, \tag{4.16}$$

The prospective data rate at k th PAC and t th virtual action cycle is

$$R_t^k = R_{ref} + t \times d, \tag{4.17}$$

where d is 4 Gb/s discretization step for data rate and R_{ref} is a convenient reference the data rate that is

$$R_{ref} = R_1^1 - d \tag{4.18}$$

The actions space for this specific case study is (see eq. (4.7)):

$$C = \left\{ c_k \middle| c_k = \begin{array}{c} data\ rate\ R_k > R_{ref},\ or \\ -4 \le P_{tx}^k \le 3\ dbm, or \\ PF_f \in \{0.01, 0.05, 0.1, 0.5, 1, 2\}, or \\ \text{probablity box\%, } PB_g \in \{90, 96, 99\} \end{array} \right\} \tag{4.19}$$

Here, $P_{tx}{}^k$ is the input launch power in kth PAC number. In addition, eqs. (4.8)-(4.10) can be used for the virtual actions posteriori estimation in this case study (see Algorithm 4.1 and in Appendix I).

Table 4.3: The numerical simulation parameters of the OFDM system [1.11]

Simulation parameters	Value
Fiber dispersion coefficient (β_2)	$-22.1\ \mathrm{ps}^2/\mathrm{km}$
Nonlinear coefficient (γ)	$1.1\ \mathrm{W}^{-1}\mathrm{km}^{-1}$
Fiber loss coefficient (α)	$0.2\ \mathrm{dB/km}$
Number of fiber spans (N)	20
Span length (L_{span})	80 km (1600 km total length)
Noise figure (NF)	4.77 dB
Number of OFDM subcarriers per frame	256
Subcarrier modulation	QAM-16
Number of data frames	128000
Number of frames for modeling	115200

Number of frames for testify	12800
Data carrying subcarriers per frame	126
Oversampling factor	2
Number of the pilot subcarrier	2
Discretization step *d*	4 Gb/s
PF_1	2
Fiber type	Standard single mode fiber (SMF)

4.3.1. Simulation parameters and system configurations

We illustrate the modeling process by performing numerical simulation of the OFDM system. The signal propagation in optical fiber is described by the nonlinear Schrodinger equation (NLSE), which is solved using the standard split-step Fourier scheme [2.3][2.4]. The fiber optic system without the CDS is identical to that described in Chapter 3 except that a fixed dispersion compensator is introduced here just before the FFT block. The output of the linear equalizer is passed to the preceptor (See Figures 2.6 and 2.7). X_n^k represents the transmitted symbols which take the values from the following set $X_n^k \in \{\pm 3 \pm 3j, \pm 1 \pm 1j, \pm 1 \pm 3j, \pm 3 \pm 1j\}$ with equal probability and Y_n^k represents the corresponding received symbols after linear equalization.

The simulation parameters of fiber optic system are presented in Table 4.3. The CDS parameters are as follows: *PF* = 2, probability box percentage = 96%, and FEC threshold = 1.03×10^{-2} (with 14.29% Overhead (OH) for the hard decision (HD)) [3.3] before post adaptive actions. However, these parameters can be changed by the CDS using internal commands, adaptively (Algorithm 4.3). Here, the updated Algorithm 4.2 is presented as the Algorithm 4.3 and, for the fiber optic link case study as an example of NGNLE.

Algorithm 4.3: Post adaptive actions for the fiber optic link (shunt cyle actions)

Input: $threshold = 4.7\times10^{-3}$, c_K, R_K, *PF set actions*

Output: the c_{K+F+G} as the final actions

Initialization:

Probability box = 96%

1: **for** $f = 2$ *to* F **then**
2: Action $c_{K+f} \leftarrow PF_f$
3: Send action by shunt cycle to perceptor
4: Receive $P(\hat{Y}_n^K \mid \hat{X}_n^K)$ and $P(\hat{Y}_n^K)$ from perceptor
5: Calculate the $P(X_n^{K+f} \mid \hat{Y}_n^K)$ then $\bar{\bar{X}}_n^{K+f}$
6: Estimate the BER_{K+f} and, rw_K^{K+f}
7: **if** the $BER_{K+f} < threshold$ **then**
8: Reduce the **OH** to 6.75%
9: Turn steady state mode on for R_K
10: **Break for**
11: **End if**
12: **End for**
13: **if** steady state mode off **then**
14: Action $c_{K+F+1} \leftarrow PB_2 = 99\%$
15: Repeat line 3 to 6 again
16: **if** the $BER_{K+f} < threshold$ **then**
17: Reduce the **OH** to 6.75%
18: Turn steady state mode on for R_K
19: **else**
20: Turn steady state mode on for R_K
21: **End if**
22: **End if**

4.3.2. Simulation results and discussions

1) Pre adaptive actions

The BER of a conventional fiber-optic communications system (i.e., without CDS, see Figure 2.6) is shown in Figure 4.5. We assume that the FEC threshold for HD is 1.03×10^{-2} (with 14.29% overhead (OH)) [3.3]. Thus, we can see from Figure 4.5 that the maximum achievable data rate is 208 Gb/s. It may be noted that the achievable data rates for the fiber optic example of Chapter 3 is around 50 Gb/s. The higher data rates for the fiber optic example of this chapter (even without CDS) is explained as follows. In Chapter 3, fiber dispersion is compensated using only an adaptive equalizer which is not very efficient. As the data rates increases, OFDM symbols overlap significantly due to fiber dispersion and thereby, subcarrier orthogonality is broken down. In the fiber optic example of this chapter, a fixed dispersion compensator is used prior to the

FFT so that the carrier orthogonality is maintained (cite our paper with Corning on OFDM).

The CDS uses 208 Gb/s for j =1 (i.e. first data rate, R_1^1 =208 *Gb/s)*. Also, we assume discretization step for data rate, d = 4 Gb/s, and a step of 1 dB for the launch power. Therefore, the reference data rate R_{ref} is *204 Gb/s* (see eq. (4.16)). Typically, the launch power is so optimized that the BER is minimum for the given data rate.

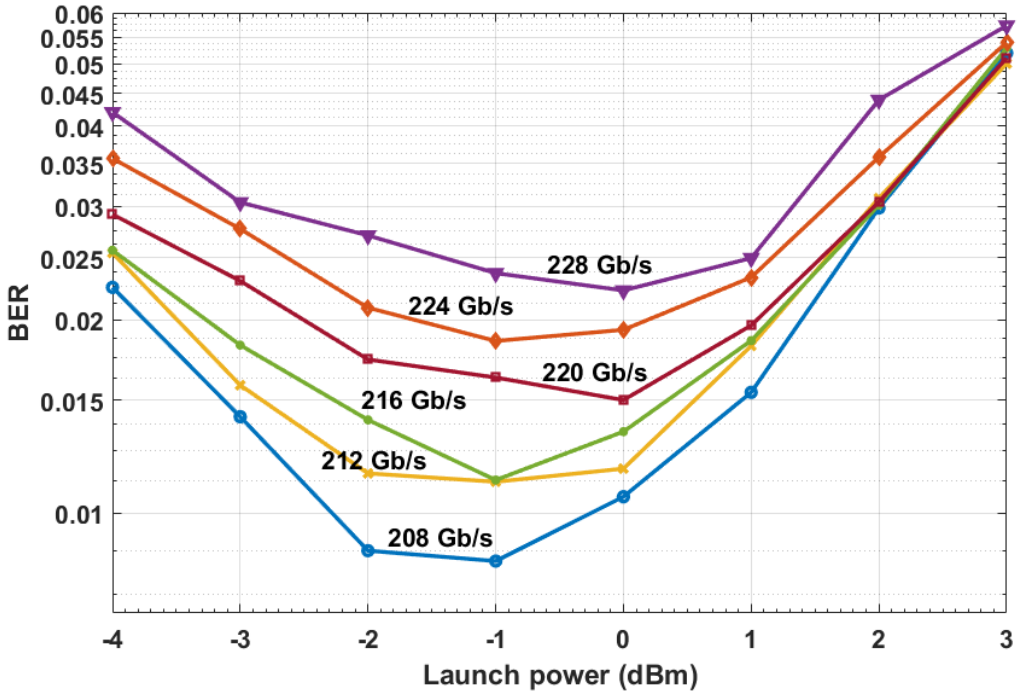
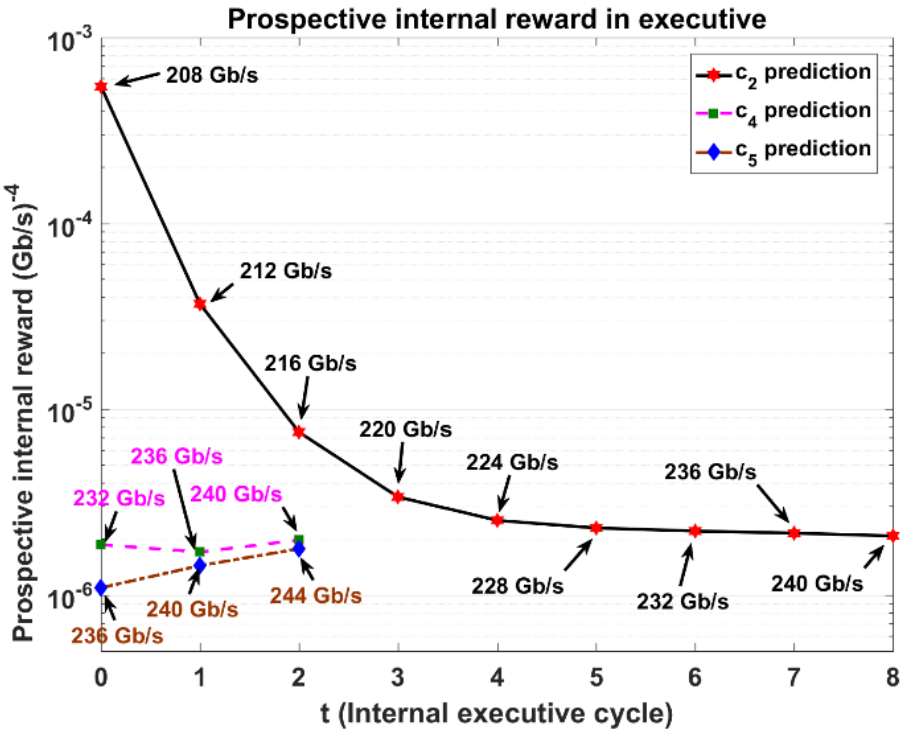


Figure 4.5: BER for a conventional system with the linear equalizer [1.11].

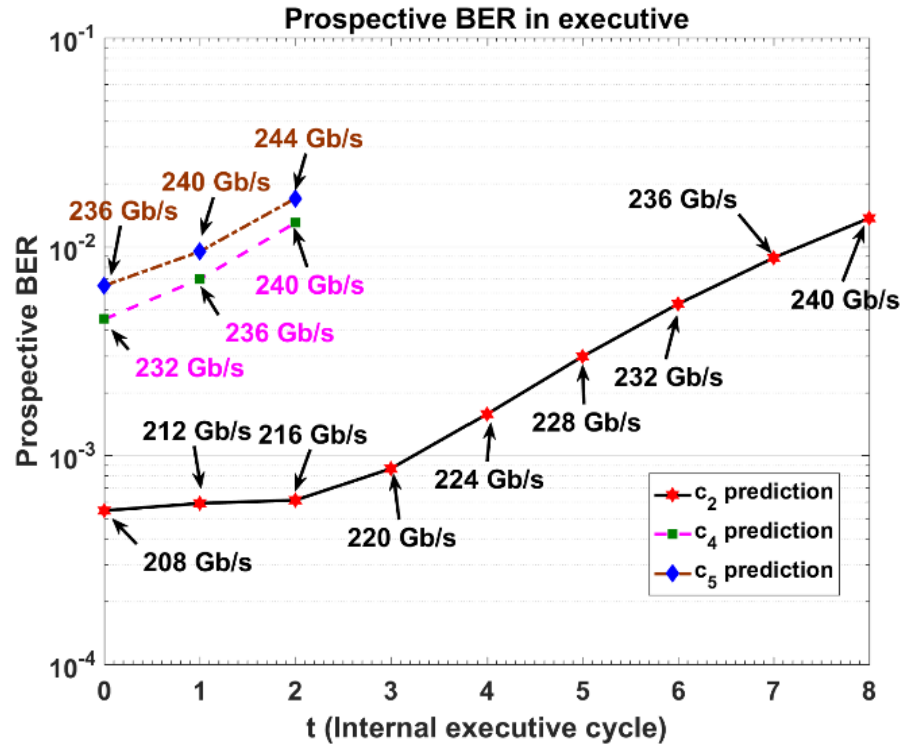
2) Results for the system with the CDS

Figures 4.6a and 4.6b show the prospective internal rewards and BER in virtual action cycles, respectively. In Figure 4.6, *"c_2 prediction"* means the executive applies virtual c_2^t action on a virtual environment (Here, $t \in \{1,2,\cdots,8\}$ and 2 is the PAC cycle index). Then, the executive predicts the best action that can be applied to a real environment as c_2. Fig. 4.8b shows the BER calculated in perceptor when this action is actually applied on the environment by the executive. The global PAC procedure can be summarized using Figure 4.4-4.7 as follows:

1. 1st PAC: CDS reduces the date rate to find the first data rate for which the BER is under the FEC-threshold (see Figure 4.5). Therefore, 208 Gb/s is applied to the fiber optic link as the first action (c_1). Because the BER is under the FEC threshold, the perceptor can extract the statistical model for 208 Gb/s. The perceptor can improve the BER using the extracted model (see Figure 4.7). Note that the minimum BER for a conventional system at 208 Gb/s is 8.0×10^{-3} which is reduced to 5.5×10^{-4} when the CDS is used. The reason for this improvement is explained in Chapter 3. The perceptor sends the extracted model to executive using internal feedback link.
2. 2nd PAC: The executive predicts the prospective internal rewards and BER for virtual action of c_2 (see Figure 4.6, and Algorithm 4.1, eq. (4.13)-(4.18)). The executive performs virtual actions of increasing the data rate, passing through a virtual fiber optic channel and measuring the BER. The data rate at the internal executive cycle index t is given by eq. (4.18). From Figure 4.6b, we see that at t = 8, the data rate R_8^2 = 240 Gb/s and the corresponding BER is above the FEC threshold. Therefore, the executive applies the data rate of 236 Gb/s as the actual action c_2 (which is the data rate at t = 7). Note that the prospective internal reward is minimum for the data rate of 236 Gb/s (See Figure 4.6a) that have the prospective BER is under the FEC threshold. After applying c_2 (i.e., a signal at a data rate 236 Gb/s is transmitted over a fiber optic channel) by the CDS, the perceptor uses the model corresponding to 208 Gb/s (see Figure 4.7, previous model selection) to predict the BER for 236 Gb/s since the model for 236 Gb/s is not available. In this case, at PAC 2, a signal at a data rate of 236 Gb/s is transmitted over a fiber optic channel and as can be seen from Figure 4.7, the BER is found to be above the FEC-threshold. Note that we used the previous model selection in this case study. By using previous model selection, the system does not need to transmit the training frames for model extraction.



(a)



(b)

Figure 4.6: BER using CDS in executive prediction for virtual actions (*PF*=2, *d*=4 Gb/s, and $P_{tx} = -1dBm$) (a) prospective internal reward, (b) prospective BER vs executive internal cycle) [1.11].

3. 3rd PAC: Since the BER is more than the FEC-threshold for action c_2, the executive reduces the data rate by d = 4 Gb/s (see Figure 4.4). As a result, the executive applies 232 Gb/s as c_3 in 3rd PAC number. Then perceptor uses 208 Gb/s model for BER estimation of the signal at a data rate of 232 Gb/s (See Figure 4.7) and it can bring the BER for 232 Gb/s under the FEC-threshold. Therefore, the perceptor can extract the model for 232 Gb/s. The extracted model results in further BER improvement (see Figure 4.7).

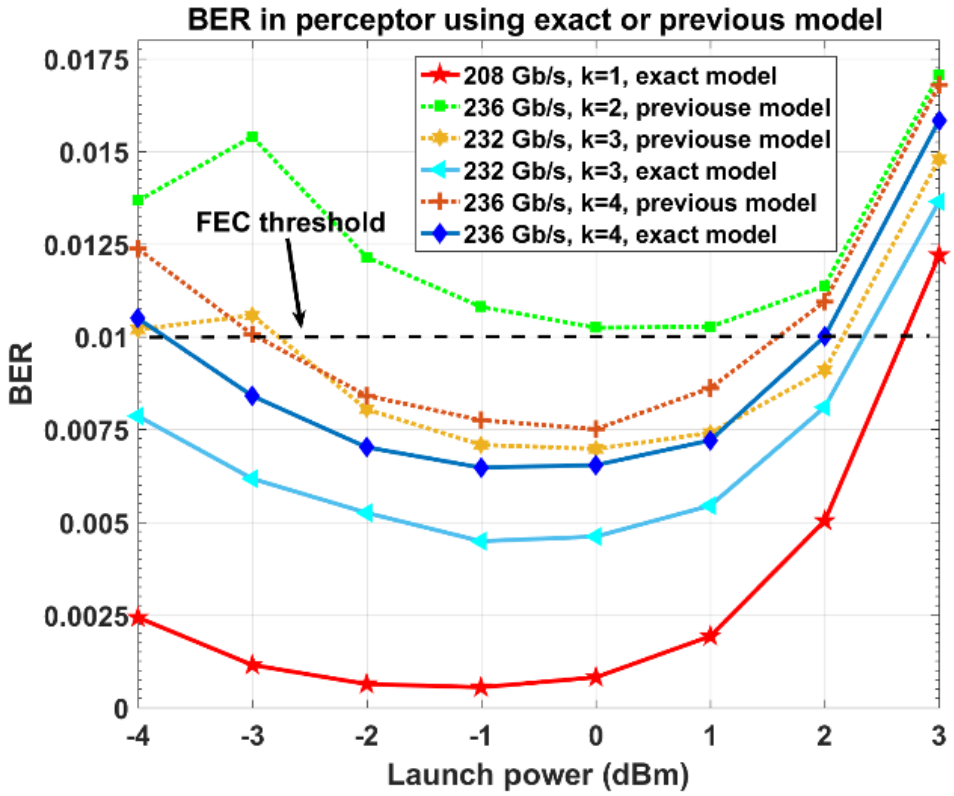
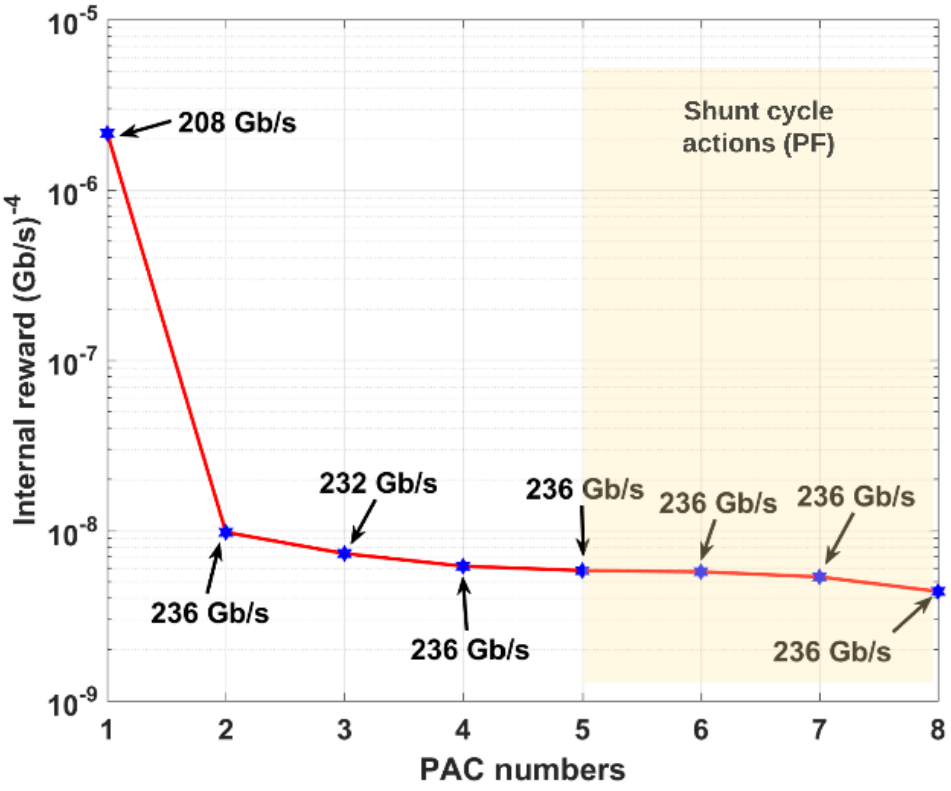
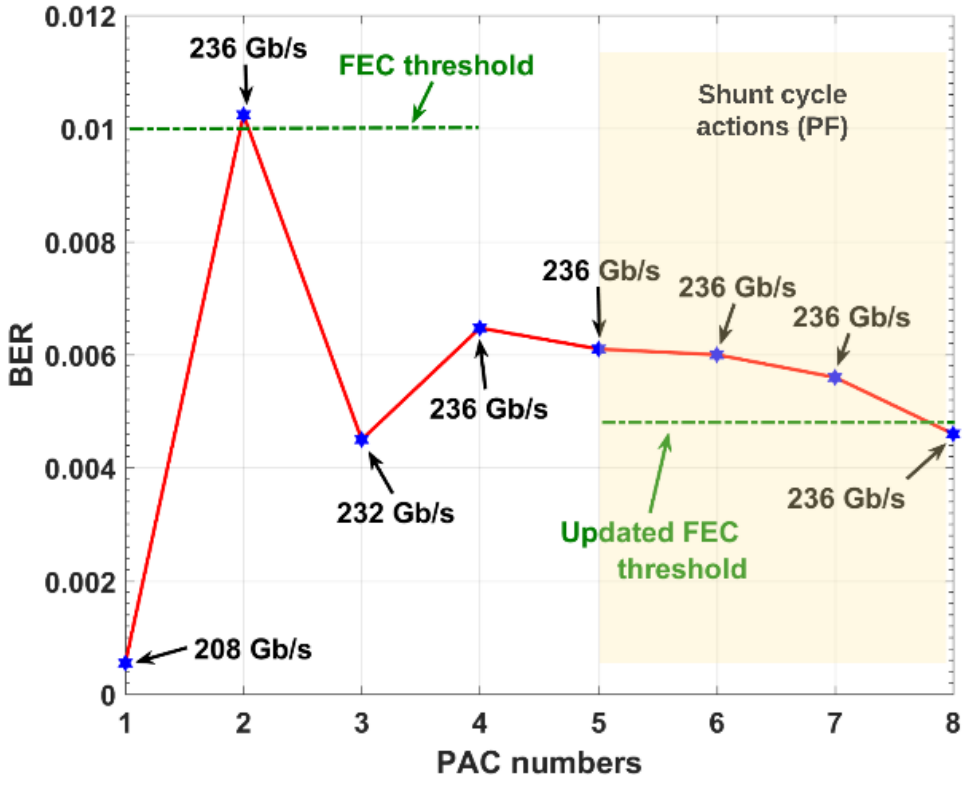


Figure 4.7: BER in perceptor after environmental actions (*PF*=2, *d*=4 Gb/s) [1.11].


Internal reward (Gb/s)^-4
PAC numbers
Shunt cycle actions (PF)
208 Gb/s
232 Gb/s
236 Gb/s
236 Gb/s
236 Gb/s
236 Gb/s
236 Gb/s
236 Gb/s

(a)



(b)

Figure 4.8: Results for the CDS after global and shunt cycle actions, (a) CDS learning curve, (b) BER vs PAC numbers [1.11].

4. 4th PAC: The executive sends the signal at a data rate of 236 Gb/s (See Figure 4.6) as c_4 to the environment because the predicted BER for 240 Gb/s is more than the FEC-threshold (see Figure 4.6b) and prospective internal reward increases for 240 Gb/s (see Figure 4.6a). Figure 4.7 shows that the perceptor can bring the BER for 236 Gb/s under the FEC-threshold using the model for 232 Gb/s. Like the 3rd PAC, the BER can improve further using the exact extracted model (see Figure 4.7).
5. 5th PAC: The executive finds that the predicted BER for 240 Gb/s is less than the FEC threshold (see Figure 4.6b), but the prospective internal reward increases in comparison to 236 Gb/s (see Figure 4.6a). Therefore, the executive turns on post-adaptive actions for the data rate of 236 Gb/s as explained below (also see Figure 4.4 and Algorithm 4.3).

Further reduction in the BER at the data rate of 236 Gb/s is possible by decreasing the precision factor. Based on [3.3], if the CDS brings the BER under 4.7×10^{-3} before the HD-FEC (new FEC threshold as the new goal for the CDS), the required OH can be reduced from 14.29 % to 6.25%. As a result, 19.85 Gb/s out of 236 Gb/s (which is 8.04 %) data rate can be used for data transmission instead of assigning for overhead. Therefore, the CDS decreases the precision factor *PF* to achieve this goal using internal commands as the post adaptive actions (see Algorithm 4.3). Decreasing the *PF* can improve the modeling accuracy, which results in improving the transmitted data estimation with lower BER.

The CDS learning curve is shown in Figure 4.8a and the corresponding BER is shown in Figure 12b. In Figure 4.8, the PACs 1,2,3 and 4 are the CDS global actions that correspond to the data rates of 208, 236, 232 and, 236 Gb/s respectively (see Algorithm 4.1). The PACs 5, 6, 7, and 8 are the CDS internal commands as the PF_2=1, PF_3=0.5, PF_4=0.1, and PF_5=0.05, respectively at 236 Gb/s data rate (see Algorithm 4.3, section 3.2.1). As can be seen from Fig. 4.8, the BER is decreased from 6.1×10^{-3} to 4.6×10^{-3} as the CDS goes from PAC 5 to PAC 8 by decreasing the precision factor.

Table 4.4: Results and the CDS parameters at steady state [1.11]

Results or CDS parameters	Value
Data rate	236 Gb/s
Precision factor	0.05
BER	$\mathbf{4.6\times10^{-3}}$
Q-factor improvement	3.5 dB

The results for the CDS at steady state are summarized in Table 4.4. The smart fiber-optic link enhances the 23.3% data rate efficiency in comparison to the conventional fiber optic link. This is the sum of (i) 28.0 Gb/s data rate enhancement from R_1 to R_4 and (ii) 19.85 Gb/s due to 8.04% HD overhead reduction by bringing BER under 4.7×10^{-3} . The Q-factor can be calculated from the measured BER [1.10][2.4]. The final Q-factor improvement is 3.5 dB, which is significant improvement for the long-haul fiber-optic link. The performance improvement using

digital backpropagation (DBP) is typically 1-2 dB [1.40], whose computational cost and complexity are significantly higher as explained below. Besides, the DBP does not provide data rate enhancement.

It is useful to compare the complexity of the proposed method in this chapter with the DBP for Q-factor enhancement using Table 4.3 simulation parameters. We run both algorithms on Microsoft Surface-Pro with Intel ® Core ™ i706650U CPU @ 2.20GHz 2.21 GHz, 16 GB RAM, system type 64-bit Operating System x64-Based processor using MATLAB. For the DBP algorithm [3.4], the runtime is 1225.4 Seconds (~20 minutes and 25 seconds). However, this runtime for the proposed CDS is 7.5, 14.03, 38.14 and, 135.8 seconds for *PF*s equals to 2, 1, 0.5, 0.1, and 0.05 respectively. In addition, The CDS runtime reduced to 3.9, 4.3, 4.4, 5.3, and 7.0 seconds for PFs 2, 1, 0.5, 0.1, and 0.05 respectively for the CDM in steady-state mode. However, the DBP running time will always remain similar for any continuous data stream.

4.4. Conclusions

In recent years, some efforts were made to develop intelligent machines using CDS for AI applications. However, the algorithms proposed in literature for CDS [1.8][1.9][1.12][1.13][1.18-1.25] are applicable to LGE, but they cannot be applied for NGNLE. Therefore, we proposed computationally efficient CDS algorithms applicable NGNLE. We redesigned the CDS in detail to make it suitable for NGNLE applications. The proposed algorithms are applied to the long-haul nonlinear fiber-optic link as a case study. By upgrading the conventional link to smart one, the achievable data-rate is increased from 208 Gb/s to 236 Gb/s with the help of CDS. Also, the CDS improves the Q-factor by 3.5 dB (the final BER = 4.6×10^{-3} at 236 Gb/s) as compared to the conventional system without CDS, which means the required HD overhead can be reduced. As a result, the final data rate enhancement is 23.3%.

The presented algorithms in this chapter are simple and fast because it used the Bayesian equation for calculating the posterior that has the lowest complexity as compared to the other algorithms proposed in literature for preceptor. This makes CDS perfect candidate for the NGNLE applications such as healthcare or long-haul fiber

optic link. Also, CDS v3 may be a good candidate for CDM sub-systems in next generations of future robots.

Chapter 5

CDS V4 FOR CDM OVER NGNLES WITH FINITE MEMORY [3]

The cyber processing layer of smart systems based on a cognitive dynamic system (CDS) can be a good solution for better decision making and situation understanding in non-Gaussian and nonlinear environments (NGNLE). The NGNLE situation understanding means deciding between certain known situations in NGNLE to understand the current state condition. A cognitive decision-making (CDM) system inspired by the human brain decision-making for NGNLE with finite memory is designed. The simple low-complexity algorithmic design of the proposed CDM system can make it suitable for real-time applications. A case study of the implementation of the CDS on a long-haul fiber-optic orthogonal frequency division multiplexing (OFDM) link was performed. An improvement in Q-factor of ~7 dB and an enhancement in data rate efficiency ~43% were achieved using the proposed algorithms. The proposed system can be extended as a general software-based platform for brain-inspired decision making in smart systems in the presence of nonlinearity and non-Gaussian characteristics. Therefore, it can easily upgrade the conventional systems to a smart one for autonomic CDM applications

The chapter is organized as follows. In section 5.1, introduction on CDS v5 is provied, In section 5.2, the architectural structure of CDS and the proposed algorithms CDM for a general NGNLE are presented. In section 5.3, we discuss the simulation results for the case study of a long-haul fiber-optic OFDM link. Finally, in section 5.4, we summarize the main contributions of our work.

[3] Most of this chapter published as: M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Natural Brain-Inspired Intelligence for Non-Gaussian and Nonlinear Environments with Finite Memory," *Appl. Sci.* 2020, *10*, 1150, 2020.

5.1. CDS v4

In chapter 3, the CDS v1 was proposed for smart fiber optic communication systems as an example of smart systems using cognitive decision-making (CDM). The CDS v1 is used to control the quality of service in long-haul fiber-optic links. The CDS v1 can take some actions such as changing the data rate so that the bit error rate (BER) is under the predefined threshold. However, the CDS v1 uses a simple executive. The simple executive cannot predict the outcome of the actions before applying actions on the environment. Besides, the simple executive cannot control the modeling configuration of the perceptor or predict the outcome of actions using a virtual environment. In chapter 4, the CDS v3 was presented for NGNLE applications using advanced executive that can predict the outcome of the actions. However, the perceptor of the CDS v3 can only extract the model of an environment without memory

In this chapter, CDS v4 is presented as a general-purpose algorithm for the CDS for a CDM system in a NGNLE with finite memory. The proposed CDS uses an advanced executive that can predict the outcome of multiple actions before applying an action to the environment. In addition, the advanced executive can change the modeling configuration of the perceptor through internal commands. Also, the perceptor can extract the model of a NGNLE with finite memory. In addition, a case study of new CDS and an algorithm are presented for a long-haul fiber-optic link as an example of NGNLE. A comparison between the proposed method and earlier versions for long-haul fiber optic communications case study is given in Table 5.1.

Table 5.1: Comparison between proposed CDS and earlier versions of CDS in previous chapters [1.35]

References	Technique implemented	Q-factor improvement	Data rate enhancement
CDS v1 [1.10]	Simple CDS	2.7 dB	13%
CDS v3 [1.11]	CDS with action outcome prediction (Virtual actions)	3.5 dB	23%
CDS v4 [1.35]	New CDS using different focus levels adaptively + Virtual actions	7 dB	43%

5.2. Design of CDS architecture for finite memory NGNLE

The detailed CDS architecture for decision-making is shown in Figure 5.1. In this section, we first describe the perceptor and executive using related algorithms when the executive takes actions on the environment, which is followed by changing the focus level when the CDS sends the command to the perceptor as post adaptive actions. he CDS can operate in four modes: (i) pre-adaptive actions mode; (ii) environmental actions mode;(iii) steady state mode; and (iv) changing the focus level. The first three modes are the same as that in v3 and the novel feature of this version of CDS is the focus level. For improving the decision-making accuracy, the CDS can increase the focus level of modeling in the perceptor and that of learning in the executive in mode (iv). In addition, the CDS can decrease the focus level for the desired complexity threshold in the mode (iv).

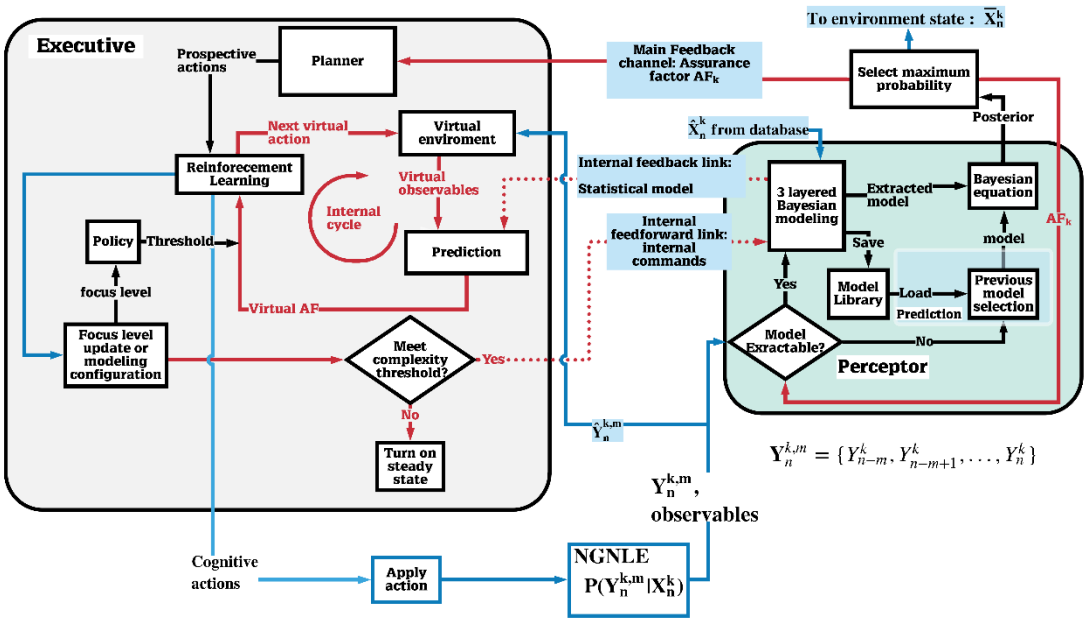


Figure 5.1: Architecture of proposed CDS for CDM in NGNLE with finite memory [1.35].

Figure 5.1 shows that the CDS can use a default database. However, it can be replaced using other techniques in some applications, such as in a long-haul nonlinear fiber optic system in which the database is replaced with transmitted training frames or the estimated data after forward error correction (FEC) as training frames. As mentioned

in previous chapters, the perceptor is responsible for knowledge extraction. The knowledge is used to reason and predict the unknown situations for decision-making. The knowledge in the perceptor is represented as a set of concepts and the relationships with the environment situations. The extracted knowledge can be refined if the CDS increases the focus level. In a specific situation or on user request, the CDS switches from steady state to the pre-adaptive mode. Also, the CDS can dynamically update knowledge in the perceptor and actions library in the executive. Next, the detailed algorithmic descriptions of environmental actions mode (ii) and changing the focus level mode (iv), are presented.

5.2.1. Perceptor

Similar to CDS v1 and v3, CDS v4 uses the *Bayesian equation* and the decision making tree approach for extracting the posterior in decision-making (i.e., $P(X_n \mid \mathbf{Y}_n^m)$), which is more straightforward than the *Bayesian filtering* approach in the conventional CDS (See Chapter 1). In addition, the multilayer Bayesian model extraction using the decision trees concept is a standard technique in machine learning. Here, we define *m* focus levels as the depth of the decision-making tree and *i* is the arbitrary decision-making node level between 0 to *m* depth levels (decision tree level). This is because it is required to design real-time algorithms for NGNLE applications with controllable complexity for example, increasing focus level will increase complexity and decision making accuracy. the main sub-blocks of the perceptor are: (i) Bayesian model extraction, (ii) Previous model selection, and (iii) *Bayesian equation* and decision-making. The multilayer Bayesian model for different focus levels implementation is disscussed next.

Bayesian Generative Model Extraction

Suppose that there exists no relevant model in a model library. The three-layered Bayesian modeling block extracts the statistical model of the system (see sections 3.2.1 and 4.2.1 for further details) using a decision-making tree. The 3-layered Bayesian modeling consists of three layers for an arbitrary focus level of *m*, as explained in a), b) and c) below:

a) Layer I:

In a NGNLE, some observables could have any value, and hence, a large number of discretized cells are required for saving vacant spaces. For example, the output of the fiber optic link is a complex variable $Y_n^{k,m}$, where *n* is the discrete time, *k* is the PAC number and *m* is the current focus level. Therefore, we define the probability box (PB) percentage. The data normalized in this way is sent to Layer II as the $\bar{\boldsymbol{Y}}_n^{k,m}$ for further processing. For the 2D observables, this box can be determined as the $x_{\min}^{k,m}$ and $x_{\max}^{k,m}$ for the minimum and maximum of the probability box in the *x*-axis together with $y_{\min}^{k,m}$ and $y_{\max}^{k,m}$ for minimum and maximum of the box in the *y*-axis. The internal reward is used to select the borders of the probabilistic box adaptively as the internal commands by the executive. More details provided on the PB parameters can be found in chapters 3 and 4.

b) Layer II:

In Layer II, the values related to observables are discretized with discretization steps Δx_i^k and Δy_i^k, which can be two-dimensional if the observables are complex numbers (e.g. see chapters 3 and 4) and $0 \le i \le m$. For simplicity, we define a *precision factor* (PF_i^k) for each focus level as (assume $\Delta x_i^k = \Delta y_i^k$)*:*

$$PF_i^k = 10\Delta x_i^k = 10\Delta y_i^k, \quad 0 \le i \le m, \tag{5.1}$$

$$\mathbf{PF}^{k,m} = [PF_0^k, PF_1^k, \cdots, PF_i^k, \cdots, PF_m^k], \tag{5.2}$$

$$F_i^k = N_x^{k,i} N_y^{k,i}, \quad 0 \le i \le m, \tag{5.3}$$

$$N_x^{k,i} = \frac{10(x_{\max}^{k,i} - x_{\min}^{k,i})}{PF_i^k}, \tag{5.4}$$

$$N_y^{k,i} = \frac{10(y_{\max}^{k,i} - y_{\min}^{k,i})}{PF_i^k}, \tag{5.5}$$

$$F_m^{total,k} = \prod_{i=0}^{m} F_i^k, \quad m \in \{0,1,2,\ldots,M\}, \quad \text{and, } F_m^{total,k} \le \textit{Complexity threshold}. \tag{5.6}$$

Here, $N_x^{k,i}$ and $N_y^{k,i}$ are the number of discretized cells in the x-axis and y-axis, respectively. Furthermore, F_i^k corresponds to the number of decision-making tree branches (see eqs. (5.1) and (5.3)). It should be noted that the required storage memory increases exponentially with increasing focus levels (see eq. (5.6)). In addition, the perceptor requires a larger database to extract a reliable model. Therefore, the complexity threshold is predefined based on the available storage memory for saving the model on hard disk or loading the model into RAM. For the desired complexity threshold, the CDS can update the current focus level *m*, $\mathbf{PF}^{k,m}$ and probability box using eq. (5.6). Then, Layer II will send the discretized observables $\hat{\mathbf{Y}}_n^{k,m}$ and $\mathbf{PF}^{k,m}$ to Layer III (see. eqs. (2.2) and (5.6)).

$$\hat{\mathbf{Y}}_n^{k,m} = [\hat{Y}_{n-m}^k(PF_m^k), \hat{Y}_{n-m+1}^k(PF_{m-1}^k), \ldots, \hat{Y}_{n-f}^k(PF_{m-f}^k), ..., \hat{Y}_n^k(PF_0^k)]. \tag{5.7}$$

c) Layer III:

In this layer, the system estimates the probability of measured observables $\mathbf{Y}_n^{k,m} = [Y_{n-m}^k, Y_{n-m+1}^k, ..., Y_{n-f}^k, ..., Y_n^k]$ for a given X_n^k (usually, this is known from the database), i.e. $P(\mathbf{Y}_n^{k,m} \mid X_n^k)$ by approximating it as the probability of $\hat{\mathbf{Y}}_n^{k,m}$ (see eq. (5.7)) for a given $\hat{X}_n^k$, using the Monte-Carlo method. The extracted model in this layer can be called the *Bayesian model*. This model is saved in the model library for future use. The executive will use the extracted statistical model for prediction by receiving them from the internal feedback to evaluate the actions virtually before applying them on the environment.

Bayesian equation and maximum posteriori selection

The posterior can be calculated using the Bayesian equation as follows:

$$b_k(\hat{X}_n^k, \hat{\mathbf{Y}}_n^{k,m}) = P(\hat{X}_n^k \mid \hat{\mathbf{Y}}_n^{k,m}) = \frac{P(\hat{\mathbf{Y}}_n^{k,m} \mid \hat{X}_n^k) P(\hat{X}_n^k)}{P(\hat{\mathbf{Y}}_n^{k,m})}, \tag{5.8}$$

$$q_k(\hat{\mathbf{Y}}_n^{k,m}, \hat{X}_n^k) = P(\hat{\mathbf{Y}}_n^{k,m} \mid \hat{X}_n^k). \tag{5.9}$$

Here, k denotes the PAC number and n means the CDM at time n. The estimated *Bayesian model* $P(\hat{\mathbf{Y}}_n^{k,m} \mid \hat{X}_n^k)$ and the evidence $P(\hat{\mathbf{Y}}_n^{k,m})$ are extracted in *Layer* III. The probability $P(\hat{X}_n^k)$ can be estimated using the database.

Maximum posteriori selection and assurance factor

The CDS can save the estimated posterior $P(\hat{X}_n^k \mid \hat{\mathbf{Y}}_n^{k,m})$ in the perceptor library after a one-time calculation. We can use the posterior $P(\hat{X}_n^k \mid \hat{\mathbf{Y}}_n^{k,m})$ to select $\bar{X}_n^k$ that has the maximum probability in each discretized cell as:

$$\bar{X}_n^k = \arg\max_{s=1,2,\ldots,S} \{P(\hat{X}_n^k = X_s \mid \hat{\mathbf{Y}}_n^{k,m})\}, \tag{5.10}$$

where, $s = 1, 2, \ldots, S$ and S *is* the total number of decision-making situations (see section 2.1 and eq. (2.1)). Also, $b_k(\bar{X}_n^k, \hat{\mathbf{Y}}_n^{k,m}) = P(\bar{X}_n^k \mid \hat{\mathbf{Y}}_n^{k,m})$ can be saved in model library for steady-state mode or future prediction by perceptor.

5.2.2. Feedback channel, assurance factor and internal rewards

Similar to the definition in Chapters 3 and 4, we define the average assurance factor (AF) as:

$$AF_n^k = \frac{\sum_{o=n-L}^{n} P(\bar{X}_o^k \mid \hat{\mathbf{Y}}_o^{k,m})}{L}, \quad n - L \geq m \tag{5.11}$$

Here, L is an arbitrary time discrete interval to calculate the assurance factor. Also, we calculate the incremental of the assurance factor in the k^{th} PAC using:

$$\Delta_k^n = AF_n^k - AF_n^{k-1} \tag{5.12}$$

where the AF_n^{k-1} and AF_n^k are the assurance factors that can be calculated at the (*k*-1) and *k*th PAC, respectively. The internal reward, denoted as the rw_n^k, can be defined as the arbitrary function of AF_n^k and Δ_k^n from the function:

$$rw_n^k = f_k(AF_n^k, \Delta_k^n) \tag{5.13}$$

where the $f_k(\cdot)$ is an arbitrary function (See Chapter 4). The perceptor sends rw_n^k to the executive by the global feedback channel. In addition, the perceptor sends the estimated *Bayesian model* and *evidence* through the internal feedback channel (see Fig. 5.1). Also, the perceptor receives the modeling configurations such as $\mathbf{PF}^{k,m}$ (see eq. (4)) through internal feedforward channel that are internal commands from the executive. Here, PF_m^k is the precision factor at the *k*th PAC and focus level *m*. Therefore, the executive uses this information and it can find the proper action for applying on NGNLE. Also, for easy reference, we provide a list of the important notations and their meanings in Table 5.2.

Table 5.2: The important notation list used in this chapter [1.35]

Notations	Value
K	The total number of environmental PACs
k	The current PAC number (environmental)
M	The Maximum possible focus level
m	The current focus level
$F_m^{total,k}$	The total number of discretized cells at *k*th PAC and *m*th focus level
C	The set of all possible actions in actions library
X	The set of all discrete situations of NGNLE
S	Maximum number of the situations for CDM
L	The discrete time interval for *AF* calculation
X_n^k	The current NGNLE situation, at time *n*
$\hat{X}_n^k$	The training data from database using for modeling
$\bar{X}_n^k$	The estimated decision by the system

$Y_n^{k,m}$	The observable at time *n* and *k*th PAC
$\overline{Y}_n^{k,m}$	The normalized observables at time *n* and *k*th PAC
$\hat{Y}_n^{k,m}$	The discretized observables at time *n* and *k*th PAC
AF_n^k	The assurance factor at time *n* and *k*th PAC
rw_n^k	The internal reward at time *n* and *k*th PAC
$b_k(\hat{X}_n^k, \hat{\mathbf{Y}}_n^{k,m})$	The estimated posterior at time *n* and *k*th PAC
$q_k(\hat{X}_n^k, \hat{\mathbf{Y}}_n^{k,m})$	The estimated Bayesian model at time *n* and *k*th PAC
c_k	Current action applied on NGNLE
$Threshold(m)$	The predefined threshold at focus level *m*
T	The number of virtual actions can predict by current available model at time *n* and *k*th PAC
PF_m^k	The precision factor at *k*th PAC and focus level *m*
$b_k(\hat{X}_n^{(k+1),t}, \hat{\mathbf{Y}}_n^{(k+1),t,m})$	The *t*th prediction using posterior at *k*th PAC
$AF_n^{(k+1),t}$	The *t*th assurance factor prediction in executive
$rw_n^{(k+1),t}$	The *t*th internal reward prediction in executive
c_{k+1}^t	The *t*th virtual actions at time *n* for prospective (*k*+1)th PAC
c_{k+1}	The final action for applying in *(k+1)*th PAC

Now, we provide Algorithm 5.1 to compute environment actions through the global feedback. In this Algorithm, we include several heading comments, so that it is easy to follow.

Algorithm 1: Environmental actions through the global feedback

Input: the observables $\mathbf{Y}_n^{k,m}$, focus level *m*

Output: the c_{k+1} as the final action in environmental actions mode

Initialization:

Pre-adaptive actions:

Turn steady state mode off, environmental actions on

$c_1 \leftarrow$ *first enviromental action from C*

Apply c_1 to the environment (such as fiber optic channel)

Threshold(M), $\mathbf{PF}^{1,m}=[PF_0^1, PF_1^1, \cdots, PF_m^1]$

for k=1 to K **then**
    Take the observable $Y_n^{k,m}$
    if the model is not available **then**
    if k>1 **then**
        Predict $P(\hat{X}_n^k \mid \hat{\mathbf{Y}}_n^{k,m})$ using $q_{(k-1)}(\hat{X}_n^k, \hat{\mathbf{Y}}_n^{k,m})$
        Estimate $\bar{\mathbf{X}}_n^k$ by MAP rule
        End if
    Extract the $P(\hat{\mathbf{Y}}_n^{k,m} \mid \hat{X}_n^k)$ and $P(\hat{\mathbf{Y}}_n^{k,m})$
            Calculate the posterior $P(\hat{X}_n^k \mid \hat{\mathbf{Y}}_n^{k,m})$
            Estimate $\bar{X}_n^k$ by decision making
    Else if model is available **then**
    Load model, evidence and posterior from perceptor library
    Estimate $\bar{X}_n^k$ by MAP rule and save $b_k(\bar{X}_n^k, \hat{\mathbf{Y}}_n^{k,m}) = P(\bar{X}_n^k \mid \hat{\mathbf{Y}}_n^{k,m})$
    End if
    Send $\mathbf{PF}^{k,m}$, $P(\hat{\mathbf{Y}}_n^{k,m} \mid \hat{X}_n^k)$ and $P(\hat{\mathbf{Y}}_n^{k,m})$ to the Executive

Internal reward
    Calculate rw_n^k and send it to executive

Planning
    $rw_n^{(k+1),0} \leftarrow rw_n^k$
    Localize the set of all close actions to c_k

Learning
If actions belong to environmental set **then**
    for t=1 to T **then**
            Apply c_{k+1}^t **virtually (**$c_{k+1}^t \in C$**)**
            Calculate $b_k(X_n^{(k+1),t}, \hat{\mathbf{Y}}_n^{(k+1),m,t})$
            Estimate $X_n^{(k+1),t}$
            Calculate $rw_n^{(k+1),t}$
            if $rw_n^{(k+1),t} \le rw_n^{(k+1),t}$, and $AF_n^k \le Threshold(m)$ **then**
                $t_{opt} \leftarrow \arg\min_{t=0,1,2,\ldots,T}(rw_n^{(k+1),t})$
                $c_{k+1} \leftarrow c_{k+1}^{t_{opt}}$
                break for
            End if
        End for
        if $t' \ge 1$ **then**
            apply c_{k+1} to the environment
        else

Policy updating

34: Update $Threshold(m+1)$

Planning update

35: Switch actions type to *internal commands*
36: Update $\mathbf{PF}^{k,m+1}=[PF_0^k, PF_1^k, \cdots, PF_{m+1}^k]$
37: **if** $F_{m+1}^{total,k} \le complexity\ threshold$ **then**
38: $m \leftarrow m+1$
39: Switch actions type: environmental actions
40: **else**
41: Turn on steady-state mode
42: **End if**
43: **End if**
44: **End for**

5.2.3. Executive

The executive designed here has three parts (see Figure 5.1): planner (it consists of the actions library also), policy, and learning using prediction of the outcome of the virtual action [1.11] (see chapter 4 also). Next, we describe the planner and policy followed by learning using prediction.

1) Planner and policy

The planner extracts the set of prospective actions that are already saved in CDS actions library. Also, the planner selects the first action in the 1st PAC using pre-adaptive actions. In addition, the planner updates the actions type and sends internal commands to update $\mathbf{PF}^{k,m}$ (see eq. (5.2)) through the internal feedback and feedforward channels (i.e., the shunt cycle).

In this chapter, the policy determines the desired goals that the CDS should achieve using the PAC. For example, the goal can be the trade-off between the desired CDM accuracy and the computational cost. Here, for simplicity, we define the CDS goal as the $Threshold^m$ which means the CDS decision making accuracy goal in focus level *m* for the desired complexity threshold. For example, the $Threshold^{m=0}$ 1is 0 percent acceptable decision-making error. The $Threshold^{m=1}$ is 5 percent acceptable decision-making error, but the computational cost is increased due to modeling complexity.

2) Learning using prediction

a) Executive actions

The actions space in the executive can be classified as the environmental actions and internal CDS commands. Therefore, we define the actions space as follows:

$$C=\left\{ c_k \left| c_k = \begin{array}{c} Environmental\ actions, \text{or} \\ Internal\ commands = \begin{cases} \text{Increase}\, m, & m < M \\ \text{Decrease}\, m, & m \geq 1 \end{cases} \end{array} \right. \right\}, \tag{5.14}$$

where c_k is the action in the current *PAC* number k and $\boldsymbol{C}$ is set of all possible actions in the actions space.

The posteriori due to the virtual environmental action of c_{k+1}^{t}(see Table 5.2) can be predicted virtually using:

$$ratio^{k+1,t} = \max_{s=1,2,...,S} \{ \frac{std(\hat{Y}_n^{k,m} \mid \hat{X}_n^{k} = X_s)}{std(\hat{Y}_n^{k-1,m} \mid \hat{X}_n^{k-1} = X_s)} \}, \tag{5.15}$$

$$\hat{\mathbf{Y}}_n^{(k+1),t,m} = h\left(\hat{\mathbf{Y}}_n^{k,m},\ ratio^{k+1,t}, c_k, c_{k-1}, t, m \right), \tag{5.16}$$

$$\begin{aligned} & P(\hat{X}_n^{(k+1),t} \mid \hat{\mathbf{Y}}_n^{(k+1),t,m}) = \frac{P(\hat{\mathbf{Y}}_n^{(k+1),t,m} \mid \hat{X}_n^{(k+1),t}) P(\hat{X}_n^{(k+1),t})}{P(\hat{\mathbf{Y}}_n^{(k+1),t,m})}, \\ & or \\ & b_k(\hat{X}_n^{(k+1),t} \mid \hat{\mathbf{Y}}_n^{(k+1),t,m}) = P(\hat{X}_n^{(k+1),t} \mid \hat{\mathbf{Y}}_n^{(k+1),t,m}). \end{aligned} \tag{5.17}$$

Here, $t \in \{1, 2, ..., T\}$ and T is a total number of desired virtual actions that the kth model is still valid for predicting the T^{th} virtual action. Also, in eq. (5.17), $q_k(\cdot)$ and $b_k(\cdot)$ (see eqs. (5.8)-(5.9)) are received through the internal feedback from the perceptor. Therefore, the predicted assurance factor can be calculated as:

$$AF_n^{(k+1),t} = \frac{\sum_{o=n-L}^{n} P(\bar{X}_o^{(k+1),t} \mid \hat{\mathbf{Y}}_o^{(k+1),t,m})}{L}, \quad n - L \geq m, \tag{5.18}$$

and,

$$\Delta_{k+1,t}^{n} = AF_n^{k+1,t} - AF_n^{k+1,t-1}. \tag{5.19}$$

Then, the predicted internal rewards for the desired virtual action c_{k+1}^{t} can be calculated using eq. (5.13) as follows:

$$rw_n^{(k+1),t} = \begin{cases} f_{(k+1)}(AF_n^{(k+1),t},\ \Delta_{k+1,t}^{n}), & t \in \{1,2,...,T\} \\ rw_n^{k} & t = 0 \end{cases}. \tag{5.20}$$

Therefore, we can find the action $c_{k+1}^{t_{opt}}$ that brings the minimum internal reward as:

$$t_{opt} = \arg\min_{t \in \{0,1,2,..,T\}} (rw_n^{(k+1),t}). \tag{5.21}$$

As a result, the actions to be applied on the environment can be selected as:

$$c_{k+1} = c_{k+1}^{t_{opt}}, \qquad for\ t_{opt} \geq 1. \tag{5.22}$$

Therefore, c_{k+1} is the best action to be applied on the environment to improve the decision-making performance. Algorithm 5.1 shows the outline of the main processes for the global PAC of our CDS. In some applications, the policy may update the $Threshold^{m}$ for more accuracy (or even higher complexity with acceptable accuracy).

b) Internal commands

As mentioned before, some actions apply on the environment and the CDS receives the internal reward through the global PAC using the main feedback channel. However, the actions can be the internal commands such as the increase or decrease the focus level, which is a trade-off between the computational cost and CDM accuracy (see eqs. (5.2), (5.6) and (5.14)).

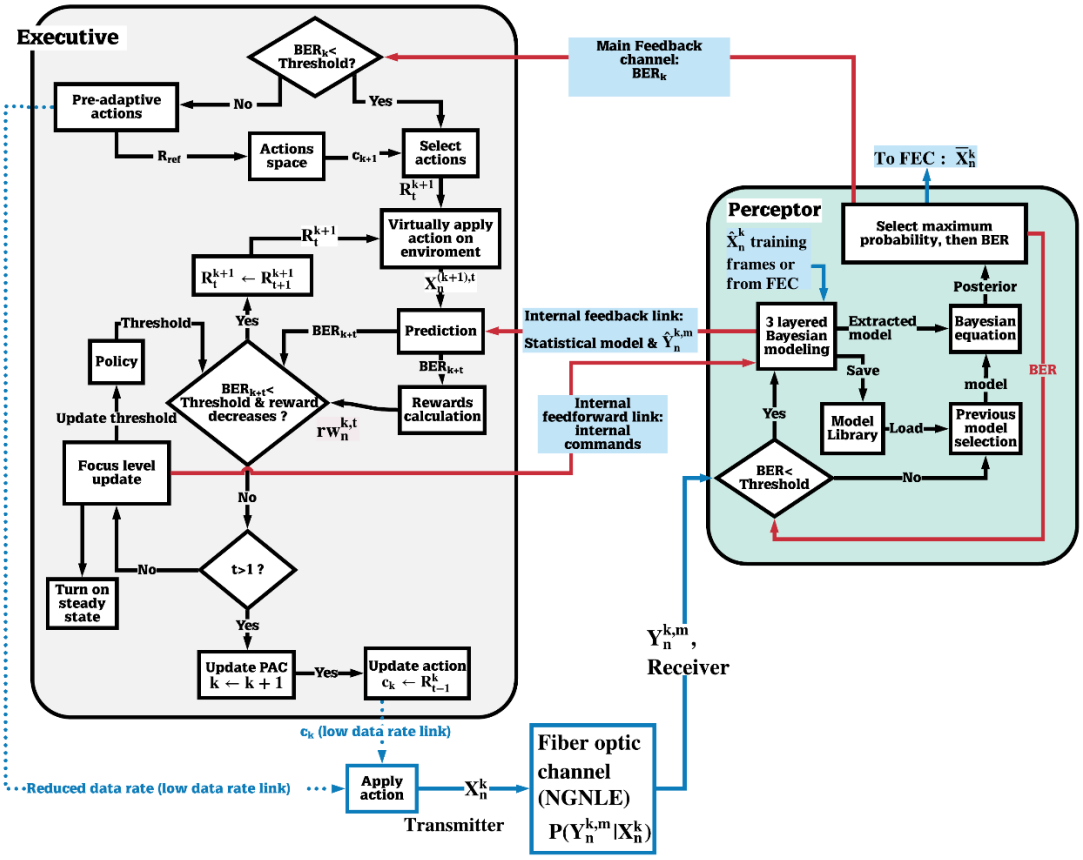


Figure 5.2: CDS architecture for long-haul NGNLE OFDM-based Fiber Optic link [1.35].

5.3. CDS Implementation Case Study: OFDM Long-haul Fiber Optic Communications

Figure 5.2 shows the CDS architecture block diagram for an OFDM long-haul fiber optic link. As mentioned before, we select the fiber optic link as an example of a NGNLE. The prospective internal reward can be calculated as follows (see eq. (5.20) and sections 3.4 and 4.3 also):

$$rw_n^{k,t} = \frac{BER_{k+t}^n}{\left(R_t^k - R_{ref}\right)^4}, \tag{5.23}$$

$$R_t^k = R_{ref} + t \times d, \tag{5.24}$$

where d is 4 Gb/s discretization step for data rate and R_{ref} is an available reference data rate that is set equal to $(R_1^1 - d)$. Also, the implementation of eqs. (5.15)-(5.17) for a fiber optic communication system is presented in [21, 23].

The actions space for this specific case study is (see eq. (5.14)):

$$C = \left\{ c_k \middle| c_k = \begin{array}{c} data\ rate\ R_k > R_{ref}, \text{or} \\ Internal\ commands \Rightarrow \begin{cases} m = 1, \\ m = 0 \end{cases} \end{array} \right\} \tag{5.23}$$

The executive performs actions such as change the data rate or change the focus levels.

5.3.1. Simulations parameters

We illustrate the modeling process by performing a numerical simulation of an OFDM system (see Figure 5.2). The description of the simulation set up is the same as that in Chapter 4 and for convenient are presented in Table 5.3 again. The CDS parameters are as follows: $PF_{i=0}^{k} = 2$, $PF_{i=1}^{k} = 5$, and 96% probability box. Also, the $Threshold^{m=0}$ which is the FEC threshold in focus level 0 is 1.03×10^{-2} (with 14.3% Overhead (OH) for the hard decision (HD)) [3.3].

Table 5.3: The numerical simulation parameters of the OFDM fiber optic communications system and CDS [1.35]

Simulation parameters	Value
Fiber dispersion coefficient (β_2)	$-22.1\ \mathrm{ps}^2/\mathrm{km}$
Nonlinear coefficient (γ)	$1.1\ \mathrm{W}^{-1}\mathrm{km}^{-1}$
Fiber loss coefficient (α)	$0.2\ \mathrm{dB/km}$
Number of fiber spans (N)	20
Span length (L_{span})	80 km (1600 km total length)
Noise figure (NF)	4.77 dB
Number of OFDM subcarriers per frame	256
Subcarrier modulation	QAM-16
Number of data frames	128,000
Number of frames for modeling	100,000

Number of frames for validation	28000
Discretization step *d*	4 Gb/s
$PF_{i=0}^{k}$	2
$PF_{i=1}^{k}$	5
Complexity threshold	10^6 discretized cells
$\max(F_{m=0}^{total,k}), k=1,2,3,4$	3600 discretized cells
$\max(F_{m=1}^{total,k}), k=5,6,7,8,9$	518,400 discretized cells
$F_{m=2}^{total,k}$, $k=10$	74,649,600 discretized cells
$Threshold^{m=0}$ or FEC threshold	1.03×10^{-2}
$Threshold^{m=1}$ or FEC threshold	4.7×10^{-3}
Fiber type	Standard single mode fiber (SMF)

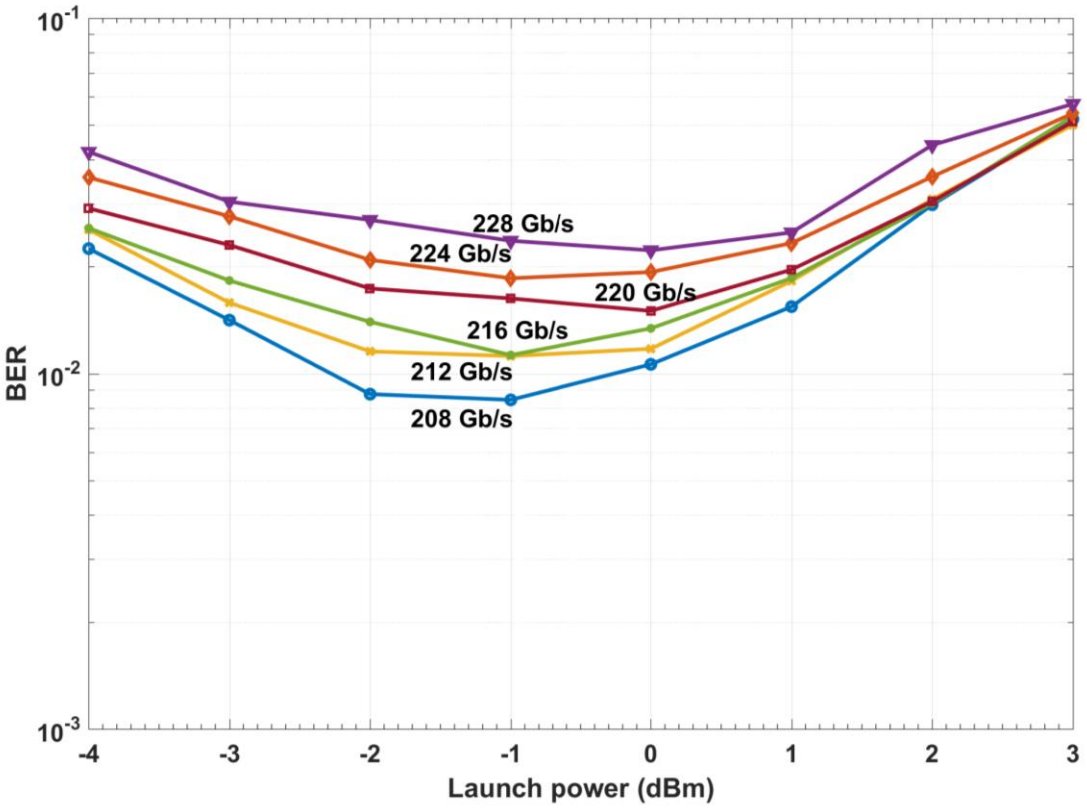


Figure 5.3: BER for a conventional system with a linear equalizer [1.35].

5.3.2. Simulation results and discussions

1) Pre adaptive actions

The BER of a conventional fiber optic communications system (i.e., without CDS, see Figure 2.6) is shown in Figure 5.3. We consider 1.03×10^{-2} (with 14.3% overhead (OH)) as the FEC threshold or *Threshold(m=0)* for HD [3.3] (see Table 5.3). Thus, we can see from Figure 5.3 that the maximum achievable data rate is 208 Gb/s. The CDS uses 208 Gb/s for $j=1$ (i.e. first data rate, $R_1^1=208$ *Gb/s)*. Also, we assume data rate variation by discretization step d=4 Gb/sand a step of 1 dB for the launch power. Therefore, the data rate for PAC $k=0$ is $R_{ref}= 204$ *Gb/s* (see eq. (5.24)). The optimum launch power can be also found using the pre-adaptive method this is already used in the conventional system.

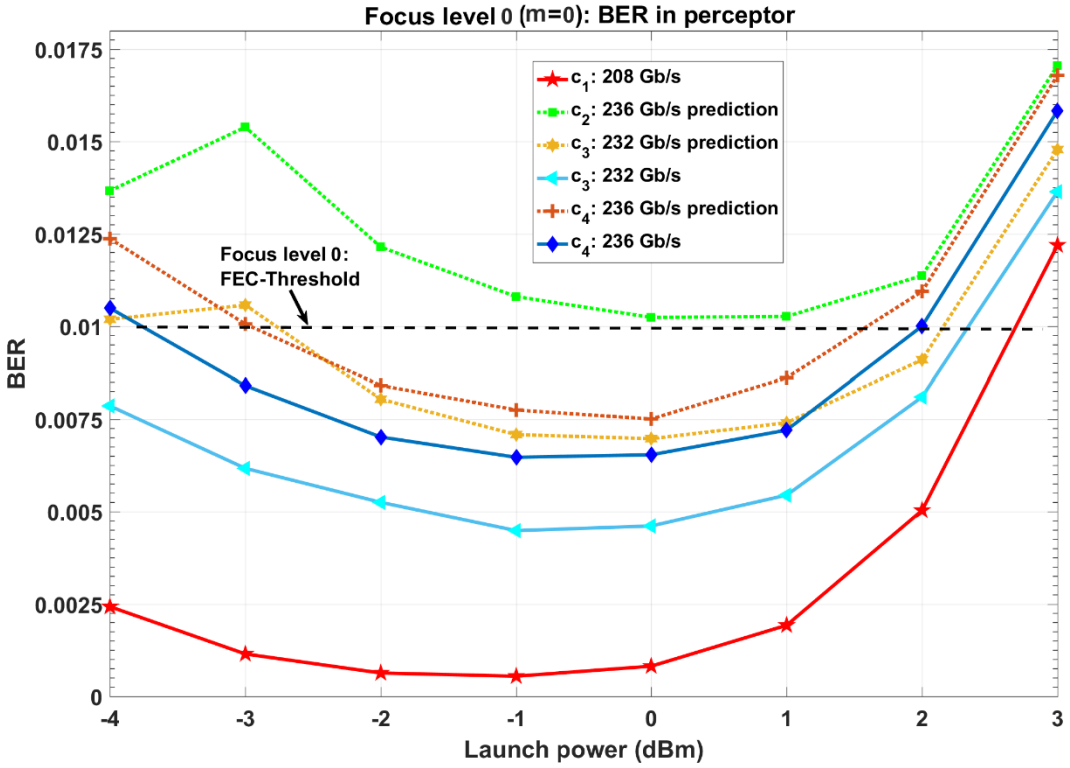
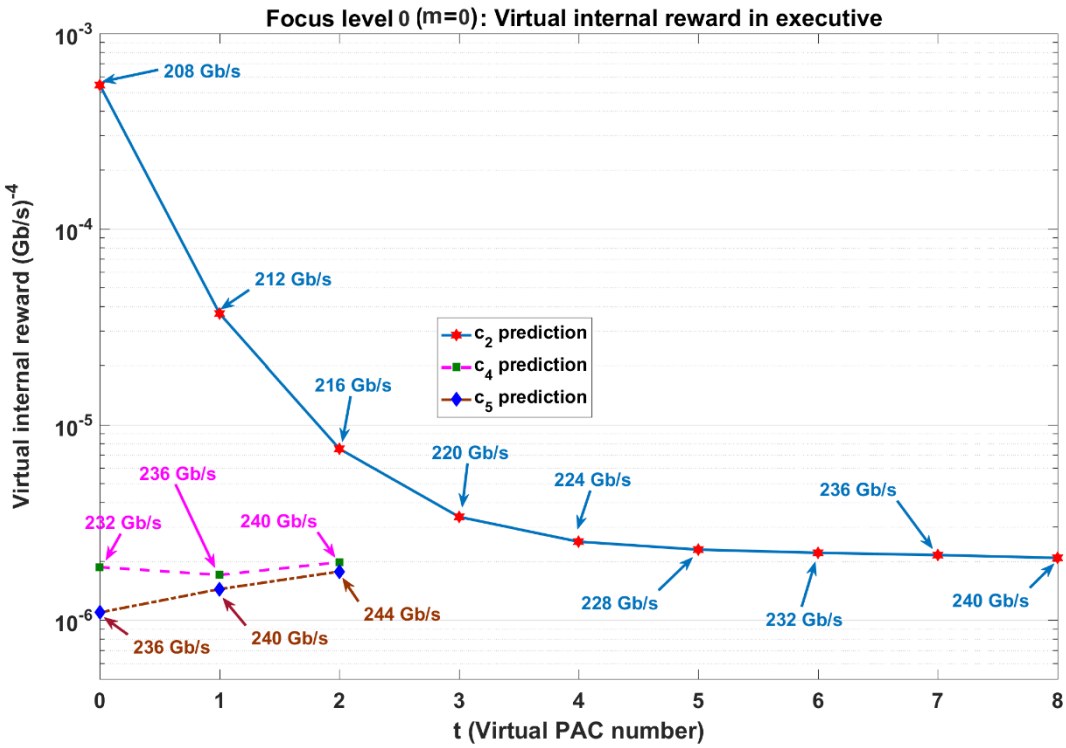


Figure 5.4: BER in perceptor after environmental actions (*PF=2, d=4 Gb/s*, focus level 0) [1.35].

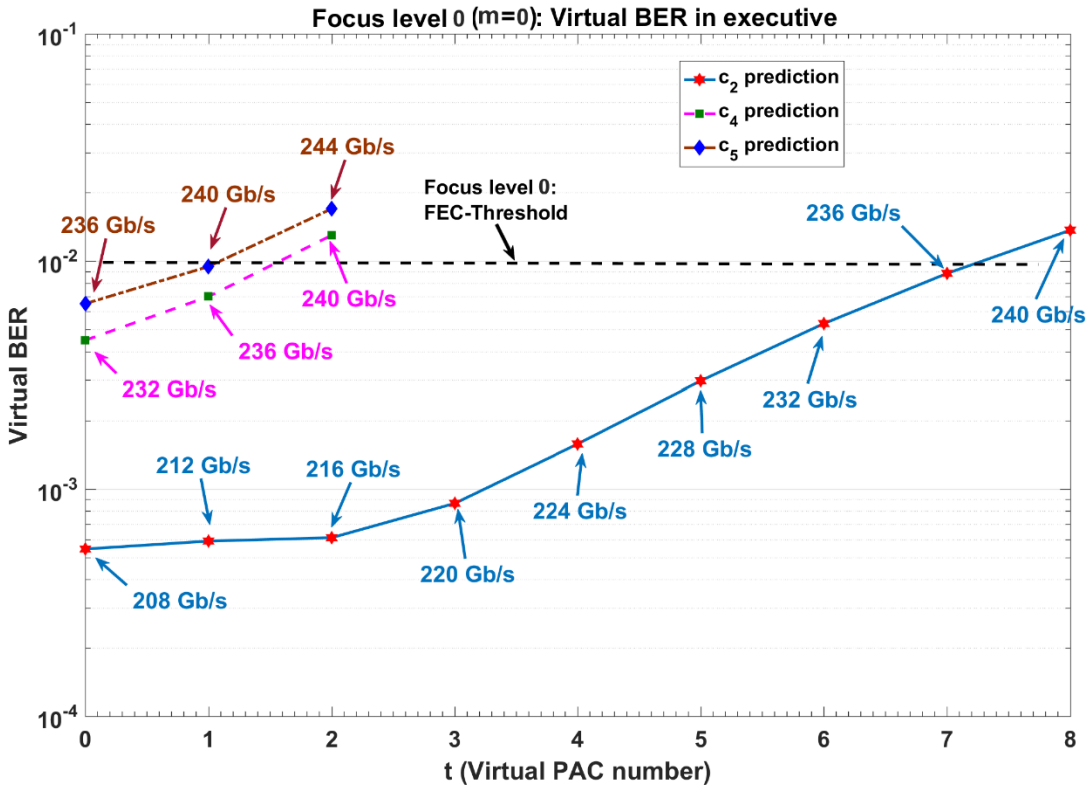
2) Results for the system with the CDS focus level 1

The global PAC procedure can be summarized using the results shown in Figures 5.2 to 5.8 as follows (Note: PAC numbers 1-4 are exactly similar to PAC numbers 1-4 are presented in chapter 4):

1. 1st PAC (focus level 0): CDS reduces the date rate to find the first data rate for which the BER is under the FEC-threshold (see Figure 5.3). Therefore, the data rate 208 Gb/s is applied to the fiber optic link as the first action (c_1). Because the BER is under the FEC threshold, the perceptor can extract the statistical model for 208 Gb/s. The perceptor can improve the BER using the extracted model (see Figure 5.3, c_1:208 Gb/s). Note that the minimum BER for a conventional system at 208 Gb/s is 8.0×10^{-3} (see Figure 5.3) whereas this is reduced to 5.5×10^{-4} when the CDS is used (see Figure 5.4, c_1:208 Gb/s). The perceptor sends the extracted model to the executive using the internal feedback link.



(a)



(b)

Figure 5.5: BER using CDS in executive prediction for virtual actions (*PF*=2, *d*=4 Gb/s, focus level 0) (a) learning curve for virtual internal reward, (b) virtual BER vs executive internal cycle) [1.35].

2. 2nd PAC (focus level 0): The executive predicts the virtual internal rewards and BER for the virtual action of c_2 (see Figures 5.2 and 5.5 as well as Algorithm 5.1). Then, the executive applies 236 Gb/s as the c_2, because the virtual BER for 240 Gb/s at t=8 is more than the FEC-threshold of 10^{-2} (see Figure 5.5b, c_2: prediction). Also, 236 Gb/s has the minimum virtual internal rewards (See Figure 5.5a, c_2: prediction) and at this data rate, the BER is under FEC threshold (See Figure 5.5b, c_2: prediction). After applying c_2 by the CDS (i.e., transmitter sends the data at 236 Gb/s), the perceptor uses the 208 Gb/s model (see Figure 5.2 and Algorithm 5.1, previous model selection) to predict the BER for 236 Gb/s. However, the BER is above the FEC-threshold (see Figure 5.4, c_2: 236 Gb/s prediction, topmost curve). Note that we used the previous model selection in this case study. By using

previous model selection, the system does not need to transmit the training frames for model extraction.

3. 3rd PAC (focus level 0): The BER is more than the FEC-threshold for c_2. Therefore, the executive reduces the data rate by $d = 4$ *Gb/s* (see Figure 5.5). As a result, executive applies 232 Gb/s as c_3 in 3rd PAC number. Then, the perceptor uses 208 Gb/s model for BER estimation of 232 Gb/s (See Figure 5.4, c_3: 232 Gb/s prediction) and, it can bring the BER for 232 Gb/s under the FEC-threshold. Therefore, the perceptor can extract the model for 232 Gb/s. The extracted model results in further BER improvement for 232 Gb/s (see Figure 5.4, c_3: 232 Gb/s).

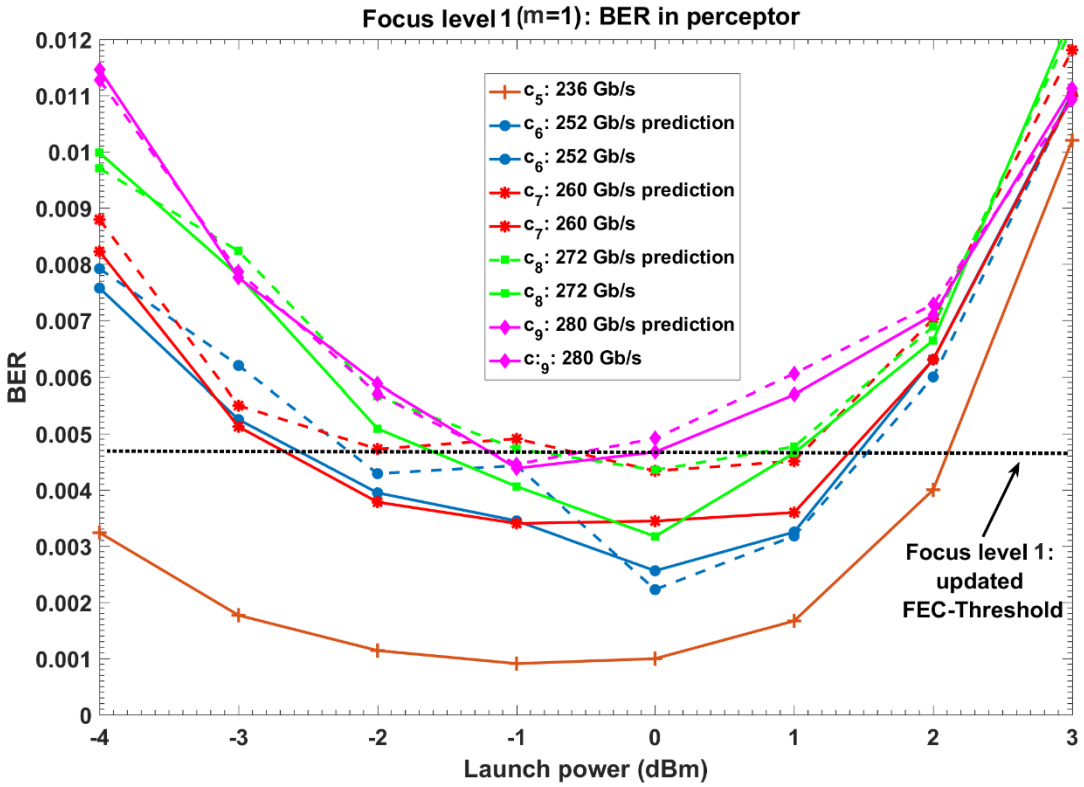


Figure 5.6: BER in perceptor after environmental actions (*PF*=2, *d*=4 Gb/s, focus level 1). Dash lines: BER prediction in perceptor and continuous lines: BER using exact model [1.35].

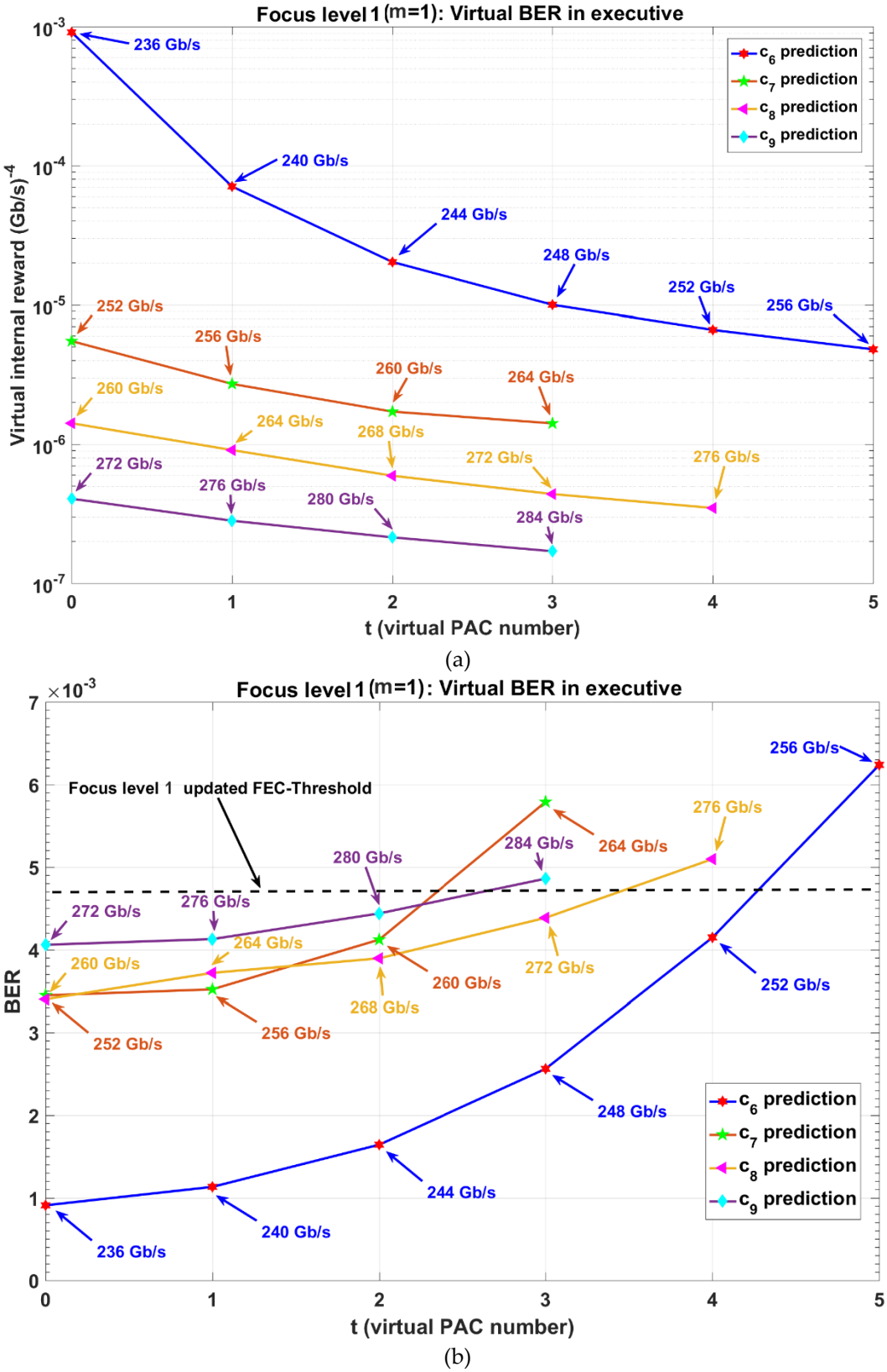


Figure 5.7: BER using CDS in executive prediction for virtual actions (*PF=2, d=4 Gb/s*, focus level 1) (a) learning curve for virtual internal reward, (b) virtual BER vs executive internal cycle) [1.35].

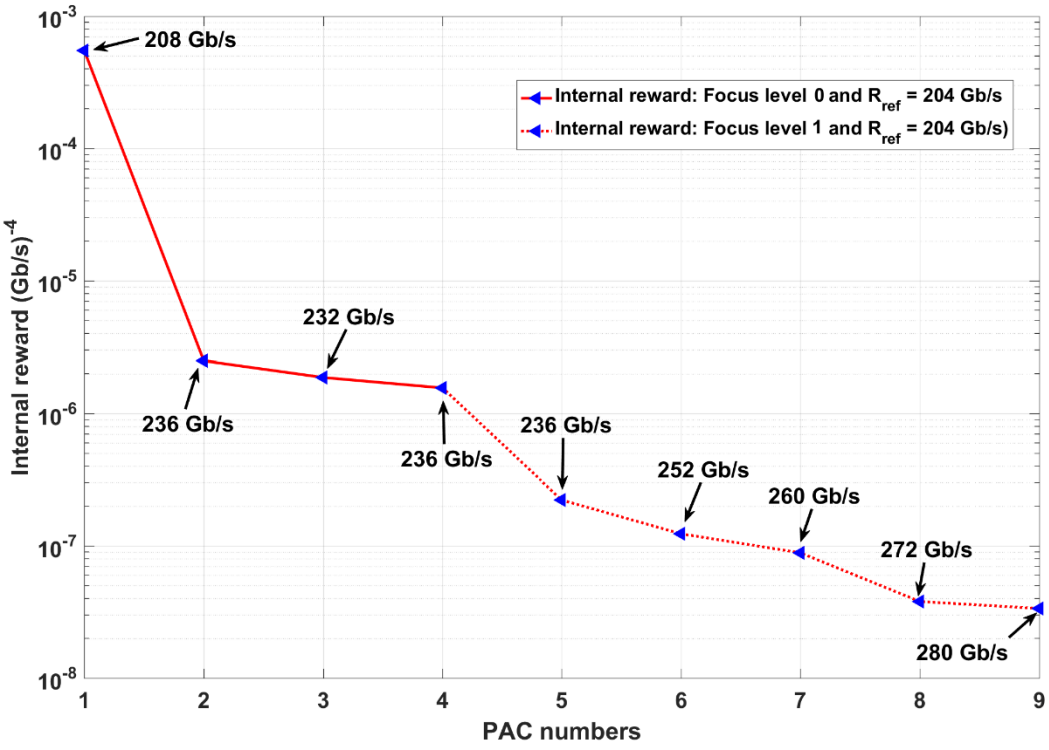
4. 4th PAC (focus level 0): The executive applies the 236 Gb/s (See Figure 5.5, c_4: prediction) as c_4 to the environment (i.e., fiber optic link) because the predicted BER for 240 Gb/s is more than the FEC-threshold (see Figure 5.5b, c_4: prediction) and virtual internal reward increases for 240 Gb/s (see Figure 5.5a, c_4: prediction). Figure 5.4 (c_4: 236 Gb/s prediction) shows that the perceptor provides the BER for 236 Gb/s under FEC-threshold using the model for 232 Gb/s (provided that the model for 208 Gb/s was used and BER was higher than FEC threshold in 2nd PAC). Like the 3rd PAC, the BER can improve further using the exact extracted model (see Figure 5.4, c_4: 236 Gb/s).
5. 5th PAC (focus level 0 and focus level 1): The executive finds that the predicted BER for 240 Gb/s is just less than the FEC threshold (see Figure 5.5b, c_5: prediction), but the virtual internal reward increases in comparison to 236 Gb/s (see Figure 5.5a, c_5: prediction). Therefore, the executive updates policy and FEC threshold or $Threshold^{m=1} = 4.7\times10^{-3}$. Then, the executive sends the new policy ($Threshold^{m=1} = 4.7\times10^{-3}$) and the internal command (increasing focus level) to the perceptor. Using the model extracted in focus level 1, the perceptor provides the BER under the updated FEC threshold (see Figure 5.6, c_5: 236 Gb/s and Figure 5.8b). Also, the internal rewards decreased correspondingly (see Figure 5.8a).
6. 6th PAC (focus level 1): The executive predicts the virtual internal rewards and the BER for the virtual action of c_6 using extended model in focus level 1 (see Figures 5.6, 5.7, and Algorithm 5.1). The executive applies 252 Gb/s as the c_6, because the virtual BER for 256 Gb/s at t=5 is above the updated FEC-threshold (see Figure 5.7b, c_6: prediction) and the action 252 Gb/s has the minimum virtual internal rewards (See Figure 5.7a, c_6: prediction) with the virtual BER under the updated FEC threshold (See Figure 5.7b, c_6: prediction). After applying c_6 by the CDS, the perceptor uses 236 Gb/s extracted model (see Figure 5.6, c_6: 252 Gb/s prediction) to predict the BER for 252 Gb/s. Figure 5.6 shows that the perceptor provides the BER for 252 Gb/s under the updated FEC-threshold using the extracted model for 236 Gb/s. Similar to the previous PACs, the BER can be improved further using the exact extracted model (see Figure 5.6, c_6: 252 Gb/s).

7. 7^{th}, 8^{th} and 9^{th} PAC (focus level 1): Similar to 6^{th} PAC, the CDS can provide 260 Gb/s, 272 Gb/s, and 280 Gb/s for the c_7, c_8, and c_9, respectively.
8. 10^{th} PAC (focus level 1): The executive finds that the predicted BER for 284 Gb/s is above the updated FEC threshold (see Figure 7b). As a result, the executive turns on steady state for 280 Gb/s. The CDS does not increase the focus level, because the number of cells in the focus level 2 is more than the desired complexity threshold (see Table 5.3 for $F_{m=2}^{total,k=10}$).

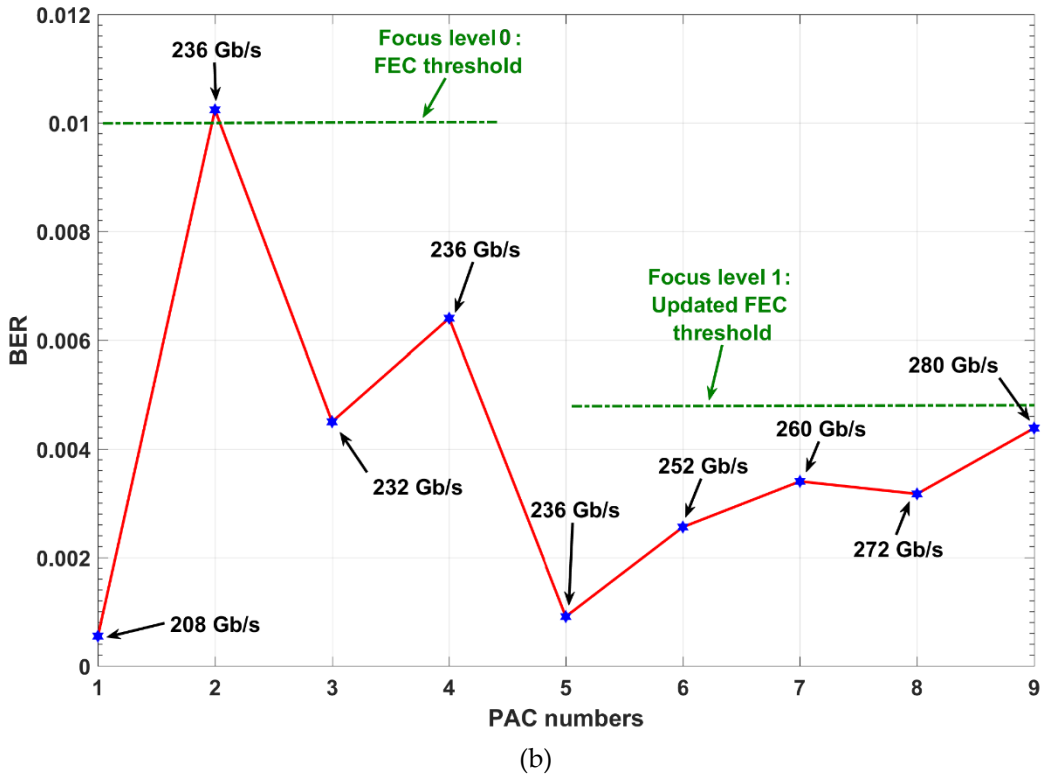
From Figures. 5.6 and 5.8b, we see that the CDS provides the minimum BER of 4.7×10^{-3} before HD-FEC, when the data rate is 280 Gb/s (see Figures 5.6 and 5.8b). Based on [3.3], the required OH can be reduced to 6.25%. As a result, an extra ~24 Gb/s in 280 Gb/s data rate can be used for data transmission instead of assigning it for overhead. It may be noted that the highest data rate that can be obtained using only the focus level = 0 (which corresponds to the results of Chapter 4) is 236 Gb/s, which is increased to 280 Gb/s.

5.3.3. CDS learning curve

The CDS learning curve is shown in Figure 5.8a and corresponding BER is shown in Figure 5.8b. In Figure 5.8, the PACs 1,2,3 and 4 correspond to focus level 0 actions which correspond to the data rate of 208, 236, 232 and, 236 Gb/s respectively (see Algorithm 5.1). The PACs 5, 6, 7, 8 and 9 are the CDS focus level 1 actions as data rates of 236, 252, 260, 272 and, 280 Gb/s, respectively.



(a)



(b)

Figure 5.8: Results for the CDS versus PAC numbers, (a) CDS learning curve, (b) BER vs PAC numbers [1.35].

The results for the CDS at steady state are summarized in Table 5.4. The smart fiber optic link that used the focus level concept enhances the data rate effieciency by ~43% in comparison to conventional fiber optic link. This is the summation of ~35% data rate enhancement (i.e., enhancement of the data rate R_1 = 208 Gb/s to R_9 = 280 Gb/s) and 8 % of HD overhead reduction (~24 Gb/s can be used for data carrying, instead of overhead redundancy) reduction, which is achieved by lowering the FEC threshold from 1.02×10^{-2} to 4.7×10^{-3}. The final Q-factor improvement is 7 dB, which is a significant improvement for the long-haul fiber optic link in comparison to the conventional OFDM system simulations at 280 Gb/s. The reason for such significant improvement is that the CDS using memory and the focus level concept can model the fiber optic channel more accurately in comparison to modeling the fiber optic link channel as memoryless. As the OFDM symbol propagates down a dispersive and nonlinear fiber, it interferes with the neighboring symbols both linearly and nonlinearly. The linear interference is removed easily by using the linear equalizers and the performance improvement brought about by the CDS with enhanced focus level is mainly due to its ability to make a smart decision in the presence of nonlinear interference. Under the same conditions, the performance improvement using the digital backpropagation (DBP) method is typically 1-2 dB [1.40] and the computational cost and complexity is significantly higher compared to our proposed system. In addition, the DBP method does not provide data rate enhancement.

Table 5.4: Results and the CDS parameters at steady state [1.35]

Results or CDS parameters	Value
Data rate (k=9)	280 Gb/s
$PF_{i=0}^{k=1,2,\cdots,9}$	2
$PF_{i=1}^{k=5,6,\cdots,9}$	5
BER (k =9)	4.4×10^{-3}
Q-factor improvement (k =9)	6.98 dB

5.3.4. CDS Complexity for proof-concept case study on long-haul OFDM fiber optic link

In order to calculate the computational cost, we distinguish two types of CDS modes: (i) Modeling extraction mode and (ii) Steady-state mode. The complexity of a CDS varies depending on the perceptor mode. When the CDS operates in the BER improvement mode, the BER converges after N OFDM frames (in simulation N = 28,000) and each frame has D data-carrying subcarriers (From Table 5.3, it is 126). To create the histogram (i.e., to calculate $P(\bar{X}_n \mid \hat{\mathbf{Y}}_n^m)$) based on the discretized data, we need $N \times D$ simple additions (just counting the samples within the tiny computational cost). To calculate $P(\bar{X}_n \mid \hat{\mathbf{Y}}_n^m)$, we need S real multiplications where S is the order of QAM (in our example, S = 16), then, we save the maximum probability for the related QAM alphabet. For simplicity, we assume that the minimum and maximum of the probability box in all focus levels are $y_{\min}^{k,m} = x_{\min}^{k,m} = -6$ and $y_{\max}^{k,m} = x_{\max}^{k,m} = 6$, respectively for this proof-of-concept example. As a result, the storage requirement depends on the precision factor. For example, with a $\mathbf{PF}^{k,m=0}$ of 2 in focus level 0, we need to store a maximum posteriori in each cell as matrix of dimension $N_x^{k,0} \times N_y^{k,0} = 60 \times 60$ (see, eqs. (6) and (7), $\max(F_{m=0}^{total,k}) = 3600$). Also, the storage requirement in focus level 1 with a $\mathbf{PF}^{k,m=0}$ of 2 and a $\mathbf{PF}^{k,m=1}$ of 5, we need to store maximum posteriori in each cell as the matrix which has a dimension of $N_x^{k,1} \times N_y^{k,1} \times N_x^{k,0} \times N_y^{k,0} = 12 \times 12 \times 60 \times 60$ (see, eqs. (6) and (7), $\max(F_{m=1}^{total,k}) = 518,400$). These matrices can be stored in a hard disk and can be loaded into RAM when needed. Therefore, the complexity threshold (here, it is 10^6) is the predefined maximum available storage for saving in a hard disk or loading into RAM, when needed. When the CDS operates in the steady-state mode, there is no cost associated with model extraction (i.e., to calculate $P(\bar{X}_n \mid \mathbf{Y}_n^m)$ or to calculate eq. (5.10)). Therefore, no multiplication or addition is needed in steady state.

For validating, we need to process 28,000 frames (see Table 5.3) using digital back propagation (DBP), and this cannot be done using typical processors, while CDS can process 28000 frames as one block. Therefore, we assumed the DBP processing the 28,000 frames as ~55 blocks of 512 frames. The computational cost associated with

DBP for processing 512 OFDM frames is as follows. In each propagation step, $J\log_2(J)$ complex multiplications are required where J is the number of samples in time domain. If there are B propagation steps, the complex multiplication is $O(B\times J\log_2 J)$ [2.14][2.15]. For example, for a 20-span system, assuming 10 steps per span and with J = 16,384, we need 45,875,200 complex multiplications! Besides, this computation needs to be done continuously for each block of data. Three complex vectors of length J (the total size of 98,304) needs to be stored. In contrast, most of the computational complexity of the CDS is present only when there is a fluctuation in the fiber optic channel. Under steady-state, the computational cost is minimal.

It is useful to compare the complexity of the proposed method in this chapter with the DBP method for Q-factor enhancement using the simulation parameters in Table 5.3 and processing 28,000 OFDM frames. We run both algorithms on a Microsoft Surface-Pro with Intel ® Core ™ i706650U CPU @ 2.20GHz 2.21 GHz, 16 GB RAM, system type 64-bit Operating System x64-Based processor using MATLAB. For the DBP algorithm [3.4], the runtime is 2465.8 seconds, that is ~41 minutes and 6 seconds. However, this runtime for 9 PACs with the proposed CDS is 792 seconds (~13 minutes and 12 seconds). In addition, the CDS running time is reduced to 100 seconds for the CDM in steady-state mode. In contrast, the DBP run time will always remain similar for any continuous data stream. We should mention that the CDS can process all 28,000 frames as one block. However, processing 28,000 frames using DBP method is not feasible using Microsoft Surface-Pro with Intel ® Core ™ i706650U CPU @ 2.20GHz 2.21 GHz, 16 GB RAM. Therefore, we break the 28,000 frames into ~55 blocks and, the DBP run time of ~41 minutes was achieved in this way. However, we should mention that processing received 28,000 frames in less than 100 seconds in the steady-state mode makes it a practical solution for almost real-time high-quality data transmission such as real-time high-quality video conferencing.

5.4. Conclusions

In recent years, some efforts were made to develop intelligent machines using cognitive dynamic system (CDS) for artificial intelligence (AI) applications. However, currently,

CDS high complexity algorithms available are only applied to LGE. These algorithms cannot be applied for a non-Gaussian and nonlinear environment (NGNLE) with finite memory. Therefore, we proposed novel CDS algorithms that could be applied on a NGNLE with finite memory. We redesigned the CDS in detail to make it suitable for NGNLE cognitive decision making (CDM) applications. Also, the system architecture and algorithms are presented. The proposed algorithm is applied on a long-haul nonlinear fiber-optic link as a case study. By upgrading the conventional link with the CDS, the applicable data-rate increased from 208 Gb/s to 280 Gb/s. Also, the CDS Q-factor improved by ~7 dB (the optimum BER = 4.4×10^{-3} for 280 Gb/s with CDS). The data rate enhancement is ~35% without considering the HD overhead reduction. However, the CDS brings the BER under the new FEC threshold (= 4.7×10^{-3}), we do not need 8% of the HD-FEC overhead. Hence, the net data rate enhancement is ~43%. Finally, the presented algorithms in this chapter are simple and fast because it used the Bayesian equation for calculating the posterior that has low complexity for the calculation. Therefore, the low complexity algorithms makes the proposed CDS a perfect candidate for the NGNLE applications such as long-haul fiber optic links or in healthcare.

Chapter 6

CDS V2 FOR REAL-TIME HEALTH SITUATION UNDERSTANDING IN SMART E-HEALTH HOME APPLICATIONS[4, 5]

The autonomic computing layer of the smart e-Health home based on a cognitive dynamic system (CDS) can be a solution for improving health situation understanding, reducing the healthcare system costs, and improving people's quality of life. It can also be a solution for reducing the large number of sudden deaths outside of a hospital due to fatal diseases such as Arrhythmia. Towards this objective, we start from understanding the health situation, by diagnosing healthy and unhealthy persons. For this, we developed a decision-making system that is inspired by the medical doctors (MDs) decision-making processes. Our system is based on a CDS for cognitive decision-making and it can create a decision-making tree automatically. The simple, low complexity algorithmic design of the proposed system makes it suitable for real-time applications. A proof-of-concept case study of the implementation of the CDS was done on Arrhythmia disease. An accuracy of 95.4% was achieved using the proposed algorithms. Also, these algorithms can make a decision in less than 80 ms, and for one User, this includes the time for training. The proposed platform can be extended for more healthcare applications such as screening, disease class diagnosis, prevention, treatment, or monitoring healing. As a result, the proposed CDS algorithms can be an

[4] The most of this chapter published as: M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain-Inspired Intelligence for Real-Time Health Situation Understanding in Smart e-Health Home Applications," *IEEE Access*, vol. 7, pp. 180106–180126, 2019.

[5] All symbols and notations are defined in this chapter are valid just for this chapter. The similar notations or symbols in other chapters have defined inside the chapters.

example of the first step for designing the autonomic computing layer of a smart e-Health home platform.

This chapter is organized as follows. In section 6.1, the introduction on CDS v2 for automatic diagnostic test is provided. Then, section 6.2, the architectural structure of CDS for the application of ADMS for NGNLHE and the proposed algorithms for zero false alarm health condition diagnosis are discussed. In Section 6.3, we discuss the simulation results for the proof-of-concept case study for Arrhythmia diagnosis. In Section 6.4, we present the research challenges and future work. Finally, the conclusions are presented in Section 6.5.

6.1. CDS v2

In this chapter, CDS v2 is presented for real-time health situation understanding for smart e-Health home applications. CDS v2 uses a perceptor that can extract the model using the adaptive decision-making tree. However, the executive of CDS cannot predict the outcome of actions using virtual actions before applying them on a NGNLE.

In chapters 3-5, a CDS was proposed for smart fiber optic communication systems as an example of using CDS for decision making in complex smart systems. It should be mentioned that we used the term cognitive decision making (CDM) for decision making using CDS. Also, complex smart e-Health systems such as smart e-Health home are non-Gaussian and nonlinear health environments (NGNLHE). Here, NGNLHE means that outputs of the healthcare system are not linearly dependent on inputs. Also, the outputs of healthcare systems do not have Gaussian distributions.

The CDS was presented for a smart home in [1.24] as an enhanced-AI. While a CDS block diagram was presented, no detailed algorithms neither discussed nor provided. It should be noted that, the smart home is different from a smart e-Health home in terms of applications. Therefore, in this chapter, we propose the detailed algorithms of a CDS for diagnostic tests in a smart e-Health home and CDS is used as the subsystem of ADMS for semi-medical diagnostic applications with a policy for low false alarm rates.

The main contributions of this work can be summarized as follows:

1. The concept of CDS is used as an alternative to AI in ADMS and for the non-Gaussian and non-linear health environment (NGNLHE). The inspiration of CDS from natural human intelligence brings the advantage to simulate a medical doctor (MD) decision-making approach with less complexity.
2. The architectural structure of an ADMS is designed as the first step of the health situation understanding and diagnosis. In this chapter, we focus on the diagnostic tests between a healthy and unhealthy situation using ADMS towards the goal of low false alarm policy.
3. Algorithms for decision making between healthy and unhealthy situations are presented. These algorithms are designed based on creating a decision-making tree in the perceptor. Also, the algorithms use a semi-medical doctors' one-dimensional decision-making approach. Therefore, the CDS decision making is based on only one feature.
4. A proof-of-concept case study in which a virtual CDS-based ADMS is applied to Arrhythmia diagnosis is performed. It is shown that this new design performs with 95.4% accuracy. This accuracy is achieved even with missing measured information for some features such as the heart rate of some Users. These missing features can be considered as sensors' failures. In Table 1.3, a comparison between the proposed method and some related works on this Arrhythmia database decision-making accuracy is shown.

6.2. Proposed ADMS using CDS architecture and algorithms

In this section, we describe the proposed CDS architecture and algorithms that can be applied for decision-making purposes in NGNHLE. The detailed architecture of the proposed CDS for health situation decision making is shown in Figure 6.1. The CDS has two main subsystems: (i) the perceptor, and (ii) the executive with a feedback channel linking them. Through interactions with the Users (environment) and e-Health network, a PAC is created in the form of the training and prediction modes, using local feedback loops and a global feedback loop, respectively.

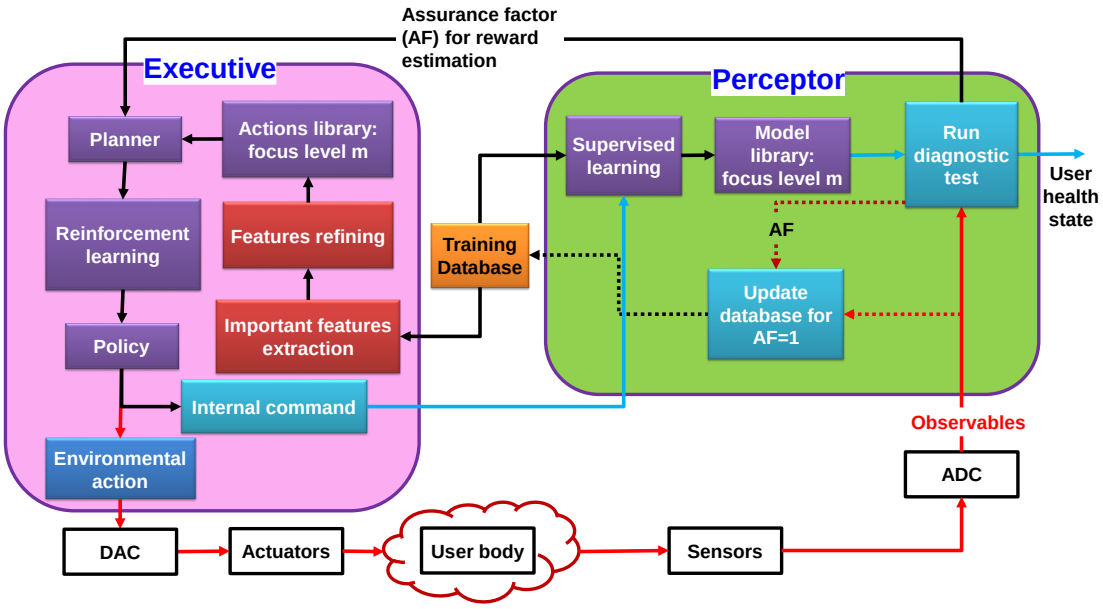


Figure 6.1: Block diagram of proposed CDS architecture for the ADMS of a smart e-Health home [1.37].

First, we describe the perceptor and executive training modes using related algorithms. Next, we discuss the decision-making tree, the feedback channel and internal rewards in training mode or prediction mode for both the perceptor and the executive. The CDS operates in three modes: (i) training (ii) prediction and (iii) steady state. The CDS functions will be different depending on its mode. The steady-state mode means that nothing seems abnormal, or there is no User request.

In Fig. 5.1, the training database can be updated dynamically in three ways:

1. The data and information collected by the sensors for Users with a known health state (Users with a known health state, or after 100% assurance decision making depending on the policy and focus level).
2. The database can be updated through e-Health and healthcare networks.
3. The smart e-Health home can be a node in a network of smart e-Health homes.

The data and information collected in the database will be converted to the model library in the perceptor and actions library in the executive after running the CDS in its training mode. The perceptor and executive will gain more knowledge dynamically after the training database is updated (see Fig. 5.1). The CDS will use the knowledge in the perceptor model library for prediction of the health conditions of Users with unknown health states.

For specific situations or on request by the Users such as something abnormal in the home or crying out with pain, needing help or the User saying directly, “I am not feeling well,” then the CDS can initiate the prediction mode. The knowledge is used to reason and predict the health situation of the User, and then to determine whether any action is needed based on the policy and objectives. Here, the policy is low false alarm in diagnosing between healthy and unhealthy conditions. Knowledge in the perceptor is represented as a set of concepts and the relationships among them, together with the User's health situations.

Therefore, in a specific situation or on a User’s request, the CDS will switch from steady-state to the prediction mode state, and the fuzzy nature of this process is presented in Section 6.2.1 below. Also, the CDS in the training mode dynamically updates the knowledge in the perceptor and actions library in the executive after the database is updated in the three ways discussed above. This update can be done in real-time and even in parallel to the prediction mode or during the steady-state mode. In Sections 6.2.2 to 6.2.4, the algorithmic description for the training and prediction modes are described. Finally, in section 6.2.5, the complexity of the presented algorithms are discussed.

6.2.1. Fuzzy nature of the proposed cognitive dynamic system for smart e-Health home applications

For real-time healthcare applications in this chapter, the algorithms should be simple, straightforward and fast. Thus, we would like to avoid using extra logarithms, complicated functions and integrals required in the entropic state calculation provided by generic CDS [1.8]. Here, the internal reward is inspired by the fuzzy human decision-making approach with lower computational cost (especially for making a decision in a complex health environment).

Fuzzy logic here means that the logic values of variables can be a real number between 0 and 1 [1.32][1.33]. Also, fuzzy logic is widely used for medical decision making in health environments such as Value-Laden choices [6.1], medical decision making in the intensive care unit (ICU) [6.2], and artrial fibrillation detection [6.3, 6.4].

Fuzzy logic can be presented as the assurance about the decision. For example, we can make the wrong decision when the assurance is less than 1. Similarly, the proposed CDS can measure the assurance about the decision after taking actions (See eq. (6.6) and Section 6.2.4).

In this chapter, the proposed CDS runs a diagnostic test for a User, then diagnoses between the healthy or unhealthy situation with a low false alarm policy. Based on this policy, the CDS assumes that the User is healthy at first, but without taking any actions, the confidence and assurance factor is zero.

As mentioned above, the first assumption/decision of the CDS is that the User is healthy. However, without doing any actions, the system has zero confidence and zero assurance about the decision of the User being healthy. If the system finds that the person is not healthy, this is 100% assurance, because the key feature has a value outside of the normal range, then all processes will be stopped. In this case, the User should be referred for disease diagnosis to the medical doctors or other algorithms. However, by doing more actions, the CDS can be more confident about the first assumption and decision (that is, the User is healthy). In any focus level, if the CDS meets the predefined threshold of assurance (e.g., 97.5% of assurance) or acceptable probability of error (e.g., 2.5% chance of error) about the initial decision that the User is healthy, then the CDS can send the User to the prevention part of the healthcare system.

If the focus level of the CDS cannot be increased, but the CDS finds all key features are within the normal range and the CDS does not assure enough about the health condition of User (e.g., 94% assurance< 97.5% accuracy threshold), then it will claim the User as healthy. However, the CDS knows that the threshold was not met for the User and refers the User to medical doctors for screening or to other screening algorithms. In screening, unlike our proposed algorithms in this chapter, the assumption and policy are that the User is unhealthy unless the healthcare system can find evidence that shows that the User is healthy. Furthermore, in screening, a high false alarm is acceptable.

6.2.2. High-level algorithm presentation for the proposed CDS

In this section, we describe the high-level presentation algorithm for the proposed CDS that can be applied for the ADMS of a smart e-Health home. The high-level algorithm shows the outline of the complete procedure of a proposed CDS in one focus level. Table 6.1 lists the notations used in this chapter for easy referencing.

The algorithm is described briefly as follows (Please see Figure 6.1):

- CDS training mode:
  a. Perceptor training mode.
  b. Executive training mode.
- CDS prediction mode:
  a. Executive (planner).
  b. Executive (reinforcement learning).
  c. Perceptor (run diagnostic test).
  d. Perceptor (assurance factor calculation).
  e. Executive (policy).

High-level algorithm for proposed CDS using algorithms 6.1-6.4

CDS training part: Re-run after updating the database or focus level increasing by internal commands

Perceptor training part (see algorithms 6.1-6.3 and Figure 6.1):

Creating the decision tree for focus level *m*

Modeling algorithm extracts the normal ranges for each feature and creates the model library for focus level *m*

Executive training part:

Extract key features from sensors and disease class of $hd_{h>1} \in \{hd_2, \dots, hd_H\}$

Important feature refining (has new information about User-health) and creating actions library for focus level *m*

CDS User-health prediction (See Algorithm 6.4 and Figure 6.1):

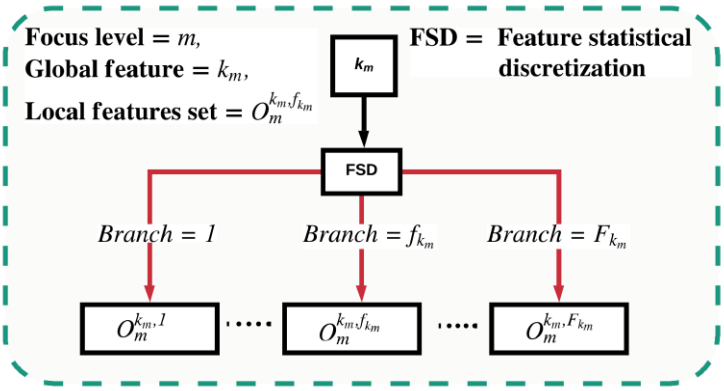
1. Depend on User request or periodically test run the diagnostic test in focus level m=1
2. Check if the User has the disease $hd_h \in \{hd_2, \dots, hd_H\}$
3. **Executive (Planner)**: Create a buffer action space for disease class of hd_h as the $\boldsymbol{A}$ vector

4. **Executive (Reinforcement learning)**: Finds the best action to minimize the reward for disease hd_h using reinforcement learning and remove this action from the buffer action space of $\boldsymbol{A}$
5. **Perceptor:** Run the diagnostic test and load the model for the related feature from the model library
6. **Perceptor (Decision making):** If the User is unhealthy: Alarm and send for the disease diagnosis process
7. **Perceptor (Raw internal reward calculation):** If the User healthy: calculate the assurance factor (AF)
8. **Perceptor:** If the (1-AF) ≤ *Threshold*: Claim User is healthy and start healthy living recommendations
9. **Executive (Reinforcement learning):** If the (1-AF) > *Threshold*: Find the next best action that minimizes the reward in the actions buffer space of $\boldsymbol{A}$
10. **Executive (policy):** If there is no more actions buffer space of $\boldsymbol{A}$ and (1-AF) > *Threshold*: Run the CDS prediction process for hd_{h+1} and h+1 ≤ H
11. **Executive (policy):** If all $hd_h \in \{hd_2, \dots, hd_H\}$ are checked, and the User does not have any $hd_h \in \{hd_2, \dots, hd_H\}$ and still (1-AF) > *Threshold* and if focus level can be increased (see eq. 2): Send internal commands to the perceptor for increasing the focus level and re-run all CDS training and prediction modes for new focus level of (m+1) ≤ M
12. **Executive (policy):** If all $hd_h \in \{hd_2, \dots, hd_H\}$ are checked, and the User has not any $hd_h \in \{hd_2, \dots, hd_H\}$ and still (1-AF)>*Threshold*, but focus level cannot be increased (see eq. 2): Claim User as the healthy and send the User for the screening

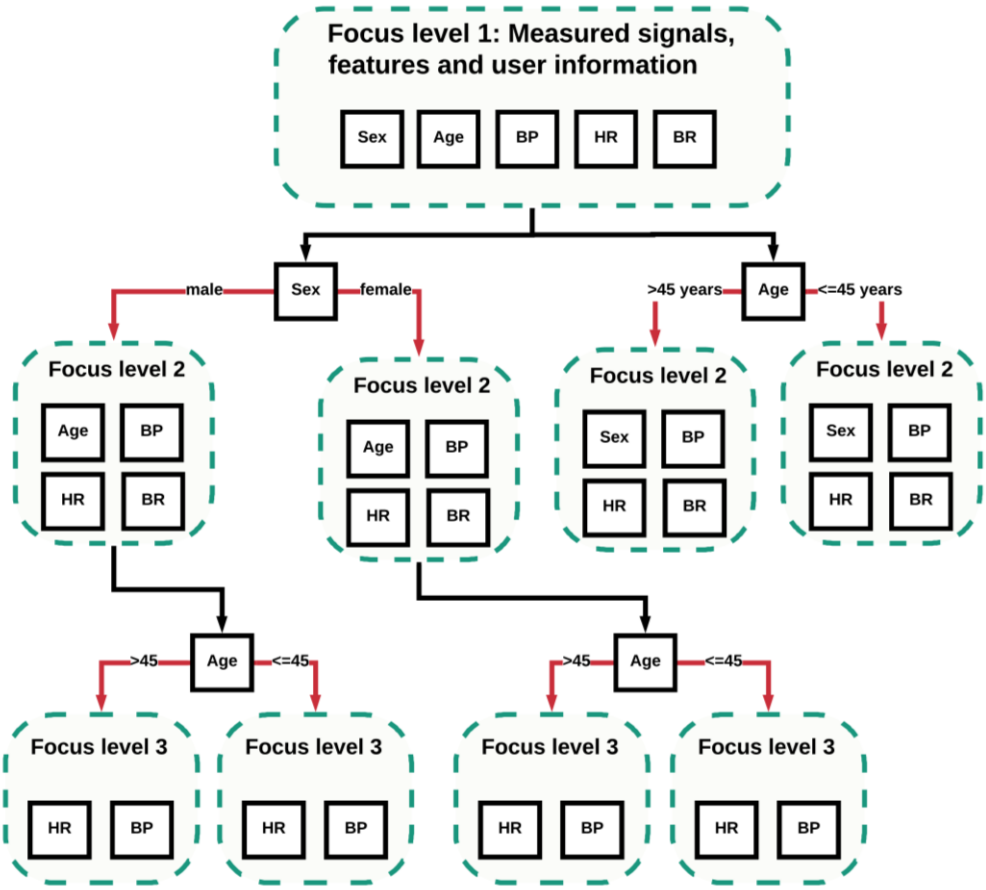
6.2.3. Training mode: perceptor and executive

As it was mentioned in chapters 1-5, in a conventional CDS, as, the *Kalman filter* cannot be used for non-Gaussian environments. Therefore, the Bayesian filter is omitted, and instead, the Bayesian equation is directly used for extracting the posterior in decision making. The multilayer Bayesian model using decision trees is a typical method in machine learning. The CDS training mode can be summarized in the following three parts which can be described as follows:

1. Creating a decision-making tree (see Algorithm 6.1),
2. Knowledge and actions extraction from the database (see Algorithm 6.2)
3. Actions refinement in the executive library (see Algorithm 6.3).



(a)



(b)

Figure 6.2: The CDS can create the proposed decision-making tree (a) the general schematic, (b) a simple example of the general schematic. BP: Blood Pressure, HR: Heart Rate, BR: Breathing Rate [1.37].

1) Creating a decision tree

Medical doctors typically use the decision-making tree approach for screening or diagnostics. For example, the pediatric tachycardia algorithm with a pulse and poor

perfusion are discussed in [6.5][6.6]. Therefore, the CDS should automatically create this decision-making tree. A general schematic of the proposed decision-making tree is shown in Figure 6.2a.

In Figure 6.2b we provide a simple example of the decision-making tree with three focus levels. In Figure 6.2b, three "Focus levels" are shown. In "Focus level 1," the measured signals, User information, features, actions, and desired false alarm rate are the inputs. Then, the CDS check all five features (Sex, Age, BP, HR, and BR), and computes the internal rewards. If the internal reward is below a predefined threshold, the CDS increases the focus level.

In "Focus level 2", for both branches of "sex" and "age," the system arbitrarily selects one of them, for example, "sex." Then, it computes the internal rewards (either male or female) based on the other four features (Age, BP, HR, and BR) and compares it to a predefined threshold. If the threshold for "sex" is not met, then the system switches to "age" and similarly computes the internal rewards and compares it to a predefined threshold. If the predefined threshold is again not met, then the system switches back to the "sex" branch and increases the focus level to "Focus level 3". In this example, the CDS continues in this way subject to eqs. (6.1) and (6.2) given below. The proposed idea of the decision-making tree is inspired by the focus level concept in a human brain [1.7], the focus level concept in CDS [1.8], and the MDs decision-making approach.

To create the decision-making tree and implement the focus levels in CDS, we use Algorithm 6.1. This proposed algorithm updates the Users and features for each tree branch f (e.g., male or female), tree node k (e.g., sex), and focus level m (e.g., second focus level). Figure 6b shows an example of the decision-making. These updates can be done by calculating the probability of the branch in the database, which can be extracted as $P_f^{(k_m m)}$. The extracted $P_f^{(k_m m)}$ is easily found using the probabilities extracted for the discretized node feature k (see Algorithm 6.1) in focus level $(m\text{-}1)$ with $m>1$.

The tree branches can be binary, because traditionally, some features have two values. For example, someone can be either a smoker or non-smoker, drinks alcohol or does not drink alcohol. However, for some features such as the age, more than two

branches can be created if there are enough number of instances in the training database. Therefore, eq. (6.1) shows the condition that can be used for branching using the feature statistically discretized (FSD) approach.

$$\text{ceil}(length\ of\ \boldsymbol{U}_{(m-1)\text{kf}} \times P_f^{(k_m m)}) \geq umin \tag{6.1}$$

$$umin = \frac{5}{Threshold}\ . \tag{6.2}$$

In eq. (6.1), $umin$ is the minimum required number of Users for branch tree f and focus level m for reliable model extraction. Also, $\boldsymbol{U}_{mkf}$ is the number of users in branch f, feature k and focus level m. In eq. (6.2), the *Threshold* is a predefined acceptable diagnostic error by the system. This minimum value for the number of Users can be found using a Monte-Carlo approach for extracting reliable conditional probabilities or a Bayesian generative model. Then, the information related to this decision tree can be saved in the perceptor and executive memory.

Besides, if the *Threshold* (see eqs. (6.1) and (6.2)) for the internal reward is not meet for all branches of the global feature of k_m in focus level m (see Fig. 6.2a), then the perceptor will use the global feature of $(k+1)_m$ for branching. However, if at least one branch of k_m achieves the *Threshold*, then the executive increases the focus level m to focus level (m+1).

Table 6.1: The notation used in this chapter

Notation	Definition
FA	The acceptable false alarm by the specified policy
h	Current health number condition
hd_h	hth disease for diagnosing
$\boldsymbol{HD}$	$\boldsymbol{HD} = \{hd_1, hd_2, \dots, hd_H\}$ is related to health conditions we would like to diagnosis. hd_h can map to number as follow: $h\in${1= healthy, 2=1st disease for diagnosing, 3= 2nd disease for diagnosing, …, the last number = unhealthy but have unknown disease}
M	Maximum desired focus level
m	The current focus level
k_m	The total number of global features in focus level m
k_m	Current global feature number

F_m^k	The total number of tree branches for the feature *k* in-focus level *m*
f_m^k	The current tree branch in the feature *k* in-focus level *m*
T_{mkf}	The total PACs number for branch *f*, feature *k*, and focus level *m*
t_{mkf}	The current local PAC number for branch *f*, feature *k*, and focus level *m*
$\boldsymbol{O}_{mkf}$ or $\boldsymbol{O}_m^{kf}$	The set of all available features for branch *f*, feature *k*, and focus level *m*
o_{mkf} or $\boldsymbol{o}_m^{kf}$	The current feature of available features for branch *f*, feature *k*, and focus level *m*
$\boldsymbol{U}_{mkf\|h}$	The set of all available users in the database for branch *f*, feature *k* and focus level *m for given h*
$u_{mkf\|h}$	The current user for branch *f*, feature *k*, and focus level *m* for given *h*
$\boldsymbol{U}_{mkf}$	The set ofAll available users in branch *f*, feature *k*, and focus level *m*
u_{mkf}	The current user for branch *f*, feature *k*, and focus level *m* for given *h*
$\boldsymbol{BD}_m^k(o_{mkf},\ \boldsymbol{U}_{mkf})$	The set of values for feature *o* for all available users in branch *f*, feature *k*, and focus level *m*
$\boldsymbol{B}_m^k$	$\boldsymbol{BD}_m^k(o_{mkf},\ \boldsymbol{U}_{mkf})$
$\Delta B_{o_m^k}$	Discretization step
$BD_m^k(o_{mkf},\ u_{mkf})$	The values for feature *o* for specific user *U* in branch *f*, feature *k*, and focus level *m*
B_m^k	$BD_m^k(o_{mkf},\ u_{mkf})$
$B_{min}^{km}(o_{mkf}, u_{mkf})$	Minimum value of B_m^k
$B_{max}^{km}(o_{mkf}, u_{mkf})$	Maximum value of B_m^k
$\hat{B}_m^k$	estimate B_m^k
$b_{min}^{mkf,o}$	The minimum boundary for the healthy range for feature *o* and in branch *f*, feature *k* and focus level *m*
$b_{max}^{mkf,o}$	The maximum boundary for the healthy range for feature *o* and in branch *f*, feature *k* and focus level *m*
$\boldsymbol{P}(\hat{B}_m^k(o_{mkf}, u_{mkf}))\|h)$	Bayesian generative model
$\boldsymbol{P}(h, f_m^k)$	Prevalence
$\boldsymbol{P}(\hat{B}_m^k(o_{mkf}, u_{mkf}))$	Evidence
$\boldsymbol{P}(h\|\hat{B}_m^k(o_{mkf}, u_{mkf}))$	Posterior
$r_{\boldsymbol{o}_m^{kf}\|h}$	The current probability for the value of feature *o* for given *h* happen out of normal range, in branch *f*, feature *k* and focus level *m*

$\boldsymbol{r}_{\boldsymbol{O}_m^{kf}\|h}$	The set of all probabilities for the value of a set of $\boldsymbol{O}$ for given *h* happen out of normal range, in branch *f*, feature *k* and focus level *m*
$\boldsymbol{R}_{\boldsymbol{O}_m^{kf}\|\boldsymbol{HD}}$	The set of all probabilities (Actions weight) for the value of a set of $\boldsymbol{O}$ for given all set o health situation $\boldsymbol{HD}$ happen out of normal range, in branch *f*, feature *k* and focus level *m*
$\boldsymbol{C}_{mkf\|h}(FA=0)$	The set of all actions in the static library for zero FA policy in branch *f*, feature *k* and focus level *m*
$c_{mkf\|h}(FA=0)$	The current action in the static library for zero FA policy, in branch *f*, feature *k* and focus level *m*
$\boldsymbol{C}_{m\|\boldsymbol{HD}}^{kf}$	The actions matrices for all given health conditions $\boldsymbol{HD}$, in branch *f*, feature *k* and focus level *m*
$c_{(h\|o_{mkf})}^{mkf}(FA=0)$	The current action for given feature *o*, in branch *f*, feature *k* and focus level *m*, for health condition *h*, for zero FA policy
$\boldsymbol{C}_{h\|\boldsymbol{O}_{mkf}}^{mkf}(FA=0)$	The set of all actions for given set of feature $\boldsymbol{O}$, in branch *f*, feature *k* and focus level *m*, for health condition *h*, for zero FA policy
umin	The minimum required users for branch tree *f* and focus level *m* for reliable Bayesian model extraction.
$AF_{t_{mkf}}$	The related assurance factor for PAC number t_{mkf}
$rw_{t_{mkf}}$	The internal rewards as defined $(1-AF_{t_{mkf}})$

In each focus level, the CDS learns which features can provide new information about the unhealthy conditions of Users. The main advantage of this method is that even without a database, when the health condition of the User is known as no disease, the system can extract the normal ranges from sensor(s) measurements. However, the information about important features (i.e., the features that provide information about a User's health) helps the CDS to find important features for doing purposeful and cognitive actions.

For example, a smart e-Health home may actuate the blood pressure, weight, and heart rate sensors to diagnose tachycardia. However, cognitive actions like those taken by MDs can only actuate the heart-rate sensor for the diagnosis. Also, more focus levels can reduce the natural diagnosis error by default. Here, the natural diagnosis

error means that there is a wrong decision due to incorrect reasoning. A wrong decision does not include errors due to measurement noise or failures of the sensors.

As an example, it is known that the normal fractional carbon monoxide in human blood is less than one percent [6.7]. As a result, more than one percent of carbon monoxide in the blood may need an alarm and treatment actions related to carbon monoxide in the immediate environment. However, with more evidence, the CDS may find that the Users are smokers. Therefore, more focus levels or branches such as how many cigarettes are smoked per day, the interval between smoking cigarettes and cigarettes brand will be implemented.

The normal amount carbon monoxide in the blood of a healthy smoker may be higher than one percent. As an example, for a smoker who smokes one pack of cigarettes per day, their normal carbon monoxide level in the blood can reach 3% to 6%. This is increased to 6% to 10%, when smoking two packs per day, and increases to 20% for three packs a day [6.7]. Therefore, the adaptive training system in the CDS perceptor makes it flexible in diagnosing severe health issues. However, if we consider smoking or tobacco addiction as the disease, then the scenario can be different.

Algorithm 6.1 shows how this training part is working in our CDS. Algorithm 6.1 shows how the CDS can find the important features in any arbitrary focus level. It uses a vector line for the calculation to reduce the algorithm complexity as much as possible. Therefore, it can be trained and can make a decision dynamically for real-time situations in a smart e-Health home. In this way, CDS is a suitable platform for ADMS.

In Algorithm 6.1, in line 8, we assumed from focus level 3 that the redundant tree branches should be removed. For example, branching from sex in focus level 2 (m=2) to age in focus level 3 (m=3) (see Figure 6.2b) is similar to branching from age in focus level 2 (m=2) to sex focus level 3 (m=3)(The detailed decision trees creation and equations are presented in chapter 7).

Algorithm 6.1: *Creating a decision tree and updating Users and features database for related branches*

1: **Input:** The observables and features from the database for each User for the focus level *m*, global feature *k*, branch tree *f*, policy, labeled dataset, actions space $\boldsymbol{U}_{(m-1)kf}$, $\boldsymbol{O}_{(m-1)kf}$

Output: For the new branch, focus level and global feature *k*, the new User set of $\boldsymbol{U}_{mkf}$, the new features set of $\boldsymbol{O}_{mkf}$

Initialization:

1: **for** m = 2 to M **do**

2: **for** $k =$ set of $\boldsymbol{O}_{(m-1)kf}$ **do**

3: Statistical discretization for features can have more than two branches [Age, weight, height, heart rate, sleep, …], $f_{k_m m} \in \{1,2,\dots,F_{k_m m}\}$

4: Creating a binary branch for: [sex, smoking, drug, family disease history, drinking, night sleep, activity/rest, ⋯], $f_{k_m m} \in \{1,2\}, F_{k_m m} = 2$

Updating Users in the database for the branch of each tree depend on the probability

5: **for** set of $f_{k_m m} \in \{1,2,\dots,F_{k_m m}\}$ **do**

6: Calculate the $P_f^{(k_m m)}$: use previous probabilities extracted in previous focus level

7: $\boldsymbol{U}_{mkf} = P_f^{(k_m m)} \boldsymbol{U}_{(m-1)kf}$

8: **if** m>2 **then**

9: $\boldsymbol{O}_{mkf} = \boldsymbol{O}_{(m-1)kf} - \{1,\dots,k\}$

10: **End if**

11: **End for**

12: **End for**

13: **End for**

2) Model library in the perceptor and actions library in the executive

Normally, a MD can diagnose someone with diseases even if they did not diagnose such diseases before. Therefore, a semi-MD's approach is used for smart e-Health home applications. Here, the ADMS layer should be flexible enough to work with different features, conditions and policies. In addition, the ADMS may face two challenges in a smart e-Health home.

1. There is no database, but we can extract and measure the features of current Users. The designed CDS need to be trained using a database, which is a collection of data corresponding to a data matrix table. In the database table, every column of the table is for a specific feature, and each row corresponds to a given User with measured data and known health status. We can extract the normal range for such a person with known health conditions and when they are healthy. Then, the CDS can find if a User has some diseases by extracting some features that seem abnormal.
2. We have enough number of instances in the database that includes healthy and unhealthy Users. However, we may have a failure of one or more sensors. In this case, the CDS should be able to find alternative features (i.e., features of available sensor signals that can help to diagnose a User's health state) in related available actions to keep the performance reliable enough. Therefore, in a real-time situation, the CDS algorithms should take proper alternative actions quickly.

The proposed CDS for ADMS in this chapter can tackle conditions 1 and 2, which are important for improving the system's reliability in a health-care scenario. To meet challenges 1 and 2, we should use an algorithm with low complexity. The first training level is to find the model and the normal ranges. For extracting the posterior, the following probabilities should be calculated, $\boldsymbol{P}(\hat{B}_m^k(o_{mkf}, u_{mkf})|h)$ (*Bayesian model*), $\boldsymbol{P}(h, f_m^k)$ (*prevalence*) and $\boldsymbol{P}(\hat{B}_m^k(o_{mkf}, u_{mkf}))$ (*evidence*) The $\boldsymbol{P}(h|\hat{B}_m^k(o_{mkf}, u_{mkf}))$ (*posterior*) can be calculated using the Bayesian equation as follows:

$$\boldsymbol{P}(h|\hat{B}_m^k(o_{mkf}, u_{mkf})) = \frac{\boldsymbol{P}\left(\hat{B}_m^k(o_{mkf}, u_{mkf})\right)|h)\boldsymbol{P}(h, f_m^k)}{\boldsymbol{P}(\hat{B}_m^k(o_{mkf}, u_{mkf}))}, \tag{6.3}$$

where the $h \in \boldsymbol{HD}$ can correspond to $\boldsymbol{HD} = [1,2, \dots, H]$ (see Table 6.1 also). $\hat{B}_m^k(o_{mkf}, u_{mkf})$ is the discretized values for the feature number o, focus level m, main feature k and branch number f. In the database, the normal range means that 100% of the values of the features of Users without diseases fall in this range that can be calculated (see Figure 6.3) using:

$$\sum_{\text{set of } \widehat{\boldsymbol{B}}_m^k(o_{mkf},\boldsymbol{U}_{mkf})} \boldsymbol{P}(h=1|b_{min}^{k,m}(o_m^k(f_m^k)) \leq \hat{B}_m^k(o_{mkf},u_{mkf}) \leq b_{max}^{k,m}(o_m^k(f_m^k))) = 1. \quad (6.4)$$

Algorithm 6.2 shows the model extraction by the perceptor for creating the model library. Also, Algorithm 6.2 creates the actions library by the executive. As a result, the actions can be assumed to be the same as MDs decision-making, when they find some features are not in normal range. In addition, in common with MDs, these normal ranges can be dynamically changed by receiving new information or knowledge after dynamically updating the database. In addition, the normal ranges can be varied with more focus for better accuracy.

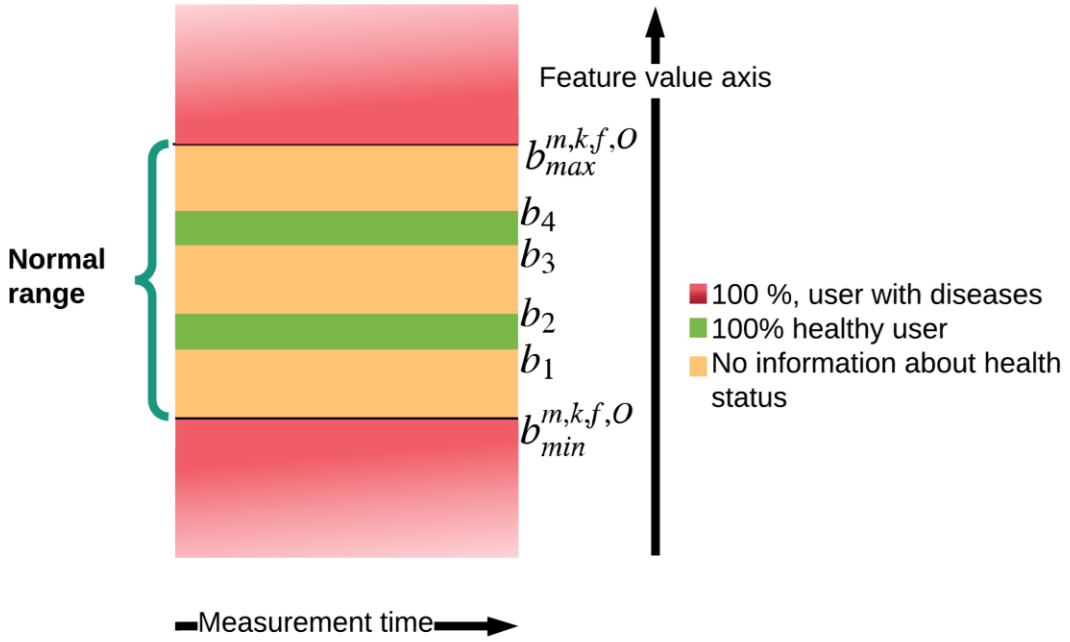


Figure 6.3: The possible feature value and ranges for users with and without diseases in the database [1.37].

Algorithm 6.2: The proposed system for the training mode of the perceptor and executive for tree branch of f, node feature k and m focus level

Input: The observables and features from the database for each User for the focus level *m*, node feature *k*, branch tree *f*, policy, database of Users and health conditions vector.

Output: Actions weight matrices $\boldsymbol{R}_{\boldsymbol{O}_m^{kf}|\boldsymbol{HD}}$, actions matrices $\boldsymbol{C}_{m|\boldsymbol{HD}}^{kf}$, probabilities and the normal range for each feature

Initialization:

1: Calculate $\boldsymbol{U}_{mkf}$

2: Calculate the feature matrix number $\boldsymbol{O}_{mkf}$

Perceptor training mode

3: **for** o_{mkf}= set of $\boldsymbol{O}_{mkf}$ **do**

Updating corresponding features signals in the database

4: $\boldsymbol{B}_m^k = \boldsymbol{BD}_m^k(o_{mkf},\ \boldsymbol{U}_{mkf})$

The maximum and minimum value for the feature o_{mkf}

5: Calculate $B_{min}^{km}(o_{mkf},\ u_{mkf})$ and $B_{max}^{km}(o_{mkf},\ u_{mkf})$

Discretization step

6: Calculate $\Delta B_{o_m^k}$

Estimation by discretization

7: Calculate $\hat{B}_m^k(o_{mkf},\ u_{mkf})$

8: **for** h = set of ***HD*** **do**

9: Calculate $\boldsymbol{P}(\hat{B}_m^k(o_{mkf},\ u_{mkf})|h)$, $\boldsymbol{P}(h, f_m^k)$, $\boldsymbol{P}(\hat{B}_m^k(o_{mkf},\ u_{mkf}))$ and $\boldsymbol{P}(h|\hat{B}_m^k(o_{mkf},\ u_{mkf}))$

10: **Save probabilities in the perceptor library**

11: **End for**

Normal ranges calculation

12: Calculate $b_{min}^{mkf,o}$ and $b_{max}^{mkf,o}$

Maximum cell number in normal ranges

13: Calculate $N_m^{kf} = \frac{b_{max}^{kmf,o} - b_{min}^{kmf,o}}{\Delta B_{o_m^{kf}}}$

14: Save normal ranges and N_m^{kf} in healthy ranges in the perceptor library

Executive training mode for FA = 0

15: $r_{o_m^{kf}|h=1}$= 0

16: **for** h = 2 to **H do**

17: $r_{o_m^{kf}|h}$= 0

Build actions library for FA = 0

18: **if** ($\boldsymbol{P}(\hat{B}_m^k(o_{mkf},\ u_{mkf}) < b_{min}^{mkf,o}|h)$>0 or $\boldsymbol{P}(\hat{B}_m^k(o_{mkf},\ u_{mkf}) > b_{max}^{mkf,o}|h)$>0) and $(\boldsymbol{P}(h, f_m^k), > 0)$ **then**

19: $r_{o_m^k|h}$= ($\boldsymbol{P}\big(\hat{B}_m^k(o_{mkf},\ u_{mkf})\big) < b_{min}^{mkf,o}|h)$+$\boldsymbol{P}\big(\hat{B}_m^k(o_{mkf},\ u_{mkf})\big) > b_{max}^{mkf,o})|h)$

Updating the action related to the current signal feature

20: $c^{mkf}_{(h|o_{mkf})}(FA=0) \leftarrow sensor\ for\ o_{mkf}$

21: **End if**

22: **End for**

23: **End for**

In the executive training mode, the executive extracts actions and their weights for low false alarm policy (diagnostic test). Algorithm 6.2 shows how these extractions are performed (See Fig. 6.3 also). If we have the features that provide information about Users with diseases (that the feature value is in the red region, see eq. (6.4) also), then this can be considered as an important feature. Data training is conducted using Algorithm 6.2. Then, Algorithm 6.3 can refine actions in the executive training mode. This means that the executive keeps important actions for diagnosing between healthy and unhealthy situations. Important actions mean that the features that can provide new information about an unhealthy situation. Algorithm 6.3 helps the system to use less memory, and because it has low complexity, it can reduce additional sensors usage and activation. This results in energy saving and longer sensors' lifetimes.

The CDS stores the features of important actions in the executive action library. The action space can be defined as follows:

$$C=\left\{ c_k \middle| c_k = \begin{array}{c} Actuate\ sensor\ c^{mkf}_{(h|o_{mkf})}(FA=0)\ for\ o_{mkf} \\ Internal\ commands = Increase\ m,\ m<M \end{array} \right\}. \quad (6.5)$$

Here, c_k is the action in PAC number k and m is the focus level (see Table 6.1). The action space means that all possible actions can be done during a User's health prediction mode by the executive.

Algorithm 6.3: Actions refining, f^k_m *and FA = 0* and just checking if someone is healthy or not, search for, if there is new information about any Users

Input: Observables and features from the database for each User for the focus level m, node feature k, branch tree f, policy, training database, actions space $\boldsymbol{C}^{kf}_{m|\boldsymbol{HD}}$, $\boldsymbol{BD}^{kf}_{m}$, $\boldsymbol{R}^{kf}_{m|\boldsymbol{HD}}$

Output: The important actions of $\boldsymbol{C}^{kf}_{m|\boldsymbol{HD}}$

Initialization:
buffer= 0, $\boldsymbol{newinf} = \emptyset$

1: **for** h =2 to **H do**

2: $r\_buf = r_{\boldsymbol{o}_m^{kf}|h}$
3: Sort the r_buf in descending order and remove 0 value elements
4: **for** o_{mkf}= set $\boldsymbol{O}_{mkf}$ **do**
5: **for** u = set of $\boldsymbol{U}_m^k(f_m^k)$ **do**
6: **if** $BD_m^k(o_{mkf},u)>b_{max}^{mkf,o}$ **or** $BD_m^k(o_{mkf},u) < b_{min}^{mkf,o}$) **then**
7: $newinf_u = 1$
8: **End if**
9: **End for**
10: s = summation of $\boldsymbol{newinf}$
11: **if** summation of s <=Buffer **then**
12: $c_{(h|o_{mkf})}^{mkf}(FA=0) \leftarrow \{\}$
13: $r_{o_m^{kf}|h}$=0
14: **End if**
15: Buffer = s
16: **End for**
17: **End for**

6.2.4. Prediction mode: perceptor and executive

Algorithm 6.4 shows the implementation of the CDS prediction mode. Also, this algorithm shows how the CDS performs when we have a User in a smart e-Health home with an unknown health situation. In this condition, the system randomly chooses an action from the actions library. Then after applying new actions, the assurance factor (AF) is calculated using eq. (7):

$$AF_{t_{mkf}} = \frac{\boldsymbol{P}(h,f_m^k)\, r_{j|h}}{\boldsymbol{P}(h>1,f_m^k)} + AF_{(t_{mkf}-1)}. \quad (6.6)$$

The assurance factor measures the expected assurance about the decision after doing current action $c_{(h|o_{mkf})}^{mkf}(FA=0)$. Therefore, the internal reward can be calculated as follows:

$$rw_{t_{mkf}} = (1 - AF_{t_{mkf}}). \quad (6.7)$$

Then, the system, depending on the health conditions related to $hd_2, \dots, hd_H$ (see eq. (2.1)), will apply related actions. Related actions mean asking for related information to the User or activating sensors. For each disease, the executive activates the sensors to obtain maximum information about the health conditions $hd_h \; and \; h > 1$ (see eq.

(2.1)) using the planning and learning sections (see Algorithm 6.4). Reinforcement learning will be done once when the database is updated with new information.

Algorithm 6.4: CDS User-health prediction (planner, reinforcement learning and policy in executive and running diagnostic test in the perceptor)

Input: The observables and features from the database for each User for the focus level *m*, node feature *k*, branch tree *f*, policy (FA=0), a database of Users, $\boldsymbol{HD}$ health condition vectors.

Output: Actions weight matrices $\boldsymbol{R}_{\boldsymbol{O}_m^{kf}|\boldsymbol{HD}}$, actions matrices $\boldsymbol{C}_m^{kf}$, probabilities and the normal range for each feature

---

Initialization:

c_0 ←The actions, [$Vital\ signs\ and\ portable\ sensors, \cdots$] apply on smart e-Health home User

Start advanced actions such as 12 leads ECG or … if c_0 shows something abnormal or voices show pain, need help, or User saying directly, "I am not feeling well."

Load the normal ranges $\boldsymbol{b}_{min}^{m=1}(\boldsymbol{O}_{m=1})$, $\boldsymbol{b}_{min}^{m=1}(\boldsymbol{O}_{m=1})$, Actions weights $\boldsymbol{R}_{\boldsymbol{O}_{m=1}|\boldsymbol{HD}}$, Actions $\boldsymbol{C}_{m=1|\boldsymbol{HD}}^{\boldsymbol{O}_{m=1}}(FA=0)$

m=1 (focus level 1)**,** $\boldsymbol{k_{m=1}} = \emptyset$, $\boldsymbol{f_{m=1}} = \emptyset$, $\boldsymbol{t} = \emptyset$, Decision = 1, *Threshold* = Acceptable estimated error.

$c_{m=1|h}^{o_{m=1}}(FA=0)$ ← an action randomly selected from $\boldsymbol{C}_{m=1|\boldsymbol{h>1}}^{\boldsymbol{O}_{m=1}}(FA=0)$

Apply to $c_{m=1|\mathrm{h}}^{o_{m=1}}(FA=0)$ the User ($h \in \{2, \dots, H\}$)

Extract features $o_{m=1}^{0,0}$and $\boldsymbol{BD}_m^k(o_m^{kf}, u_{test})$ and choose one of them with maximum $r_{o_{m=1}|h}$

Calculate $AF_{t_{m=1,00}} = \frac{\boldsymbol{P}(h,\mathrm{m=1})\, r_{o_{m=1}|h}}{\boldsymbol{P}(h>1,\mathrm{m=1})}$

calculate first internal rewards $as\ 1 - AF_{t_{m=1,00}}$,

1: **for** *k*= set of $\boldsymbol{k_m}$ **then**
3: **for** f = set of $\boldsymbol{f_{mk}}$ **do**
4: $t_{mkf} = 0$
5: **for** *h* = 2 to **H do**
6: $\boldsymbol{r\_buf} = \boldsymbol{r}_{\boldsymbol{O}_m^{kf}|h}$
7: Sort the $\boldsymbol{r\_buf}$ decently and remove 0 value elements
8: $\boldsymbol{O}_m^{kf}$ = update features based on elements on $\boldsymbol{r_{buf}}$

Planning

9: $\boldsymbol{C\_buf} \leftarrow \boldsymbol{C}_{h|\boldsymbol{O}_{mkf}}^{mkf}(FA=0)$
10: $\boldsymbol{A}_{t_{mkf}} \leftarrow \boldsymbol{C\_buf}$

Reinforcement learning

11: **for** all actions $\left(c \in \boldsymbol{A}_{t_{mkf}}\right)$ **do**

12: **for** $(j \in set\ of\ \boldsymbol{O}_m^{kf})\ and\ (j \in features\ extracted\ from\ c)$ **do**

13: $AF^{cj}_{(t_{mkf}+j)} = \frac{\boldsymbol{P}(h,f_m^k)\, r_{j|h}}{\boldsymbol{P}(h>1,f_m^k)} + AF^{cj}_{(t_{mkf}+j)}$

14: $rw^{cj}_{(t_{mkf}+j)} = 1 - (AF^{cj}_{(t_{mkf}+j)} + AF_{t_{mkf}})$

15: **End for**

16: **End for**

17: $J \leftarrow \underset{\boldsymbol{J}\in\boldsymbol{A}_{t_{mkf}}}{\textbf{argmin}} \left\{\sum_j rw^{\boldsymbol{A}_{t_{mkf}}\, j}_{(t_{mkf}+j)}\right\}$

18: remove J from $\boldsymbol{C\_buf}$

19: Apply action (sensor activation) J on User

Run diagnostic test

20: $\boldsymbol{O_{buf}} \leftarrow features\ of\ J$

21: $\boldsymbol{V_{O_{buf}}} \leftarrow \boldsymbol{save\ values\ related\ to\ O_{buf}}$

22: **for** $(j \in set\ of\ \boldsymbol{O_{buf}})$ and $(j \in set\ of\ \boldsymbol{O}_m^{kf})$ **do**

23: $t_{mkf} = t_{mkf} + 1$

Internal rewards calculation

24: $AF_{t_{mkf}} = \frac{\boldsymbol{P}(h,f_m^k)\, r_{j|h}}{\boldsymbol{P}(h>1,f_m^k)} + AF_{(t_{mkf}-1)}$

25: $rw_{t_{mkf}} = 1 - AF_{t_{mkf}}$

26: **if** $\boldsymbol{V}_j < b^{mkfj}_{min}$ or $\boldsymbol{V}_j > b^{kmfj}_{max}$ **then**

27: Alarm and disease diagnosis process

28: Decision = Unhealthy

29: **Return** Decision

30: **End if**

31: **End for**

32: **if** $\boldsymbol{C\_buf} \neq \emptyset$ **then**

33: Go to line 18

34: **End if**

35: **End for**

36: **if** $rw_{t_{mkf}}$ <=*threshold* **then**

37: Healthy life recommendations

38: Decision = Healthy

39: **Return** Decision

40: **elseif** *Users in focus level m*+1>=$\underline{u_{min}}$ **then**

41: increase focus level

42: **else**

43: Run Screening process

44: **End if**

45: **End for**

46: **End for**

6.2.5. Complexity of the proposed algorithms

Typical machine learning (ML) techniques select the key features based on different methods such as principal component analysis. Then, the typical ML techniques use all key features simultaneously to extract the model. After this, the extracted model by ML can be used for decision making in a multi-dimensional space. In this approach with typical ML techniques, modeling, and decision making for Users with unknown health conditions will take much time. However, in our proposed algorithms, the key features can be found in the executive training mode (See Algorithms 6.2 and 6.3).

In addition, for decision making, the CDS will check all features one by one in each PAC. Therefore, instead of creating a multi-dimensional feature space, the CDS checks the features sequentially. Because of checking the features one by one, CDS can decide in a one-dimensional space. Then, the decision-making accuracy is improved using the PAC concept. As a result, the proposed algorithms are fast enough for real-time healthcare applications such as in a smart e-Health home.

Having fast enough algorithms for real-time healthcare applications are an important advantage of our proposed approach using CDS. However, it should be noted that the required memory for saving models and action space will increase as the number of focus levels are increased in the CDS. In addition, to address increasing algorithms complexity by increasing the number of focus levels, we can define a bound for the maximum possible focus level as the *Complexity threshold*. Then, we can calculate the total acceptable branches of the decision tree as:

$$F_m^{total,k} = \sum_{m=1}^{M} F_{k_m}, \quad \text{and, } F_m^{total,k} \le Complexity\ threshold \tag{6.7}$$

Furthermore, F_{k_m} corresponds to the maximum number of decision-making tree branches in focus level *m* and global feature *k.* For the desired predefined *Complexity threshold,* the CDS cannot increase the focus level more than *M*. However, in practice, the bound is determined by the lower value from either eq. (6.7) or eqs. (6.1) and (6.2), and that will be the maximum focus level.

6.3. Case study: Diagnosing between a user with or without arrhythmia

Every year, about 326,000 out-of-hospital sudden cardiac arrest (OHSCA) can happen in the USA [6.8]. The median age of these persons is about 65 years. Unfortunately, the survival rate is 10-11%. It is important to know that the survival rate can improve to 33.3% if OHSCA happens in front of another person(s) [6.9]. More than 80% of sudden cardiac death is due to ventricular Arrhythmia [6.9][6.10]. Globally, the prevalence of Arrhythmia is unknown. However, it is assumed that millions of people in the world have Arrhythmia [6.11]. Also, Arrhythmia is most common in persons older than 35 years [6.8][6.11].

In the USA, 850,000 persons are hospitalized each year due to Arrhythmia [6.8]. Some researches show that at least 16-17% of Canadians are not aware that they have Arrhythmia disease [6.12]. Diagnosis of Arrhythmia is made by the 12-leads ECG [6.13] (see Figure 6.4) or by using the Holter monitor for two or three weeks monitoring [6.14]. However, these methods can only diagnose about 50% of Arrhythmia [6.15]. Figures 8a and 8b show an example of 12-leads ECG output signal and ECG electrodes locations for leads on the human body, respectively.

In the standard 12-leads ECG, six of the leads are known as “limb leads” because they are placed on the wrists and feet of the User. These six limb leads are known as leads I, II, III, aVF, aVR, and aVL (See. Figure 6.4b). The letter “a” represents for “augmented,” as aVF, aVR and aVL leads are calculated as a combination of leads I, II, and III. Therefore, as shown in Figure 6.4b, aVF, aVR and aVL are the names for ECG leads for foot and right and left wrists, respectively. The leads V1-V6 are known as “precordial leads” because leads V1-V6 are placed on the chest (precordium). Also, in this chapter, leads V1-V6 corresponds to the ECG leads 1-6.

In this section, simulation results are presented for the Arrhythmia case study as a proof-of-concept study, using the proposed algorithms in the previous sections. We used the MATLAB Arrhythmia database [6.16] for the case study simulations. Also,

the MATLAB database is known as the UCI Machine Learning Repository [6.17]. The MATLAB Arrhythmia database includes 279 extracted features including age, sex, height, weight, heart rate and 12-leads ECG features (see Fig. 8) such as T wave angle, T wave duration and QRS angle for each lead for 452 persons (for ECG signal feature definition see [6.18]).

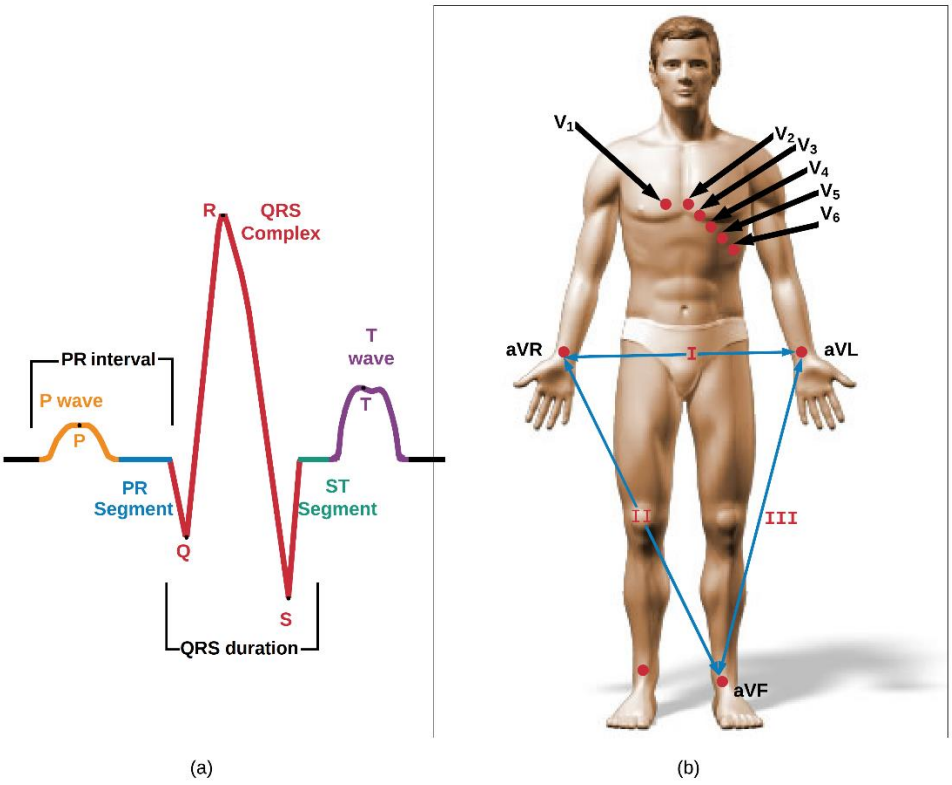


Figure 6.4: The 12 leads ECG (a) an example of an ECG signal (b) electrode places on the human body [1.37].

The MATLAB Arrhythmia database is mentioned as one of the hardest databases for classification [6.19]. This is because the MATLAB Arrhythmia database has a few numbers of instances (i.e., 452 instances) and there are some missing features such as J angle or heart rate (key feature) for some Users. Therefore, these missing features can simulate sensor failure in practical applications.

Table 6.2: List of most prevalent Arrhythmia classes and example of actions in Figure 6.5 (Focus level 1).

Class	Action example to diagnose this class (Sensor) (see Figure 6.4b)	Feature for this action (see Figures 6.4a and 6.4b)
2 (Arrhythmia)	Lead 6 (V6)	T wave amplitude
3 (Arrhythmia)	Lead 2 (V2)	Q wave duration
4 (Arrhythmia)	Lead 2 (V2)	DII N intrinsic (or intrinsicoid) of deflection [6.16]
5 (Tachycardia)	Heart rate	Heart rate per minute
6 (Bradycardia)	Heart rate	Heart rate per minute
9 (Arrhythmia)	Lead 2 (V2)	QRS amplitude
10 (Arrhythmia)	Lead 1 (V1)	QRS amplitude
16 (Unknown, Unhealthy)	aVF	P wave amplitude

Table 6.3: Prevalence in MATLAB database (Prior)

Class	Users out of 452 persons	Prevalence %
1 (Healthy)	245	54.20
2 (Arrhythmia)	44	9.73
3 (Arrhythmia)	15	3.32
4 (Arrhythmia)	15	3.32
5 (Tachycardia)	13	2.88
6 (Bradycardia)	25	5.53
7 (Arrhythmia)	3	0.66
8 (Arrhythmia)	2	0.44
9 (Arrhythmia)	9	1.99
10 (Arrhythmia)	50	10.11
11 (Arrhythmia)	0	0
12 (Arrhythmia)	0	0
13 (Arrhythmia)	0	0
14 (Arrhythmia)	4	0.89
15 (Arrhythmia)	5	1.11
16 (unknown, unhealthy)	22	4.87

Total number of persons with Arrhythmia	207	45.80
Female	249	55.09
Male	203	44.91
Age >45 years	242	53.54

The persons in the MATLAB database were classified into 16 classes. Class 1 is for no Arrhythmia, classes 2-15 correspond to different classes of Arrhythmia and class 16 corresponds to an unknown Arrhythmia class. Also, the prevalence and demographics of Users in the MATLAB Arrhythmia database are presented in Table 6.3. Arrhythmia is defined as irregular, too fast (tachycardia) or too slow (bradycardia) heartbeats.

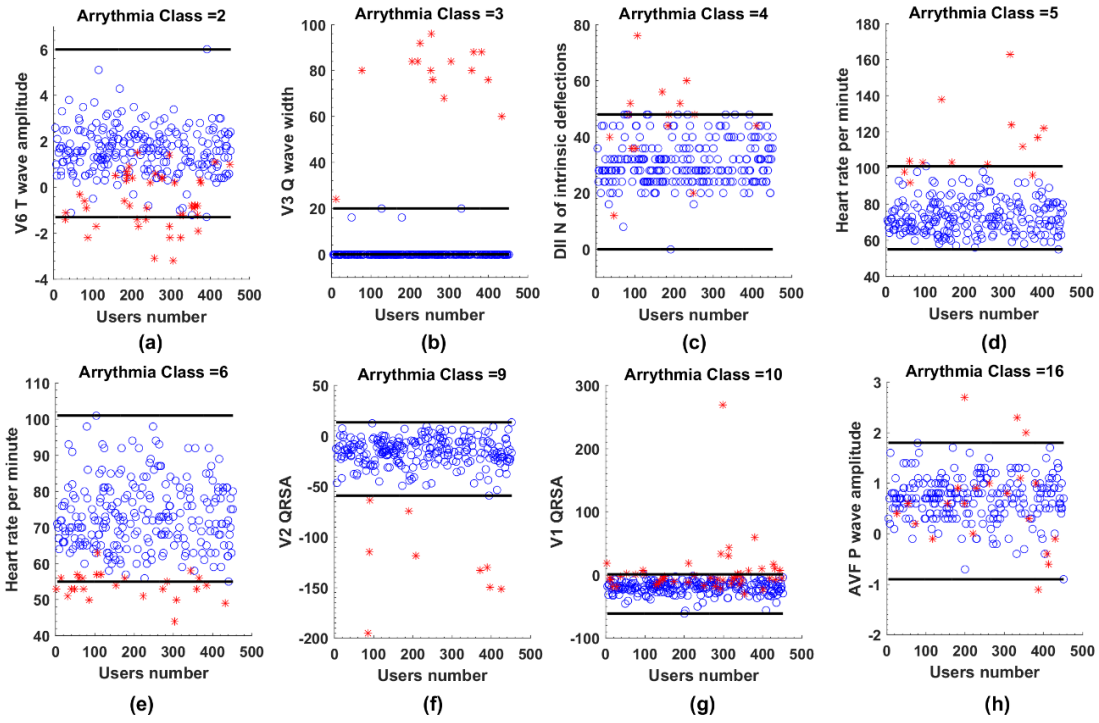


Figure 6.5: Examples of the actions selected by the CDS for the most prevalent class of Arrhythmia in the MATLAB database (Focus level 1), blue open circles are a healthy person and Red stars are users with Arrhythmia. Normal ranges are between the solid black lines. V1: ECG lead 1, V2: ECG Lead 2, DII N intrinsic (or intrinsicoid) of deflection: Deflection of ECG lead II (see. Figure 6.4 also) [1.37].

6.3.1. Simulation parameters and results

Here, we use the leave-one-out cross-validation (LOOCV) for training and verifying the performance of the CDS. In this method, all Users except one are used for training, and the excluded User is used for testing and health-state diagnosis accuracy. This process is repeated for N = 452 times for N Users based on the MATLAB Arrhythmia database. The advantage of LOOCV is that the entire database can be used for training and testing. The error rate can be calculated as the average of the error rate of each iteration. More detailed information related to the LOOCV validation technique can be found in [6.20].

The health *space* is $hd_h \in \{hd_1 = healthy, hd_2 = AC\ 2, hd_3 = Arrythmia\ class\ (AC)\ 3,\ hd_4 = AC\ 4, hd_5 = AC\ 5, hd_6 = AC\ 6,\ hd_7 = AC\ 7, hd_8 = AC\ 8, hd_9 = AC\ 9,\ hd_{10} = AC\ 10, hd_{11} = AC\ 14, hd_{12} = AC\ 15, hd_{H=13} = Others\ (class\ 16\ of\ database)\}$ where $h \in \{1,2,\dots,H = 13\}$. Based on the proposed algorithms, examples of some actions for Focus level 1, which can provide information about the Users' health status, are shown in Figures 6.5(a)-(h) for the most prevalent Arrhythmia classes (see Table 6.2) in the MATLAB database. Also, Table 6.3 shows the list of examples of actions in Figure 6.5 for the Focus level 1. Here, the CDS can diagnose Arrhythmia (see Figure 6.5) by activating the specific electrodes in the 12-leads ECG (see Algorithm 6.4 and section 6.2.4).

Actuating only required electrodes results in lower sensors usage, lower energy usage and longer sensor lifetime. We set the *Threshold* for the estimated error at 2.5% (i.e., (1-*AF*)≤0.025 and 97.5% assurance factor (*AF*), see eq. (6.2)). Therefore, using eq. (6.2), the *umin* is equal to 200. Due to the lack of enough Users in the MATLAB database, the CDS maximum focus level is 2 with two tree branches (see eq. (6.1), and *umin*=200).

In Focus level 2, CDS uses the sex feature in the MATLAB database only to extract the important features that depend on smart e-Health home User. Most features in the Arrhythmia database have a non-Gaussian distribution. For example, Figure 6.6 shows the conditional probability for the heart rate (HR) without and with Arrhythmia, which has a non-Gaussian distribution.

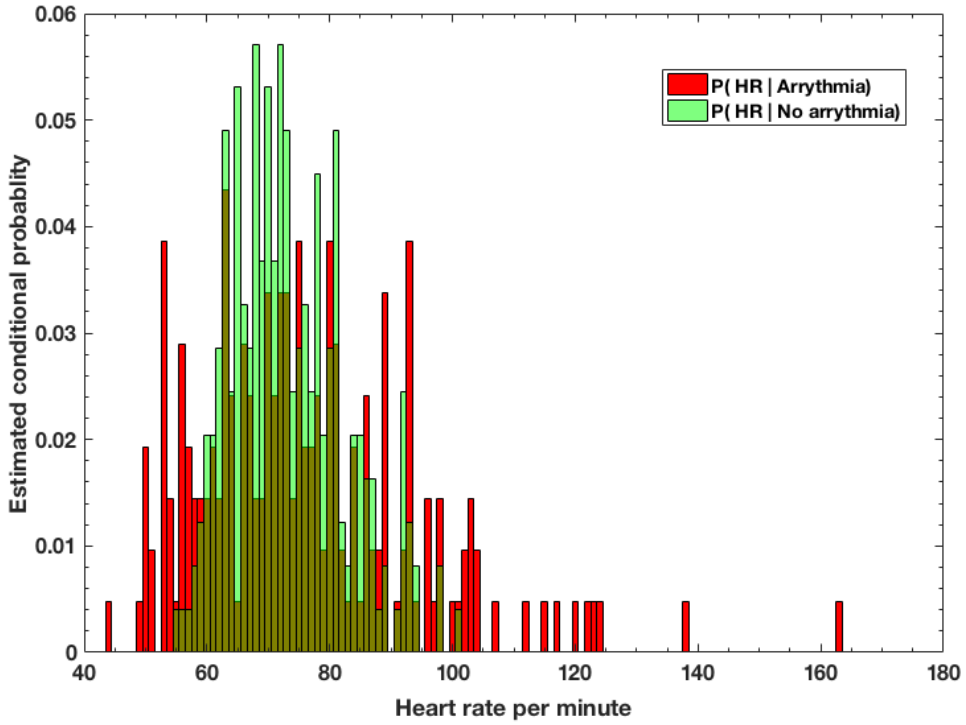


Figure 6.6: The conditional probability of heart rate for given health conditions: Light transparent green bar is persons without Arrhythmia, and the red bar is for persons with Arrhythmia (Focus level 1) [1.37].

As mentioned before, we used the LOOCV method to verify the proposed CDS algorithms performance for diagnosing the Arrhythmia as a proof-of-concept example. Therefore, the User-under-test (UUT) is removed from the Arrhythmia database. Then, at Focus level 1, using Algorithm 6.2 in the CDS training mode, the perceptor creates the model library, and the executive extracts the important features. Next, in Algorithm 6.3, these important features are refined and the features that did not provide new information are removed. Also, in Algorithm 6.3, an action space for the executive is created.

Using Algorithm 6.4 (also, see section 6.2.2) at Focus level 1, the executive (see Algorithm 6.4 and section 6.2.2, planner part) apply the actions from the actions library for diagnosing the disease class of $hd_{h>1}$. If the User is healthy and (1-*AF*)>*Threshold* = 0.025, then the CDS checks the User's health for $hd_{h+1\leq H=13}$. Here, in Figure 6.7, the average results are shown for the estimated diagnosis error as filled circles (i.e., 1-*AF*) and the average real error as the open circles for the LOOCV method. Also, in

Figure 6.7, the average diagnosis of the real error is 18% at PAC number 52 (the CDS does not know the real error shown as the open circles). However, at PAC number 52, the assurance factor is ~95%, and the estimated error of 0.05 is larger than the *Threshold* = 0.025.

Table 6.4: Summary of proposed CDS performance for the proof concept case study on Arrythmia.

CDS Global actions	False Alarm (1-Specificity) %	Diagnosis error (1-Sensitivity) %	Total error %	If Healthy	If Arrhythmia
Focus level 1	0	18	8.3	Next global action	Alarm, treatment actions
Focus level 2 (Male or female)	0	10	4.6	Next global action	Alarm, treatment actions

As the focus level can be increased (see Algorithm 6.1) for branches male and female (see eq. (6.1), Table 6.2: male data: 203 and female data: 249 > *umin*=200), the executive sends internal commands to the perceptor to increase the focus level. The same training procedure is done for creating models and actions library in Focus level 2 for the perceptor and the executive, respectively (see Algorithms 6.2 and 6.3, and section 6.2.2).

The CDS rechecks the User, depending on whether the User is male or female (see Algorithm 6.4 and section 6.2.2: Prediction part). Figure 6.8a shows that the real final diagnosis error for the male Users is 5% at PAC number 34, and the estimated error is 0.02, which is less than 0.025, the predefined threshold. Therefore, if all the measured features of a male User are normal, then the system can claim him as a healthy User and can send him to receive healthy living and disease prevention recommendations.

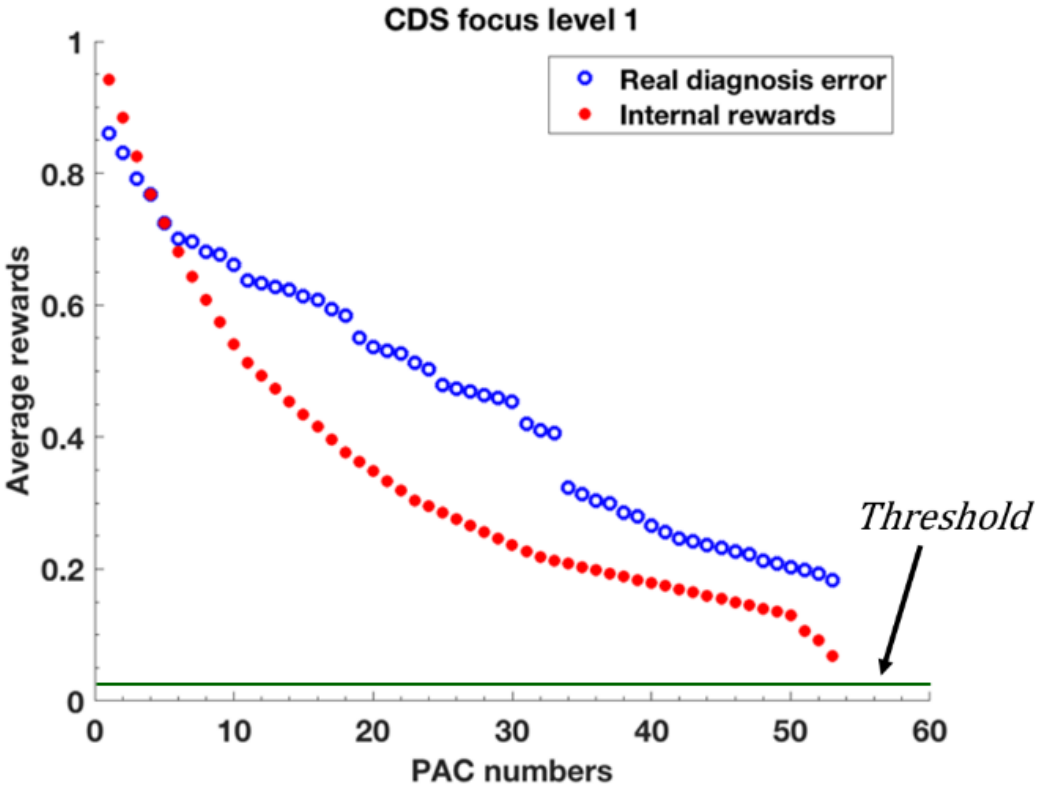


Figure 6.7: Internal rewards vs. real diagnosis error, in focus level 1 using the LOOCV approach [1.37].

Figure 6.8b shows that the real diagnosis error (open circles) for female Users is 14%. In addition, the estimated error for female Users using the assurance factor is ~0.18 (filled circles), which is higher than 0.025 as the predefined threshold. Also, due to the lack of the number of available instances in the Arrhythmia database, the focus level cannot increase to 3. Therefore, for the female Users, if the CDS do not find any disease class in the female User signals, then the female Users are sent for the screening process.

In Table 6.4, the total real diagnosis accuracy improved from 82% to 90% by increasing Focus level 1 to Focus level 2 for the low false alarm policy (*FA*=0). In addition, the total accuracy improved from 91.7% in Focus level 1 to 95.4% in Focus level 2. Another important improvement is the reduction in required number of PAC. The reduction of the maximum required PAC numbers can be seen by comparing Fig. 11 and Figures 6.8a and 6.8b. In Focus level 1 (if the system does not know about the User's sex), we needed 53 PACs. However, in Focus level 2 for males, number of PACs

is reduced from 53 to 34. Similarly, the required PACs for female Users are reduced to 29. Thus, the usage of the sensors can be reduced in Focus level 2.

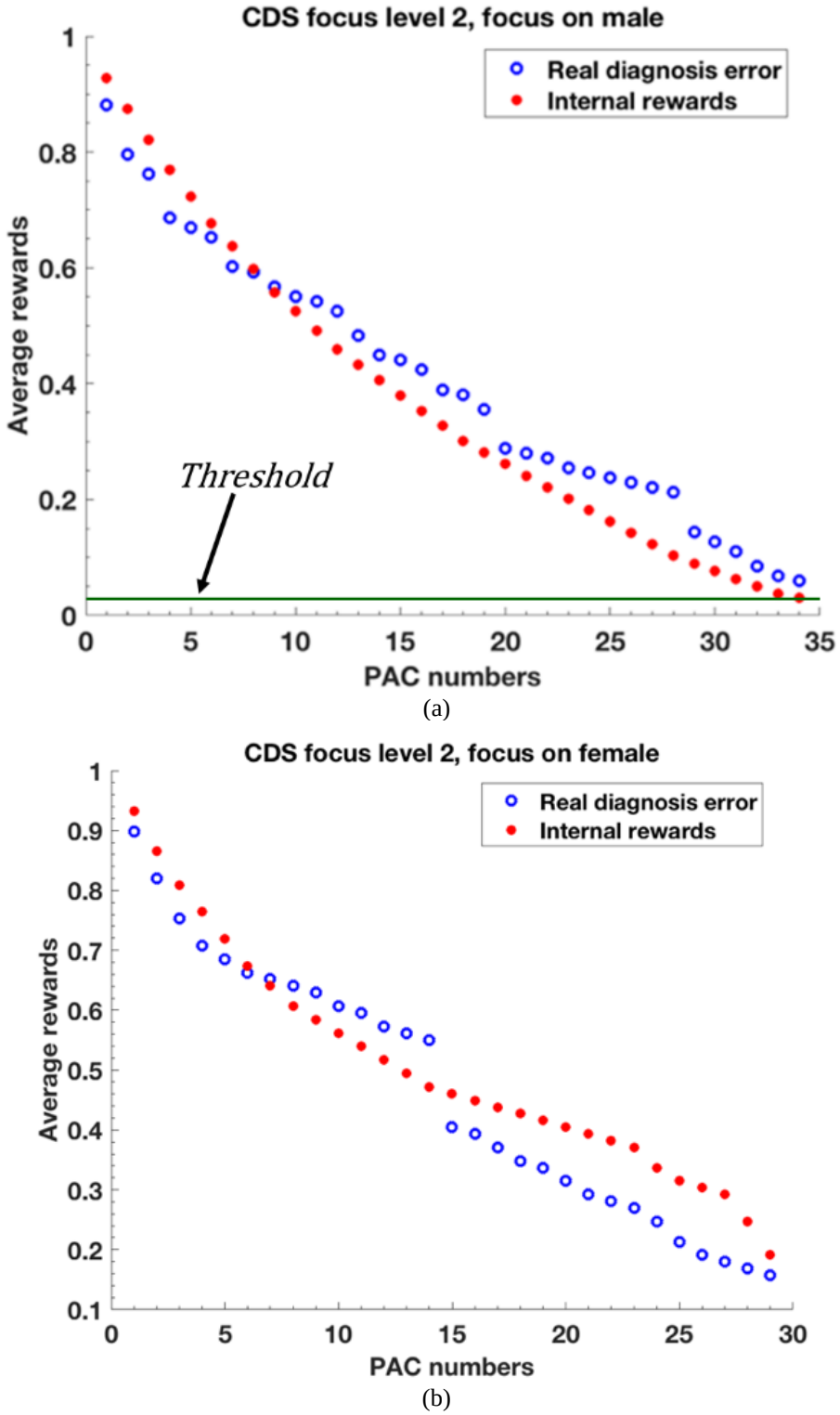


Figure 6.8. Focus level 2, internal rewards vs. real diagnosis error, using the LOOCV approach (a) male, (b) female [1.37].

In Figures 6.7 and 6.8, there are some breaks in the real error graphs (open circles). The actual error is unknown in the proposed CDS during running diagnostic test in all PACs. In addition, the estimated error (filled circles) using the internal reward (1-AF) is an approximation of the real diagnosis error that cannot follow these breaks in the real diagnosis error (open circles). By seeing the description of Algorithm 6.4 and the high-level algorithm presentation in section 6.2.2, the executive does the actions to check if the User has a certain disease class, for example, the second disease class of Arrhythmia (hd_2). If the User does not have the disease class hd_2, then the CDS will check if the User has the disease hd_3 with other actions in action space for hd_3. This process continues until the disease class hd_H (here, for this Arrhythmia database, it is $hd_{H=13}$). For example, the first best action that is selected for diagnosing the disease class of hd_3 may perform better than the last action for diagnosing the disease class hd_2. Therefore, the breaks in the real diagnosis error are due to switching for diagnosing different Arrhythmia classes.

6.3.2. Proposed Algorithms complexity and run time

It is useful to compare the complexity of the proposed method in this chapter with the techniques presented in Table 1.3. We run all algorithms on a Microsoft Surface-Pro with Intel ® Core ™ i706650U CPU @ 2.20GHz 2.21 GHz, 16 GB RAM, system type 64-bit Operating System x64-Based processor using MATLAB. For the proposed algorithms 6.1-6.4, the runtimes are between 2 to 10 minutes. However, this runtime for the proposed CDS is about 35.8 seconds (including LOOCV for 452 times modeling and decision making for two focus levels). Therefore, the proposed algorithms using the CDS concept are much faster than the typical methods such as support vector machine [1.43]. As a result, the proposed algorithms need less than 80 ms to do real-time modeling in two focus levels and to make a decision about a User in a smart e-Health home. Besides, the proposed algorithms are fast enough for real-time applications in a smart e-Health home.

6.4. Conclusions

In recent years, efforts were made to develop an autonomic computing layer in a smart e-Health home. However, currently, there are no comprehensive autonomic decision-making system (ADMS) or standards for the smart e-Health home autonomic computing layer. Practically, the ADMS functions needed are for real-time dynamic training or decision-making, screening, treatment, healing tracking as well as healthy living recommendations. In this chapter, we proposed the first step for an ADMS based on a cognitive dynamic system (CDS) for running the diagnostic test and health situation understanding with a low false alarm policy.

The system architecture and algorithms are presented for health situations (i.e., healthy or unhealthy) diagnosis with low false alarm policy. To illustrate the application of our proposed system, a proof-of-concept case study is done on the Arrhythmia database. Our system provided an acceptable total accuracy of 95.4% that was achieved by increasing the focus level of the CDS. Also, for the proposed Arrhythmia case study, the run time of our algorithms is less than 80 ms for a User of the smart e-Health home. This time included the time for the real-time modeling in two focus levels and for decision making.

The smart e-Health home can be implemented using IoT technology as a more generalized form of a cyber-physical system. In this scenario, the smart e-Health home using IoT technology must consider issues related to fog and edge computing and latency. Furthermore, the proposed, low complexity fast algorithms for the autonomic computer layer do not intensify these IoT technology issues because in the example provided, less than 80 ms was required time for training and prediction.

For the Arrhythmia case study as a proof-of-concept example, we used the database in which some key features such as heart rate were missing. It should be noted that missing key features can also be regarded as sensor(s) failure. Therefore, we could simulate the proposed CDS flexibility and reliability in the presence of the sensor(s) failure. Further, the 95.4% accuracy shows that the proposed CDS can find alternative actions for the relevant diagnostic tests.

In summary, for implementing the CDS, the following concepts are implemented in this chapter: decision-making tree, inspiration from medical doctors (MDs) decision-making approach, converting data in the database to knowledge, prediction using the Bayesian model, and the characteristics of non-Gaussian and non-linear health features. These concepts are used in the presented CDS algorithms. The CDS can check one feature in each perception-action cycle (PAC). Checking one feature in each cycle makes the proposed algorithms simple and fast. Therefore, the presented algorithms are well suited for a real-time smart e-Health home or a future robotic nurse. Finally, this chapter is the first step for designing the ADMS for a smart e-Health home as the platform that can be extended for different healthcare policies such as screening process or diagnosing the disease class.

Chapter 7

CDS V5 FOR SCREENING IN HEALTHCARE APPLICATIONS IN PRESENCE OF DEFECTIVE DATASET[6]

In recent years, there has been a growing interest in smart e-Health systems to improve people's quality of life by enhancing healthcare accessibility and reducing healthcare costs. Continuous monitoring of health through the smart e-Health system may enable automatic diagnosis of diseases like Arrhythmia at its early onset that otherwise may become fatal if not detected on time. In this work, we developed a cognitive dynamic system (CDS)-based framework for the smart e-Health system to realize an automatic screening process in the presence of a defective dataset. A defective dataset may have poor labeling and/or lack enough training patterns. To mitigate the adverse effect of such a defective dataset, we developed a decision-making system that is inspired by the decision-making processes in humans in case of conflict of opinions (CoO). We present a proof-of-concept implementation of this framework to automatically identify people having Arrhythmia from the single lead Electrocardiogram (ECG) traces. It is shown that the proposed CDS performs well with the diagnosis errors of 13.2%, 9.9%, 6.6%, and 4.6%, being in good agreement with the desired diagnosis errors of 25%, 10%, 5.9%, and 2.5%, respectively. The proposed CDS algorithm can be incorporated in the autonomic computing layer of a smart-e-Health-home platform to achieve a pre-defined degree of screening accuracy in the presence of a defective dataset.

[6] All symbols and notations are defined in this chapter are valid just for this chapter. The similar notations or symbols in other chapters are defined inside the chapters.

The chapter is organized as follows. In section 7.1, the introduction on CDS v5 as CDS with CoO decision making is presented. Section 7.2 presents the reason for using CDS v5 for screening process instead of earlier versions of CDS presented in chapters 3-6. In section 7.3, we discuss ADMS based on CDS v5 for automatic screening process. Finally, in section 7.4, simulation results and discussions are provided for case study screening process for someone with or without arrhythmia. we summarize the main contribution of our work in section 7.5.

7.1. CDS v5

In this chapter, we present a cognitive dynamic system (CDS) for the screening process in smart e-Health systems based on the perception and multiple action cycles (PMAC) and the decision-making processes in humans in case of a conflict of opinion (CoO).

In chapters 3-5, a CDS was proposed for smart fiber optic communication systems to demonstrate its high precision decision-making ability in complex smart systems. It should be mentioned that we use the term cognitive decision making (CDM) to define decision making using CDS. The CDS was presented in chapters 3-6 as an enhanced-AI that exploited the maximum probability (MAP) approach for the CDM. The CDS thus implemented resulted in a high-performance CDM, which however used a reliable dataset to train the model. When the datasets are not reliable due to poor labeling and/or insufficient training patterns, the PAC-based CDS cannot perform well enough to satisfy requirements. This can be explained with an analogy to the decision-making process of the human brain where it makes a judgment based on some ambiguous information, thus running a risk of making a wrong decision. In this chapter, we propose a CDS algorithm to realize a reliable screening method in a smart e-Health system from a defective dataset. Here we exploited the concept of CoO to realize the CDM for the NGNLHE system, unlike the CDM implemented earlier for LGE systems. We also generalized the concept of PAC in the PAC-based CDS to perception-multi actions cycles (PMAC) to implement the CoO.

The basic model of a CDS based on PMAC is presented in Figure 7.1, which includes three main subsystems in a CDS: (1) Perception by the Perceptor; (2) a

Feedback channel for sending the multiple raw internal rewards; and (3) the Executive to perform multiple actions on the environment.

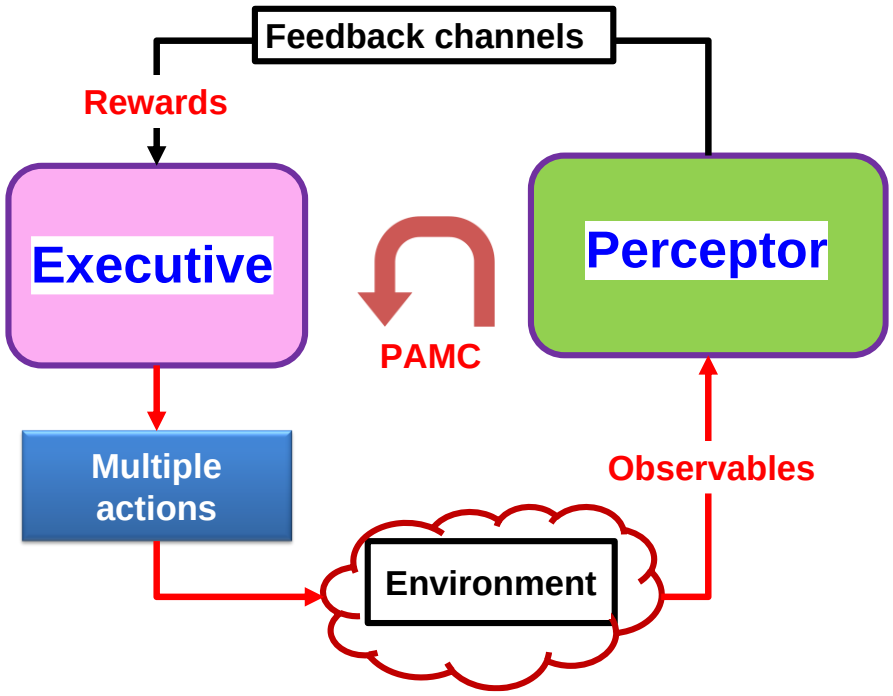


Figure 7.1: Block diagram of a cognitive dynamic system (CDS) based-on PMAC: Perception multiple actions cycle.

The main contributions of this work can be summarized as follows:

1. An improved CDS with the PMAC and CoO-based CDM is proposed for an NGNLHE system in contrast to the PAC-based CDS for the LGE system based on PAC and MAP-based CDM. This improvement allows for the mitigation of screening inaccuracies due to a defective dataset.
2. The structure of a PMAC-based CDS is designed as the first stage of the screening process. In this paper, we aim to screen human health automatically between two binary (healthy and unhealthy) states based on their single lead ECG traces. We implemented a CoO-based CDM in the second stage to achieve a desired level of diagnosis error at an acceptable high false alarm rate. The first stage was based on the diagnostic test or low false alarm policy presented in chapter 6 using a PAC-based CDS.

3. Algorithms for decision making between healthy and unhealthy conditions in a NGNLE system is presented based on the screening process.
4. A proof-of-concept case study is presented in which a PMAC-based CDS is applied to screen for Arrhythmia from a defective dataset. It is shown that the proposed CDS performs well, giving good agreement with the desired diagnosis errors of 25%, 10%, 5.9%, and 2.5%, achieving average final diagnosis errors of 13.2%, 9.9%, 6.6%, and 4.6%, respectively. These diagnosis errors correspond to a clinically acceptable false alarm rates [19] of 20.1%, 25%, 28.4%, and 54.7% respectively, even with a defective dataset.

7.2. Why a cognitive dynamic system with CoO?

In the earlier implementations of CDS in chapters 3-6, the PAC was realized by combining conventional machine learning (ML) approaches, such as reinforcement learning (RL), and supervised learning (SL). Here, we focus on how a PAC-based CDS and the proposed CDS in this chapter can overcome the weaknesses of SL and RL.

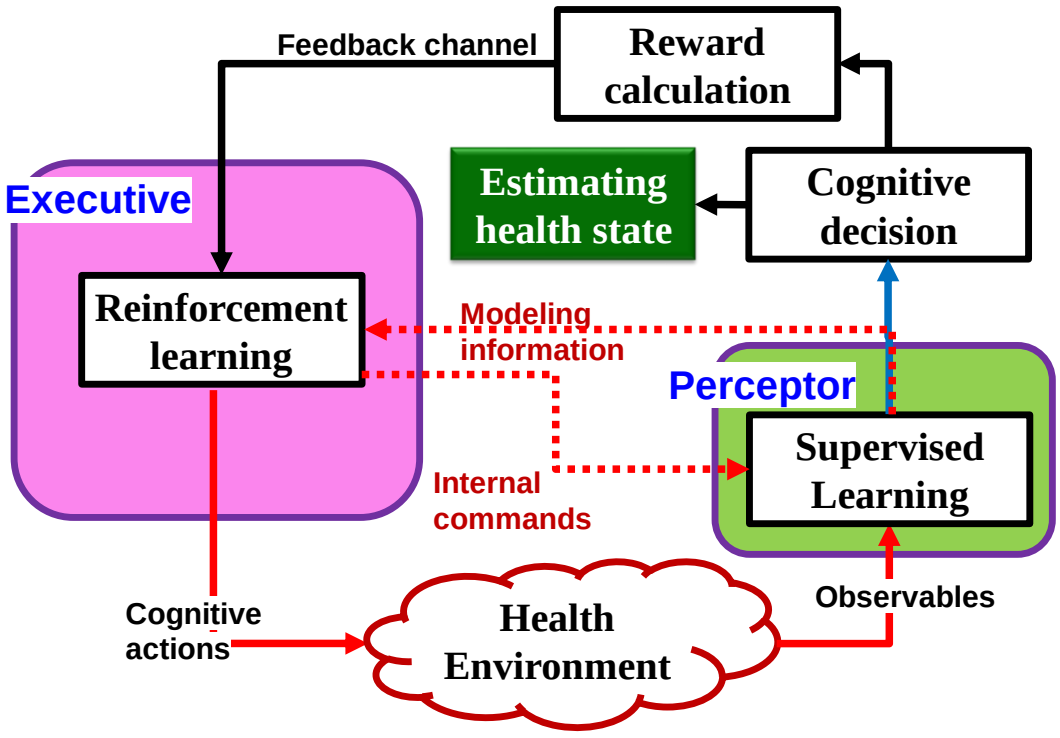


Figure 7.2: Conceptual implementation of PAC-based CDS

7.2.1. Typical PAC-based CDS

A PAC-based CDS can be implemented by combining both SL and RL techniques of conventional machine learning. Figure 7.2 shows the block diagram of a PAC-based CDS for the healthcare environment. The perceptor of the CDS can extract a model using the SL algorithm. The perceptor then generates an internal reward and predicts the outcome of the dynamic environment (human health in this case) using the extracted model [1.8]. The executive receives the internal reward from the Perceptor through the feedback channel. The executive is built upon an RL-based ML approach that is based on the internal reward in the current PAC finds an action, which can optimize the internal reward for the next PAC. The internal reward gives the CDS self-awareness, self-consciousness, and independence from the dynamic environment. In short, the RL-based executive of a PAC-based CDS uses the internal reward produced by the model extracted in the SL-based perceptor to apply a cognitive action on the dynamic environment. Therefore, a PAC-based CDS can be considered as an enhanced AI.

7.2.2. Proposed PMAC-based CDS

In a typical PAC, the CDS applies an action on the environment and then uses the calculated reward (internal/external) to gain experience. The RL in the Executive then optimizes the reward in the following PAC by finding the most appropriate action. As mentioned before, the Perceptor of a PAC-based CDS uses SL to extract a model of the environment that requires a well-labeled dataset with enough number of training patterns to enhance the reliability of the model. For example, in the case of orthogonal frequency division multiplexing (OFDM) long-haul fiber optic communication systems (see section 3.2.1) when the CDS operates in the bit error rate (BER) improvement mode, the internal rewards and model converges after $N=512$ frames.

Unlike fiber-optic communications, where training data with accurate labeling are available at a much faster rate, the number of training patterns in the available datasets for healthcare applications is generally limited. In addition, the labeling of the healthcare dataset can be erroneous owing to its dependence on human skills. Moreover, a dataset can be inherently defective or manipulated by hidden cyber-attack. These shortcomings in the available data may result in an overfitted model, potentially

causing the test accuracy of the model to drop significantly compared to the training accuracy. In such a case, one can infer that the used dataset for model extraction is badly labeled and/or lacks enough training patterns required to extracting a reliable, accurate and converged model. Without a converged model in a PAC-based CDS, the SL-based perception process based on PAC and internal reward generation can limit the accuracy and consistency of the performance of the Executive in the case of healthcare applications.

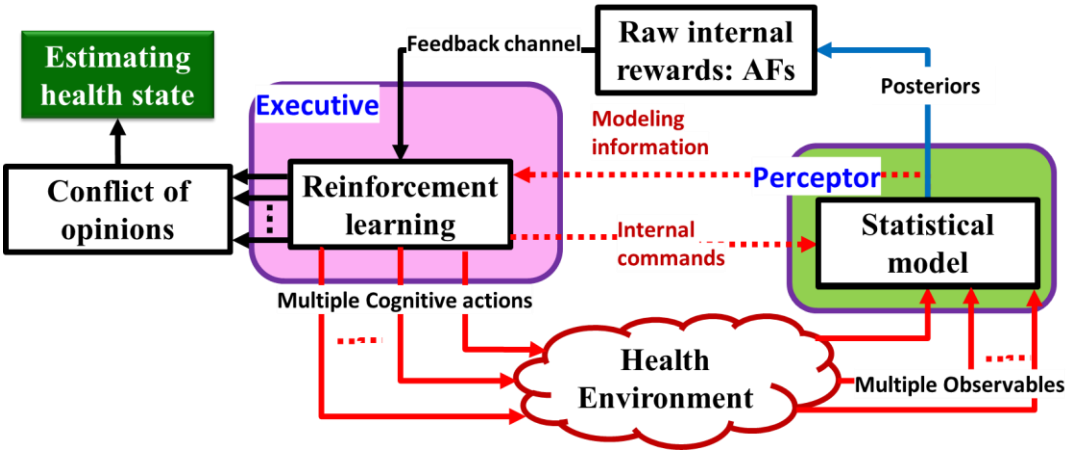


Figure 7.3: Conceptual implementation of PMAC based CDS

In this chapter, we extended the typical PAC to the proposed PMAC (Figure 7.3) in an attempt to mitigate the inherent deficiency of SL. In the PMAC-based CDS, the Executive applies multiple actions simultaneously on the health environment in one cycle, for example, activates different ECG leads, or the Photoplethysmogram (PPG) sensor. The Perceptor receives new measurements from the environment to extract model(s) and uses the model(s) to generate multiple raw internal rewards for the Executive. In the following cycle, the Executive performs multiple actions on the environment to optimize each internal reward based on some pre-defined policies by the users. The Executive also provides weighted internal rewards using the cost-to-go-functions based on the received raw internal rewards through the feedback channels. These weighted internal rewards can be used by the CoO for CDM (Fig. 4).

For the screening process, the PAC-based CDS initially assumes that the user is ‘Unhealthy’. In the case of a ‘Healthy’ user, the CDS applies some actions on the environment (user body) and finds the relevant evidence of the user being ‘Healthy’. Then, the CDS feeds the evidence i.e. the newly measured parameters from the environment (user body) into the extracted model in the Perceptor and changes the decision about the user's health state to ‘Healthy’. Similar to the PAC-based CDS, the PMAC-based CDS also assumes an ‘Unhealthy’ state as the initial decision for the user's health. However, unlike the PAC-based CDS, the Executive of the PMAC-based CDS applies two actions simultaneously for two states - ‘Healthy’ and ‘Unhealthy’ - on the environment (user body), to find evidence in favor of each state separately.

In the case of conflicting outcomes, when the CDS has evidence in favor of both states, the amount of evidence in favor of a particular state plays a crucial role in resolving the conflict in decision making. For example, 100 evidence in favor of ‘Healthy’ state with 99% assurance provides a final reward of 99, while 105 evidence in favor of ‘Unhealthy’ state with 95% assurance results in a final reward of 99.75. Thus, the final decision of the CDS about the user under test (UUT) would be ‘Unhealthy’, as this state results in a higher reward than the ‘Healthy’ state. In this work, we termed this method of decision making as the Conflict-of-Opinions (CoO).

In summary, unlike a PAC-based CDS, the Executive of the proposed CDS uses multiple evidence generated from the model extracted by the Perceptor and applies multiple cognitive actions on the dynamic environment. The internal rewards generated in favor of every possible decision state give the PMAC-based CDS information to make a final decision by comparing among the rewards, giving the CDS self-awareness, self-consciousness about the dynamic environment, and independence from a defective dataset. The PMAC-based CDS thus has the “conscience” about the actions and non-reliability of the extracted model by the Perceptor. To generate multiple internal rewards, a CoO-based decision-making algorithm is applied in the Executive. In contrast, a PAC-based CDS uses the MAP rule for decision making based on a single internal award at each PAC, relying on the dataset and extracted model by the SL-based Perceptor. Therefore, the PMAC-based CDS is a more appropriate choice rather than a

PAC-based CDS in an intelligent healthcare screening process, where the SL cannot extract the converged reliable model from the defective dataset.

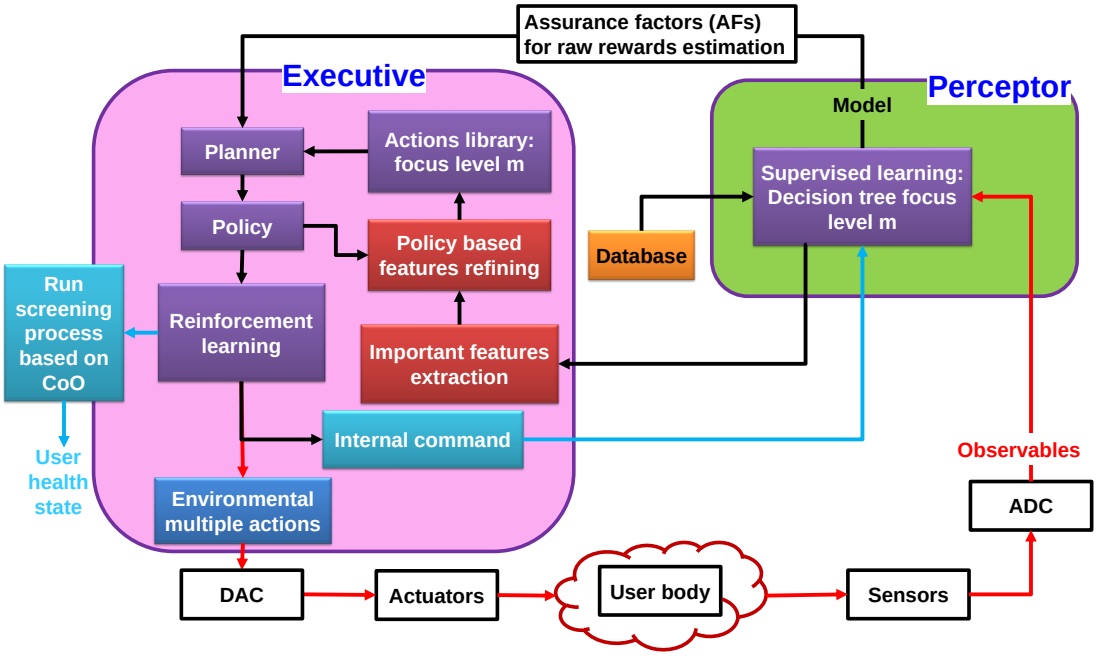


Figure 7.4: Block diagram of proposed CDS architecture for the ADMS of a smart e-Health home

7.3. Proposed ADMS using CDS architecture and algorithms

In this section, we describe the proposed CDS architecture and algorithm using a CoO-based decision-making approach for the screening process. The detailed architecture of the proposed CDS for health screening is shown in Figure 7.4. Similar to the PAC-based CDS, a PMAC-based CDS has two main subsystems: (i) the Perceptor, and (ii) the Executive with a feedback channel linking them and it operates in three modes: (i) training (ii) prediction and (iii) steady-state.

During the training mode the data and information collected from the dataset will be converted to a 'model library' in the Perceptor and an 'action library' in the Executive. Also, the CDS in the training mode dynamically updates the knowledge in the Perceptor and the 'action-library' in the executive when the database is updated through the e-Health network. This update can be done in real-time, in parallel to the prediction mode, or during the steady-state mode.

The CDS will use the knowledge in the Perceptor 'model library' for prediction of the health conditions of users with unknown health states. In the case when an anomaly in the user's health is detected or a request is placed by the user, the CDS can initiate the prediction mode. The trained CDS collects evidence about the user's health state and uses that information from the user to reason and predict the health state of the user and to determine whether any action is required based on a pre-defined set of policies and objectives. The policy can be, for example, a certain level of false alarm rate is acceptable in screening between the 'Healthy' and 'Unhealthy' states.

When the CDS makes a final decision based on current measurements, or if there is no new request from the user, it goes into the steady-state mode. However, in specific situations or upon receiving the user's request, the CDS will switch from steady-state to prediction mode. In brief, the following modes and functionalities constitute the proposed CDS as shown in Figure 7.4.

I. CDS training mode
   a. Perceptor training mode.
   b. Executive training mode.

II. CDS prediction mode
   a. Feedback channel (multiple assurance factor calculation).
   b. Executive (planner).
   c. Executive (policy).
   d. Executive (reinforcement learning & rewards calculation).
   e. Executive (CoO-based CDM for screening process).

7.3.1. Training mode: Perceptor and Executive

We exploited the decision tree to extract the posterior in CDS applied to NGNLE systems (5.3.1 and 6.2.3). This concept of posterior extraction using decision trees is extended in the proposed PMAC-based CDS. Extracting the posterior using decision trees is a common method in machine learning. The CDS training mode can be summarized in the following four parts:

a. Creating a jungle of decision-making trees.
b. Extracting the posterior

c. Extracting the knowledge- and action-space from the database
d. Action refinement in the Executive library based on predefined policy.

Initially, when there exists no relevant model in a model library, the four-layered Bayesian modeling in the Perceptor extracts the statistical model of the system (see [10] for further details) using decision trees. The Bayesian modeling consists of four layers for an arbitrary focus level *m.* Here, in this chapter decision tree level *m* is considered as the focus level *m* for the CDS.

Layer I: Normalization

The extracted features from the measured physiological signals could have any value, and hence a large number of discretized cells are required for saving vacant spaces. For example, the output of the ECG signal can be represented as a set of features,

$$O = [\, feature(1), \ldots, feature(l), \ldots, feature(L)]. \tag{7.1}$$

where, L is number of extracted features from the dataset. Therefore, we normalize all extracted features as

$$\bar{O}(l) = \overline{feature}(l) = \frac{feature(l)}{\max\{abs(b^l_{\min}),\ abs(b^l_{\max})\}} \tag{7.2}$$

Here, $b^l_{\min}$ and $b^l_{\max}$ are the minimum and maximum value of the l_{th} feature. The data normalized in this way is sent to Layer II as $\bar{O}$ for further processing. Also, in this layer, the perceptor calculates L_m as the number of global nodes at focus level *m*, and $1 \le m \le M$ using permutation with repetition equation as:

$$L_m = \binom{L}{m} \tag{7.3}$$

For example, in Figure 7.5 we have five extracted features in the dataset [sex, age, height, breathing rate (BR), and heart rate (HR)]. Therefore, using eq. (7.3) at focus level 1, the number of global nodes is five. Similarly, the number of new global nodes is 10 at focus level 2.

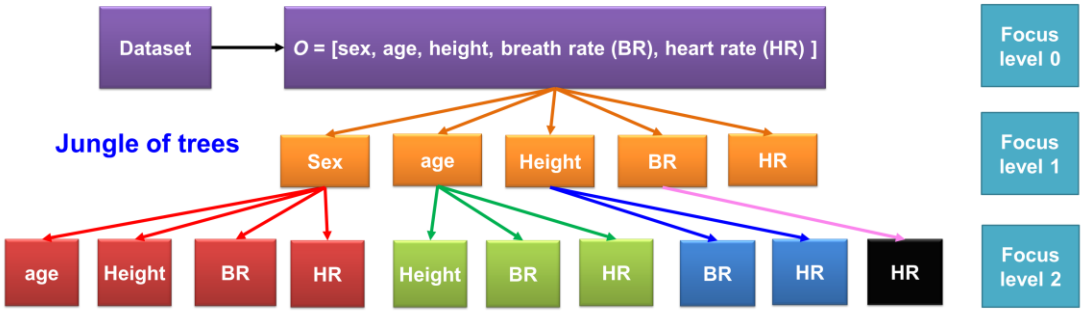


Figure 7.5: Example of global nodes of jungle of trees (here, 5 trees)

Layer II: Creating jungle of trees

In Layer II, the normalized values related to features are discretized with discretization steps Δx_i^k, and $1 \le i \le m$. Here, *k* is the PAC number and *m* is the current focus level. For simplicity, we define a *discretization factor* (DF_i^k) of features for focus level *i* as:

$$DF_i^k = 10\Delta x_i^k, \quad 1 \le i \le m \tag{7.4}$$

$$\mathbf{DF}^{k,m} = [DF_0^k, DF_1^k, \cdots, DF_i^k, \cdots, DF_m^k], \tag{7.5}$$

$$N_x^{k,i,l} = \frac{10(x_{\max}^{k,i,l} - x_{\min}^{k,i,l})}{DF_i^k}, \tag{7.6}$$

$$F_i^k = \sum_{l=1}^{L_i} N_x^{k,i,l}, \quad 1 \le i \le m,\ F_0^k = 1 \tag{7.7}$$

Here, $N_x^{k,i,l}$ is the number of discretized cells in the *x*-axis for feature number *l* as $\hat{\mathrm{O}}_{DF}^{k,i}(l_i) = \overline{feature}(l_i, DF_i, i)$. Here, $\hat{\mathrm{O}}_{DF}^{k,i}(l_i)$ is discretized feature *l* at focus level *i* with discretization factor DF_i at *k*th PAC. Also, $x_{\min}^{k,i,l}$ and $x_{\max}^{k,i,l}$ are the minimum and maximum of the normalized feature $(abs(x_{\min}^{k,i,l}) \le 1,\ abs(x_{\max}^{k,i,l}) \le 1)$. In this chapter, there is no discretization along y direction.

Furthermore, F_i^k corresponds to the number of decision- trees branches (eqs. (7.3) and (7.7)). For the desired complexity threshold, the CDS can update the current focus level *m*, and $\mathbf{DF}^{k,m}$. Then, Layer II will send the normalized set of features $\bar{\mathrm{O}}^{k,m+1}$, the discretized set of features $\hat{\boldsymbol{O}}_{DF}^{k,m}$ and $\mathbf{DF}^{k,m}$ to Layer III (eqs. (7.2) and (7.8)).

$$\hat{\boldsymbol{O}}_{\boldsymbol{DF}}^{k,m} = [\hat{O}_{DF}^{k,m}, \hat{O}_{DF}^{k,m-1}, \ldots, \hat{O}_{DF}^{k,i}, \ldots, \hat{O}_{DF}^{k,1}],$$
$$\hat{\mathrm{O}}_{DF}^{k,i} = [\overline{feature}(1, DF_i, i), \ldots, \overline{feature}(l_i, DF_i, i), \ldots, \overline{feature}(L_i, DF_i, i)].$$
$$\bar{\mathrm{O}}^{k,m+1} = [\overline{feature}(1, m+1), \ldots, \overline{feature}(l_{m+1}, m+1), \ldots, \overline{feature}(L_{m+1}, m+1)]. \quad (7.8)$$

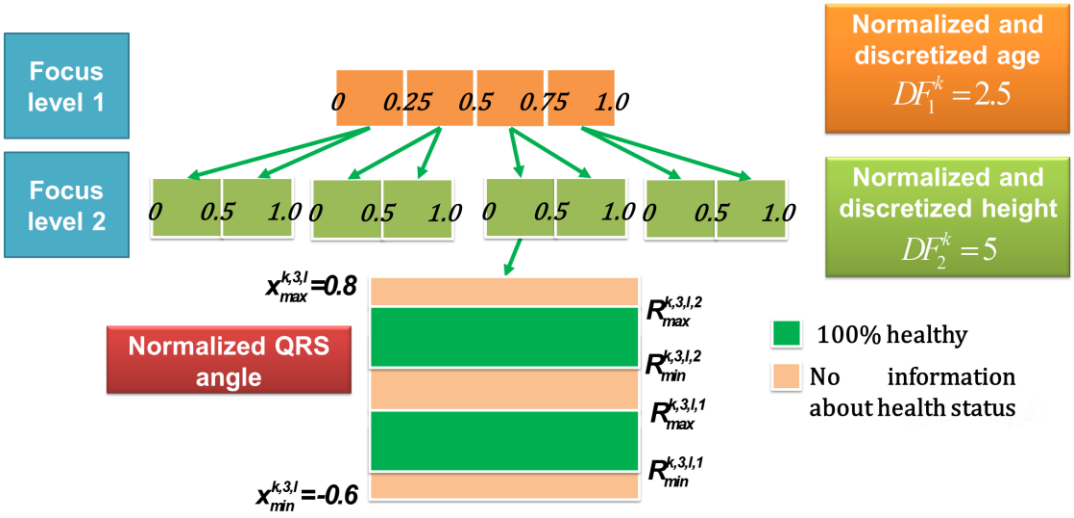


Figure 7.6: Example of decision boundaries extraction and action-library creation.

It should be noted that the required memory for saving models and action-space will increase as the number of focus levels increases in the CDS. Thus, to address the increasing algorithms complexity when the number of focus levels increase, we can define a bound for the maximum possible focus level as the *Complexity threshold*. Then, we can calculate the total acceptable branches of the decision tree as:

$$F_m^{total,k} = \prod_{f=2}^{m} F_i^k, \qquad m \in \{1, 2, \ldots, M\}$$
$$\text{and, } F_m^{total,k} \leq Complexity\ threshold \quad (7.9)$$

Furthermore, $F_m^{total,k}$ corresponds to the maximum number of tree branches at focus level *m* and perception action cycle *k*. For the desired predefined *Complexity threshold*, the CDS cannot increase the focus level more than *M*.

Layer III: Estimation of probabilities discretization

In this layer, the system estimates the probability of a $\mathbf{HD}_n^k \in \{"Healthy","Unhealthy"\}$ for a given set of features, $\mathbf{O}^{k,m+1}=[O^{k,1},O^{k,2},...,O^{k,i},...,O^{k,m},O^{k,m+1}]$, i.e. $P(\mathbf{HD}_n^k \mid \mathbf{O}^{k,m+1})$ by approximating it as the probability of $\tilde{\mathbf{O}}^{k,m+1}=[\bar{\mathrm{O}}^{k,m+1},\hat{\boldsymbol{O}}^{k,m}]$ (see eq. (7.8) and Figure 7.6) for a given $\mathbf{HD}_n^k$, using the Monte-Carlo method and a jungle of decision trees as $P(\mathbf{HD}_n^k \mid \tilde{\mathbf{O}}^{k,m+1})$, where *n* is the discrete time. The extracted model in this layer can be called the *Bayesian posteriori*. Similarly, the system can estimate the *evidence* $P(\mathbf{O}^{k,m+1})$ as $P(\tilde{\mathbf{O}}^{k,m+1})$. These estimated probabilities are saved in the Perceptor 'model library' for future use. The Executive will use the extracted statistical probabilities for prediction by receiving them through the internal feedback channel to evaluate the actions virtually before applying them on the environment and creating the 'action-library'.

Layer IV: Creating action-library and extraction of evidence boundaries

In this layer, based on the applied system policy, the Executive estimates the decision boundaries and creates the action-library. The action-library is the library of all possible actions the Executive can take to make a screening decision about the UUT. Here we describe the process how the CDS finds the evidence that a UUT is healthy with an assurance of 100%. We use eq. (7.10) to find the decision boundaries, which provide an assurance of 100% in favor of the 'Healthy' or 'Unhealthy' states (Figure 7.6).

$$P(\mathbf{HD}_n^k \mid R_{\min}^{k,m+1,l,b} \le \bar{\mathrm{O}}^{k,m+1}(l_{m+1}) \le R_{\max}^{k,m+1,l,b}, \hat{\boldsymbol{O}}_{\boldsymbol{DF}}^{k,m}) = 1 \tag{7.10}$$

Here, $b \in \left\{1,2,...,B_{m+1}^l\right\}$ and B_{m+1}^l is the total number of desired actions required for achieving a decision boundary containing evidence of 100% 'Healthy' state (or 'Unhealthy' state) at the focus level (*m*+1) and branch *l* (see Figure 7.6). The evidence boundaries of $R_{\min}^{k,m+1,l,b}$ and $R_{\max}^{k,m+1,l,b}$ will be saved in the model-library of the Perceptor. In addition, the precision factor (PF) or the number of users within the decision boundaries (Healthy or Unhealthy) of the features are

$$PF_{b,m+1}^{k,l} = U_t \times P(R_{\min}^{k,m+1,l,b} \le \bar{\mathrm{O}}^{k,m+1}(l_{m+1}) \le R_{\max}^{k,m+1,l,b}, \hat{\boldsymbol{O}}_{\boldsymbol{DF}}^{k,m}) \tag{7.11}$$

Here, U_t is total number of users in the training dataset. Therefore, $PF_{b,m+1}^{k,l}$ is the number of training patterns between $R_{\min}^{k,m+1,l,b}$ and $R_{\max}^{k,m+1,l,b}$ that are all 'Healthy' or 'Unhealthy'. The Executive keeps a record of $PF_{b,m+1}^{k,l}$, $[\hat{O}_{DF}^{k,1}(l_1),\ldots,\hat{O}_{DF}^{k,i}(l_i),\ldots,\hat{O}_{DF}^{k,m}(l_m),\bar{O}_{DF}^{k,m+1}(l_{m+1})]$ and information about the sensors required to extract the feature set of $[\hat{O}_{DF}^{k,1}(l_1),\ldots,\hat{O}_{DF}^{k,i}(l_i),\ldots,\hat{O}_{DF}^{k,m}(l_m),\bar{O}_{DF}^{k,m+1}(l_{m+1})]$ in the action-library. The action-space can be defined as follows:

$$C=\left\{c_k \middle| c_k=\begin{array}{l}1-\textit{Actuating sensor}\\ (\textit{Healthy / Unleathy evidence}):\\ \quad c^k(\tilde{\mathbf{O}}^{k,m+1})\\ 1.1-\textit{Time domain features}\\ 1.2-\textit{Frequencydoman features}\\ 1.3-\textit{Statistical features}\\ 2-\textit{Internal commands}:\\ \quad \textit{Increase } m,\ (m+1)<M\end{array}\right\}. \tag{7.12}$$

Also, all decision boundaries (DB) for action space can be shown as:

$$DB_{\textit{healthy or unhealthy}}=\left\{r_k^{m,l,b}\ \textit{for}\ c_k \middle| \begin{array}{c} r_k^{m,l,b}=[R_{\min}^{k,m+1,l,b},R_{\min}^{k,m+1,l,b}],\\ PF_{b,m+1}^{k,l}(r_k^{m,l,b}).\end{array}\right\}. \tag{7.13}$$

Here, c_k is the action at the kth PAC and m is the focus level. The action-space contains information about all possible actions that the Executive can perform during the prediction mode to provide evidence in favor of 'Healthy' or 'Unhealthy' state. Using the complexity threshold defined in eq. (7.9). Using eqs. (7.14)-(7.16), we can obtain the number of training users in each branch of decision trees:

$$U_{m+1}^{k,l}=U_t\times P(x_{\min}^{k,m+1,l}\le\bar{O}^{k,m+1}(l_{m+1})\le x_{\max}^{k,m+1,l},\hat{\boldsymbol{O}}_{\boldsymbol{DF}}^{k,m}) \tag{7.14}$$

$$U_{m+1}^{k,l}\ge U_{\min} \tag{7.15}$$

$$U_{\min}=\frac{5}{Threshold_{DE}} \tag{7.16}$$

Here, $U_{m+1}^{k,l}$ is number of training data in branch l of the decision tree and focus level (m+1). Also, in eq. (7.16), $U_{\min}$ is the minimum required number of users for the l_{th} branch and focus level (m+1) to enable extraction of a reliable model. In eq. (7.15), the

$Threshold_{DE}$ is a predefined desired diagnostic error. Then, the information related to the extracted decision trees is saved in the memories of the Perceptor and Executive. The procedure for finding an evidence of 100% assurance that the UUT is 'Unhealthy' follows a similar approach as described above.

7.3.2. Prediction mode: Calculating raw internal rewards in feedback channel

The estimation process of internal reward (or diagnosis error) for the proposed CDS can be explained with help of Figure 7.7. The estimation process of false alarm follows a similar approach. As mentioned earlier, the initial decision of the CDS about the UUT (U_{test}) is 'Unhealthy'. The CDS performs some actions on the environment and uses the CoO to reach a final decision. Once the CDS finds stronger evidence in favor of the 'Healthy' state than the 'Unhealthy' state, it changes the decision to 'Healthy'. For example, in the case of $PF = 4$, there are 4 healthy users from the training dataset whose data remain with a range of r, i.e. within the range, r all 4 users from the training dataset are healthy. Now if the data of the UUT falls within the range r, the conditional probability error (estimated diagnosis error) for a decision of 'Healthy' would be 0.2.

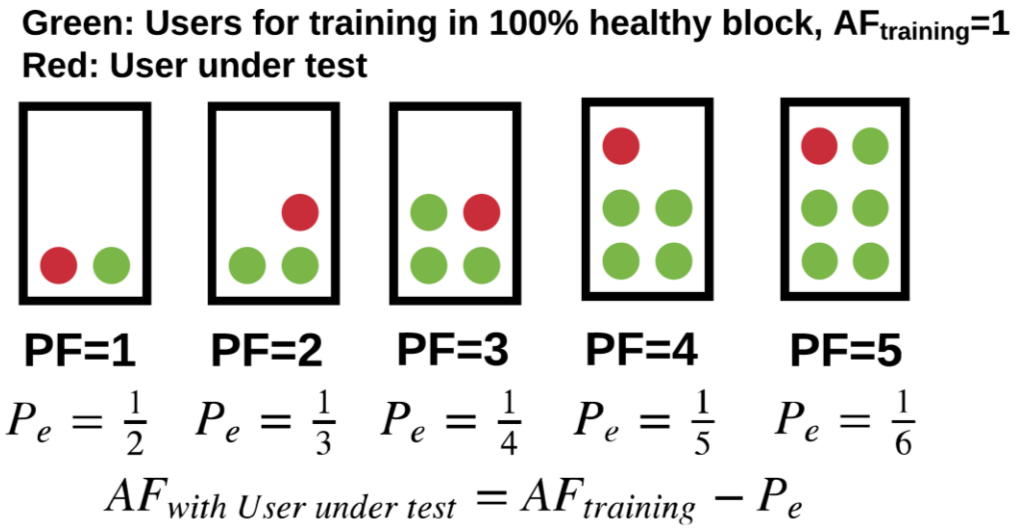


Figure 7.7: Example for diagnosis error (DE) and False alarm (FA) estimation; precision factor (PF)

In general, we can define the AF for the UUT as 'Healthy' (defining the AF for the UUT as 'Unhealthy' follows similar procedure) using,

$$P(U_{test} = healthy \mid R^{k,m+1,l,b}_{\min,healthy} \le \bar{O}^{k,m+1}_{U_{test}}(l_{m+1}) \le R^{k,m+1,l,b}_{\max,healthy}, \hat{\boldsymbol{O}}^{k,m}_{\boldsymbol{DF},\boldsymbol{U}_{test}}) = \frac{PF^{k,l,healthy}_{b,m+1}}{PF^{k,l,healthy}_{b,m+1}+1}, \tag{7.17}$$

$$AF^{k}_{DE} = \frac{PF^{k,l,healthy}_{b,m+1}}{PF^{k,l,healthy}_{b,m+1}+1}. \tag{7.18}$$

Here, AF^{k}_{DE} is the assurance factor in favor of the 'Healthy' state and *DE* stands for the estimated diagnosis error. Similar to AF^{k}_{DE}, the assurance factor for false alarm, AF^{k}_{FA} can be estimated at *k*th PAC as

$$P(U_{test} = unhealthy \mid R^{k,m+1,l,b}_{\min,unhealthy} \le \bar{O}^{k,m+1}_{U_{test}}(l_{m+1}) \le R^{k,m+1,l,b}_{\max,unhealthy}, \hat{\boldsymbol{O}}^{k,m}_{\boldsymbol{DF},\boldsymbol{U}_{test}}) = \frac{PF^{k,l,unhealthy}_{b,m+1}}{PF^{k,l,unhealthy}_{b,m+1}+1}, \tag{7.19}$$

$$AF^{k}_{FA} = \frac{PF^{k,l,unhealthy}_{b,m+1}}{PF^{k,l,unhealthy}_{b,m+1}+1}. \tag{7.20}$$

7.3.3. Prediction mode: Perceptor and simple Executive

The CDS initiates the prediction mode when an anomaly in the user's health is detected or a request is placed by the user. The trained CDS gathers evidence about the user's health state and uses that information to reason and predict the health state of the user. The CDS then determines whether any action is required based on a pre-defined set of policies and objectives. In this chapter, we use a simple Executive that performs basic multiple actions without considering permutation and probable complicated weighted

relationships between multiple actions, type of extracted features and internal commands.

The Executive is an essential part of any CDS. It is responsible for improving the decision-making accuracy by applying action on the NGNLHE. For example, the Executive can activate the actuators in the smart home or sends the internal commands to the Perceptor for changing the modeling configurations such as $\mathbf{DF}^{k,m}$. The Executive provides non-monotonic reasoning to the CDS by using the internal reward and changing its focus level. The Executive designed in this work includes three parts (see Figure 7.4): planner including the action-library, policy, and learning using a cost-to-go function (see sections 3.2.3, 4.2.3. 5.2.3 and 6.2.4).

Planner and policy

In a CDS, the policy is defined as the desired goals that the CDS attempts to achieve in each PAC during the screening process. Here, the goal is to achieve a pre-defined, $Threshold_{DE}$ of diagnosis error (DE) by minimizing the false alarm (FA) rates. Then, based on the $Threshold_{DE}$, the actions for providing evidence in favor of 100% healthy state in the action-space can be refined by the planner as:

$$PF_{b,m+1,healthy}^{k,l} \geq PF_{b,m+1,healthy}^{k,l,\min} = \frac{1}{Threshold_{DE}} - 1, \tag{7.21}$$

For example, if the predefined desired diagnosis error threshold is 10% at focus level 3, then the number of training patterns within the range with 100% 'Healthy' states should be $PF_{b,3,healthy}^{k,l} \geq PF_{b,3,healthy}^{k,l,\min} = 9$ or more than 9. Then, the acceptable actions that can give evidence in favor of 100% 'Unhealthy' state can be refined by the planner using eqs. (7.22) and (7.23).

$$PF_{b,m+1,unhealthy}^{k,l} \geq PF_{b,m+1,unhealthy}^{k,l,\min}, \tag{7.22}$$

$$PF_{b,m+1,unhealthy}^{k,l,\min} = floor\left(\frac{P(U_{test} = unhealthy)}{P(U_{test} = healthy)} PF_{b,m+1,healthy}^{k,l,\min}\right) \tag{7.23}$$
$$if \quad PF_{b,m+1,unhealthy}^{k,l,\min} < 1 \Rightarrow PF_{b,m+1,unhealthy}^{k,l,\min} = 1$$

The reason behind multiplying $PF_{b,m+1,healthy}^{k,l,\min}$ with the ratio of 'Unhealthy' prior, $P(U_{test} = unhealthy)$ to the 'Healthy' prior $P(U_{test} = healthy)$ in eq. (7.23) can be

explained by an example. Let us assume a dataset where there are 1000 ‘Healthy’ and 100 ‘Unhealthy’ training patterns. Therefore, it more likely to find a subset of at least 50 'Healthy' persons, whose data remain within a certain range ($PF_{b,m+1,healthy}^{k,l,min} = 50$) than to find such a subset of 50 'Unhealthy' persons. In such a case, the CDS uses eq. (7.23) and finds a subset with at least 5 ‘Unhealthy’ persons, whose data remain within a certain range. Therefore, the Bayesian statistics inspired eq. (7.23) represents a better approach of applying the desired predefined policy to refine actions by the planner.

Error Calculation and CoO-based decision making

The assurance factor measures the expected assurance about the decision after the current action c_k . Therefore, multiple final rewards for the estimation of FA and DE can be calculated as:

$$rw_{FA}^{k} = \begin{cases} 1 - \dfrac{\sum_{t=1}^{k} AF_{FA}^{t}}{k} & k \le T_{unhealthy} \\ rw_{FA}^{T_{unhealthy}} & k > T_{unhealthy} \end{cases} \tag{7.24}$$

$$rw_{DE}^{k} = \begin{cases} 1 - \dfrac{\sum_{t=1}^{k} AF_{DE}^{t}}{k} & k \le T_{healthy} \\ rw_{DE}^{T_{healthy}} & k > T_{healthy} \end{cases} \tag{7.25}$$

Here, $T_{unhealthy}$ and $T_{healthy}$ are the maximum number of actions available in the action-space to provide evidence of 100% ‘Unhealthy’ and 100% ‘Healthy’ states, respectively. Also, rw_{FA}^{k} and rw_{DE}^{k} is the estimated false alarm and the estimated diagnosis error for the UUT, respectively. The process of CoO uses the estimated rewards in eqs. (7.24) and (7.25) that are presented in Algorithm 7.1.

For the screening process, the initial decision of the CDS about the UUT at the beginning of the PMAC (k=0) is ‘Unhealthy’. For a specific UUT or based on the pre-defined policy, if there exists no such action in the action-space that can provide 100% evidence that the UUT is ‘Healthy’, the Executive does not change its initial decision and moves to the steady-state with a final decision of 'Unhealthy' about the UUT. If the CDS finds 100% evidence of the UUT being ‘Healthy’ but does not find 100% evidence

of the user being 'Unhealthy', the CDS changes the final to decision to 'Healthy' and moves to the steady-state. However, in the case when the CDS finds 100% evidence for the UUT in favor of both the 'Healthy' and 'Unhealthy' states, then the CDS checks whether the eq. (7.26) is satisfied at *k*th PMAC (Details are provided in Algorithm 7.1):

$$(k\times(1-rw_{DE}^{k}))^{w_{DE}} > (k\times(1-rw_{FA}^{k}))^{w_{FA}} \tag{7.26}$$

Here, w_{DE} and w_{FA} are arbitrary pre-defined weights associated with the diagnosis error and the false alarm, respectively, and are determined based on the trade-off between them.

Algorithm 7.1: Conflict of opinions (CoO)

CDS initial decision: UUT is unhealthy

if $T_{healthy} == 0$ **then**
    Keep decision to unhealthy.
    Stop process and turn on steady state.
elseif $T_{unhealthy} == 0$ **then**
    Keep decision as unhealthy.
    Stop process and turn on steady state.
elseif $k \le T_{unhealthy}$ & $k \le T_{healthy}$ **then**
    if $(k\times(1-rw_{DE}^{k}))^{w_{DE}} > (k\times(1-rw_{FA}^{k}))^{w_{FA}}$ **then**
        Change decision to healthy.
    else
    Keep/change decision as unhealthy.
    end of if
elseif $k > T_{unhealthy}$ & $k \le T_{healthy}$ **then**
    if $(k\times(1-rw_{DE}^{k})^{w_{DE}} > T_{unhealthy}\times(1-rw_{FA}^{k}))^{w_{FA}}$ **then**
        Change decision to healthy.
        Stop process and turn on steady state.
    else
        Keep/change decision as unhealthy.
    end of if
elseif $k \le T_{unhealthy}$ & $k > T_{healthy}$ **then**
    if $(T_{healthy}\times(1-rw_{DE}^{k}))^{w_{DE}} > (k\times(1-rw_{FA}^{k}))^{w_{FA}}$ **then**
        Change decision to healthy.
    else
    Keep/change decision as unhealthy.
        Stop process and turn on steady state.
    end of if
else

28: **if** $(T_{healthy} \times (1 - rw_{DE}^{k}))^{w_{DE}} > (T_{unhealthy} \times (1 - rw_{FA}^{k}))^{w_{FA}}$ **then**
29: Change decision to healthy.
30: Stop process and turn on steady state.
31: **else**
32: Keep/change decision as unhealthy.
33: Stop process and turn on steady state.
34: **end of if**
35: **end of if**

Learning

The purpose of the Executive is to find prospective actions that can optimize the cost-to-go function. Then, the system will apply some relevant actions on the NGNLE based on the actions required for finding 100% evidence in favor of 'Healthy' or 'Unhealthy' states. Here, relevant actions can be, for example, asking for related information to the user or activating sensors for new measurements. For each health condition, the Executive activates the sensors to obtain maximum information about the health conditions using the planning and learning sections. Reinforcement learning will be done once the database is updated with new information or the smart e-Health system is upgraded with new sensors. The data obtained from the Executive through reinforcement learning can also be updated with new measurements from the sensors in a later time.

The false alarm and diagnosis error due to the virtual environmental action of c_{k+1}^{s} can be predicted by,

$$rw_{FA}^{k+1,s} = \begin{cases} 1 - \dfrac{\sum_{t=1}^{k} AF_{FA}^{t} + AF_{FA}^{k+1,s}}{k+1} & k+1 \le T_{unhealthy}, \\ rw_{FA}^{T_{unhealthy}} & k+1 > T_{unhealthy} \end{cases} \tag{7.27}$$

$$rw_{DE}^{k+1,s} = \begin{cases} 1 - \dfrac{\sum_{t=1}^{k} AF_{DE}^{t} + AF_{DE}^{k+1,s}}{k+1} & k+1 \le T_{healthy}, \\ rw_{DE}^{T_{healthy}} & k+1 > T_{healthy} \end{cases} \tag{7.28}$$

$$AF_{FA}^{k+1,s} = \frac{PF_{b,m+1,unhealthy}^{k+1,l,s}}{PF_{b,m+1,unhealthy}^{k+1,l,s} + 1}. \tag{7.29}$$

$$AF_{DE}^{k+1,s} = \frac{PF_{b,m+1,healthy}^{k+1,l,s}}{PF_{b,m+1,healthy}^{k+1,l,s} + 1}. \tag{7.30}$$

Here, $s \in \{1,2,...,S\}$ and S is the total number of desired actions that can be applied for the screening process. Also, in eqs. (7.29) and (7.30), $PF_{b,m+1,healthy}^{k+1,l,s}$ or $PF_{b,m+1,unhealthy}^{k+1,l,s}$ are the precision factors received through the internal feedback from the Perceptor corresponding to the 'Healthy' and 'Unhealthy' states, respectively. Then, the cost-to-go for the desired action c_{k+1}^{s} can be calculated using eq. (7.31) as,

$$f_s^{(k+1)}(rw_{DE}^{k+1,s}, rw_{FA}^{k+1,s}) = \begin{cases} \frac{(rw_{DE}^{k+1,s} - rw_{DE}^{k})^{w_{DE}}}{(rw_{FA}^{k+1,s} - rw_{FA}^{k})^{w_{FA}}}, & s \in \{1,2,...,S\} \\ f^{k}(rw_{DE}^{k}, rw_{FA}^{k}) & s = 0 \end{cases}, \tag{7.31}$$

$$(feature\_location)_s^{(k+1)} = [l, b]. \tag{7.32}$$

Therefore, we can find the action $c_{k+1}^{S'}$ that minimizes the cost-to-go function as,

$$\begin{aligned} S' &= \arg\min_{s \in \{0,1,2,..,S\}} (f_s^{k+1}). \\ [L', B'] &= f_{S'}^{k+1}. \end{aligned} \tag{7.33}$$

As a result, the actions to be applied on the environment can be selected as,

$$c_{k+1} = c_{k+1}^{S'}, \qquad for\ S' \geq 1. \tag{7.34}$$

Therefore, c_{k+1} is the best action to be applied on the environment to improve the CoO-based decision-making performance based on the desired policy set by the user. Algorithm 7.2 shows the outline of the main processes of the global PAC of the proposed CDS.

Algorithm 7.2: CDS for user-health prediction (planner, reinforcement learning and policy in Executive, running diagnostic test in the Perceptor)

Input: The observables and features from the database for each user for the focus level *m*, models, ranges, policy (screening process, desired diagnosis error, w_{DE}, w_{FA} ,…), a database of users

Output: Decision about the health state of the UUT

Initialization:

c_0 ←The actions, [Vital signs and portable sensors,⋯] apply on the user
Start advanced actions such as 12 leads ECG or … if c_0 shows an anomaly in the user's health or a request is placed by the user.
Load the model at focus level m=0 and action-space C, Decision = 0 (UUT unhealthy), $Threshold_{DE}$.

$c_{k=0} \leftarrow$ an action randomly selected from C and focus level 0

Apply to $c_{k=0}$ to UUT

Extract features $\tilde{\mathbf{O}}^{k=0,1}$ and choose one of them with maximum $PF_{b,1}^{k=0,l}$, load $r_{k=0}^{m=0,l,b}$

Calculate $rw_{DE}^{k=0}$ and $rw_{FA}^{k=0}$

1: **while** $k \leq (K-1)$ **then**
2: Update features based on

Planning
3: $\boldsymbol{C\_buf} \leftarrow C(m)$

Learning
4: **for** all actions $(s \in \boldsymbol{C\_buf})$ **do**
5: **for** l=1 to L_m **do**
6: **for** b = 1 to B_{m+1}^{l}
7: Calculate $rw_{DE}^{k+1,s}$
8: Calculate $rw_{FA}^{k+1,s}$
9: Calculate $f_s^{(k+1)}(rw_{DE}^{k+1,s}, rw_{FA}^{k+1,s})$
10: **End for**
11: **End for**
12: **End for**
13: Extract $S', L' and\ B'$
14: Remove S' from $\boldsymbol{C\_buf}$
15: **if** S'<1 **then**
16: Calculate $F_{m+1}^{total,k}$
17: **if** $F_{m+1}^{total,k} \leq Complexity\ threshold$ **then**
18: Increase focus level m by 1 (if $U_{\min}$ met)
19: $k \leftarrow k+1$
20: **else**
21: Message: Not meeting $Threshold_{DE}$
22: Decision =Unhealthy
23: **Return** Decision
24: **end if**
25: **else**
26: Apply action (sensor activation) S' on user ($c_{k+1}^{S'}$)
27: $k \leftarrow k+1$

Run screening process
28: $O_{U_{test}} \leftarrow Extracted\ features\ of\ c_{k+1}^{S'}$

29: **if** $R_{\min}^{k,m+1,L',B'} \le \bar{O}_{U_{test}}^{k,m+1}(L') \le R_{\max}^{k,m+1,L',B'}$ and $\hat{\boldsymbol{O}}_{\boldsymbol{DF},\boldsymbol{U}_{test}}^{k,m}$ **then**

30: Calculate rw_{DE}^{k}

31: Calculate rw_{FA}^{k}

32: Run Algorithm 7.1 for CoO Decision making

33: **End while**

7.4. Case study: screening of a user with or without Arrhythmia

Cardiovascular disease (CVD) is among the leading causes of death in the world [7.2]. CVD results in Arrhythmia. Therefore, we used the proposed CDS to distinguish between 'Arrythmia' and 'Normal' ECG as a proof-of-concept application of CDS for health screening in presence of a defective dataset. Here, we chose a dataset of single-lead ECGs posted in PhysioNet Computing in Cardiology Challenge (CINC) 2017 [7.3] to implement the proposed algorithm.

The CINC 2017 dataset can be considered as an example of defective dataset, because all the submitted trained models have problems of overfitting, resulting in the performance of the models to drop significantly in the testing phases [7.4]. It was mentioned in [7.4] that estimation of health state using CINC 2017 database is a non-trivial problem. In addition, the examiners concluded that the number of training patterns in the CINC 2017 dataset are not enough to provide advantage for complicated algorithms over simple algorithms (Defective database) [7.4]. It is also mentioned in [7.4] that more training patterns and better labeling of CINC 2017 dataset are required for superior performance. However, defective CINC 2017 dataset may not be reliable for accurate arrhythmia diagnostic test, but it may still be useful for primary screening process [7.4]. Figure 7.8 shows that the extracted Heart rate (HR) using an available MATLAB program from the Physionet website for CINC 2017 database [7.5] and the HR in clinically valid MATLAB (UCI) dataset. The details about the MATLAB dataset are provided in [1.37]. The MATLAB dataset contains age newborn to elderly people. However, in the CINC 2017 database, some important features such as age or sex are not provided. Figure 7.8 shows that a normal resting heart rate (HR) for adults' ranges should be between 55-101 Beats/min based on MATLAB dataset (see section 6.3 or

[1.37]). So, if we assume that the HR in CINC 2017 are the resting heart rate, then, HR lower than 55 Beats/min is called Bradycardia or slow HR and HR higher than 101 Beat/min is called Tachycardia or fast HR. Both Bradycardia and Tachycardia are two well-known Arrhythmia classes. Figure 7.8 shows that 1371 out of 5076 (~27% of Normal rhythms) are out of normal HR range and this misleads any ML-based AI approach that trusts the dataset and labeling. Therefore, we selected CINC 2017 as the defective dataset to implement a proof-of-concept case study of the proposed PMAC-based CDS.

Table 7.1: Prevalence in CINC 2017 database (Priors)

Class	Users out of 8528 persons	Prevalence %
Normal rhythms	5076	59.5
Atrial fibrillation	758	8.9
Other rhythms	2415	28.3
Noisy recordings	279	3.3

The provided ECG recordings in the CINC 2017 database are collected using a US Food and Drug Administration (FDA) approved single-lead portable ECG device, KardiaMobile from AliveCor [7.4]. The database contains 8,528 single-lead ECG recordings from 9 s to just over 60 s collected at a sampling rate of 300 Hz. Table I shows the number of ECG waveforms in each of the four classes. However, we used this dataset to do screen between two binary classes of healthy (normal rhythms) and unhealthy (AF and other rhythms and noisy recordings) states as a proof-of-concept application of our proposed CDS.

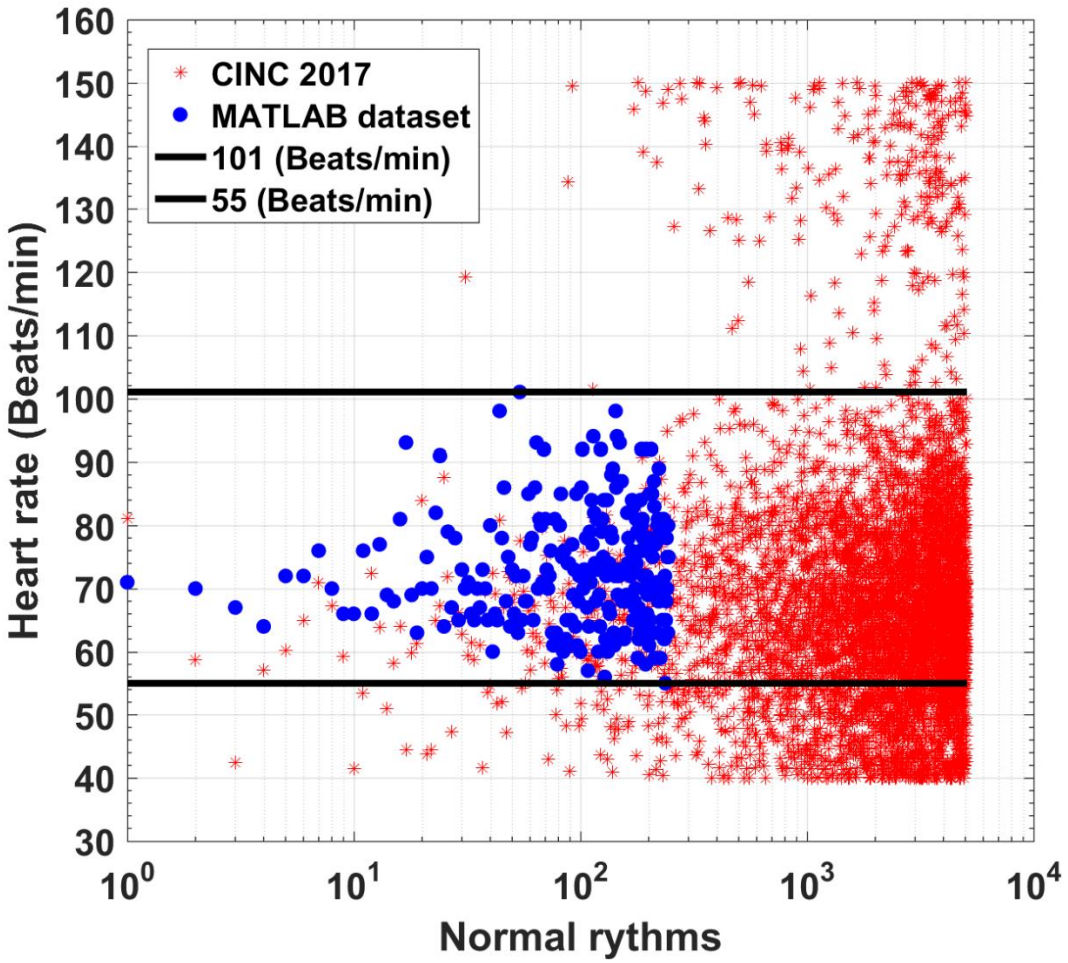


Figure 7.8: Heart rate of persons labeled as normal rhythms in CINC2017 [7.3] vs MATLAB dataset [1.37].

Table 7.2: Simulation parameters of CDS

Simulation parameters	Value
Focus level m	3
Feature in i = 1	1 (feature 80)
Feature in i = 2	187
$DF_{i=1}^{k}$	5
$DF_{i=2}^{k}$	5
Normalization range	|feature value| ≤1
w_{DE}	1

w_{FA}	1

Table 7.3: Summary of simulation results for proposed CDS

Policy Diagnosis error (DE)	DE≤25%	DE≤10%	DE≤5.9%	DE≤2.5%
Final Real DE %	13.2	9.9	6.6	4.6
Final average Internal rewards: DE %	8	6.5	5.8	2.8
Final Real False alarm (FA) %	20.1	25	28.4	54.7
Final average Internal rewards: FA %	22	35	47	42
% of the UUTs resolved by CoO	99.3	98.3	83.3	11.7
$PF_{b,3}^{k,l,\min}(healthy)$	3	9	16	39
$PF_{b,3}^{k,l,\min}(unhealthy)$	2	6	11	26

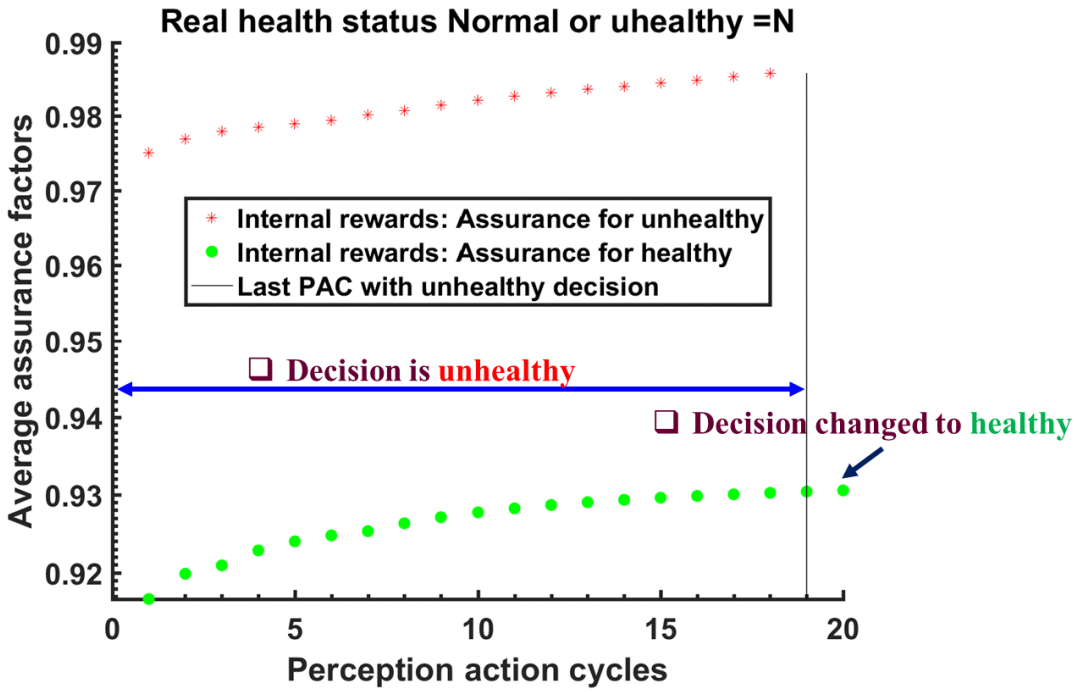


Figure 7.9: Example of decision-making based on conflict of opinions (CoO) ($PF_{b,3}^{k,l,\min}(healthy)=3$, $PF_{b,3}^{k,l,\min}(unhealthy)=2$, N: Normal rhythm/Healthy).

7.4.1. Simulation parameters and CoO decision making example for Case study

Here, we used a validation dataset from [7.3, 7.4] and removed them from the training dataset to verify the performance of the proposed CDS. Detailed information related to the dataset and validation dataset can be found in [7.4]. The simulation parameters used for the CDS-based screening process are presented in Table 7.2. We extracted the features using an available MATLAB program from the Physionet website [7.5]. The program extracts 188 features from each ECG signal [7.5]. For simplicity, the CDS uses only one feature of 80 as the node at focus level 1, while at focus level 2 the rest 187 features are used (see sections 7.3.1, Figures. 7.5 and 7.6). At focus level 3, the CDS extracts important features from the measured ECG signals that carry information corresponding to the healthy or unhealthy state of the UUT. Also, all features are normalized to their absolute maximum value at focus level 0 (see Table 7.2).

Figure 7.9 shows an example of the CoO-inspired decision-making process. The UUT in this example had an ECG with 'Normal rhythm' therefore, was considered to be in a 'Healthy' state. As it can be seen from Figure 7.9, the decision about the UUT's health state is 'Unhealthy' in all PACs from 0 to 19. However, the CDS changes the decision to 'Healthy' in PAC 20 through non-monotonic reasoning i.e. earlier decision was invalidated by adding new evidence. The CDS stays on the decision of 'Healthy' based on the current measurement from the sensor as 1) there remains no more evidence showing that the person is 'Unhealthy' and 2) the weight of the evidence in favor of a 'Healthy' state is higher (See section 7.3.2, eq. (7.26) and Algorithm 7.1). However, if the CDS finds new evidence in favor of the 'Unhealthy' state in future measurements, it may change the decision to 'Healthy'.

7.4.2. Average estimated and real learning curves for Case study for different predefined policies

We simulate the CDS for four predefined goals and policies. The objective of the CDS is to minimize the false alarm at four different levels of desired diagnosis errors, *i.e.* 25%, 10%, 5.9% and 2.5% (see section 7.3.1-7.3.2, $Threshold_{DE}$). The simulation parameters and a summary of the results are presented in Table 7.3. The simulation

results of average internal rewards for false alarm and diagnosis error estimation (see section 7.3.2 and eqs. (7.24) and (7.25)) as well as the real false alarm and diagnosis error are shown in Figures 7.10-7.13 with respect to the PAC number for the desired policy of diagnosis error ($Threshold_{DE}$) less than 25%, 10%, 5.9% and 2.5%, respectively. It can be seen from Figures 7.10-7.13 that the CDS-generated average internal rewards for diagnosis error and false alarm reach to a good agreement at higher PMAC with the actual average diagnosis error and average false alarm (see Table 7.3, also).

It can be seen from Figures 7.10 and 7.11 and Table 7.3 that the CDS can fulfill the requested policies. However, in Figures 7.12 and 7.13 (See Table 7.3 also), the final diagnosis errors are a little higher than the desired values. This is mainly because when the policy changes from the required DE of 5.9% to 2.5%, the number of UUTs who were successfully resolved by CoO decreased from 83.3% and 11.7% (see Table 7.3). This is the proposed CDS cannot meet the required the diagnosis errors when the number of users for CoO decrease significantly (See Table 7.3).

In Figure 7.10, both the estimated and real diagnosis error are high at the beginning. This is due to the fact that the actions used at the beginning cause some actual 'Unhealthy' UUTs to be falsely diagnosed as 'Healthy'. However, as the number of PMACs increases and the CDS finds new evidence at every PMACs, it changes the decision about them (See Figure 7.9 also). For a similar reason, the average estimated diagnosis error in Figure 7.11 is initially high, but showed a good agreement with the real average diagnosis error at higher PAC numbers. In addition, the number of PMACs decreased from Figure 7.10 to Figure 7.13. This is due to the fact that 1) the number of users with both actions decreases (Figures 7.12-7.13); and 2) based on the policies applied (Figure 7.11-7.13), less actions are now available in the action-space (see section 7.3.3 and eqs. (7.21)-(7.23)).

Like the diagnosis error, estimation of the false alarm by the CDS also depends on the number of available users with both available actions. The CDS performs the screening process by minimizing the false alarm for a predefined desired diagnosis error. It can be seen in Figure 7.10 that there is a good agreement between the final estimated false alarm and real false alarm. Also, the final real false alarms are lower than the CDS-estimated false alarms (Figures 7.10-7.12). However, in Figure 7.13, the

estimated false alarm is lower than the actual false alarm, which results from the fact that the estimation was done based on ~11.7% of the users (see Table 7.3). Also, unlike the estimated false alarm rates in Figures 7.10 and 7.11, it can be seen in Figure 7.12 that the estimated false alarm is high for the earlier PMACs, while real false alarm decreased. This is mainly because when the policy changes to the required DE of 5.9%, the number of UUTs who were successfully resolved by CoO also decreased to 83.3%. The remaining 16.7% UUTs contribute to the inaccurate estimation of real false alarm in the earlier PMACs.

In general, based on Table 7.3, the average final diagnosis error decreased from 13.2% in Figure 7.10 to 9.9% in Figure 7.11, 6.6% in Figure 7.12 and 4.6% in Figure 7.13. However, the trade-off of decreasing average final diagnosis error is the increment of the average final false alarm from 20.1% in Figure 7.10 to 25% in Figure 7.11, 28.4% in Figure 7.12 and 57% in Figure 7.13. It can be seen from Figure 7.12 and Figure 7.13 that the trade-off of 2% improvement of average final diagnosis error is a 28.6% increase in the average final false alarm. This can be attributed to the fact that the proposed CDS can apply CoO only on 12% of users in this policy in Figure 7.13 (see Table 7.3 also). As a consequence, for improving diagnosis error more than 5.9% the significant penalty for false alarm increment should be paid as the penalty.

In summary, we can conclude that the CDS should have more than 80% (ideally, 98%) UUTs with the CoO (For example Figure 7.10-7.12, see Table 7.3 also). The percentage of decisions made by CoO out of the total decisions can be used as another internal metric and reward for the proposed CDS. This metric would give the proposed CDS a 'self-awareness' about the reliability of its performance.

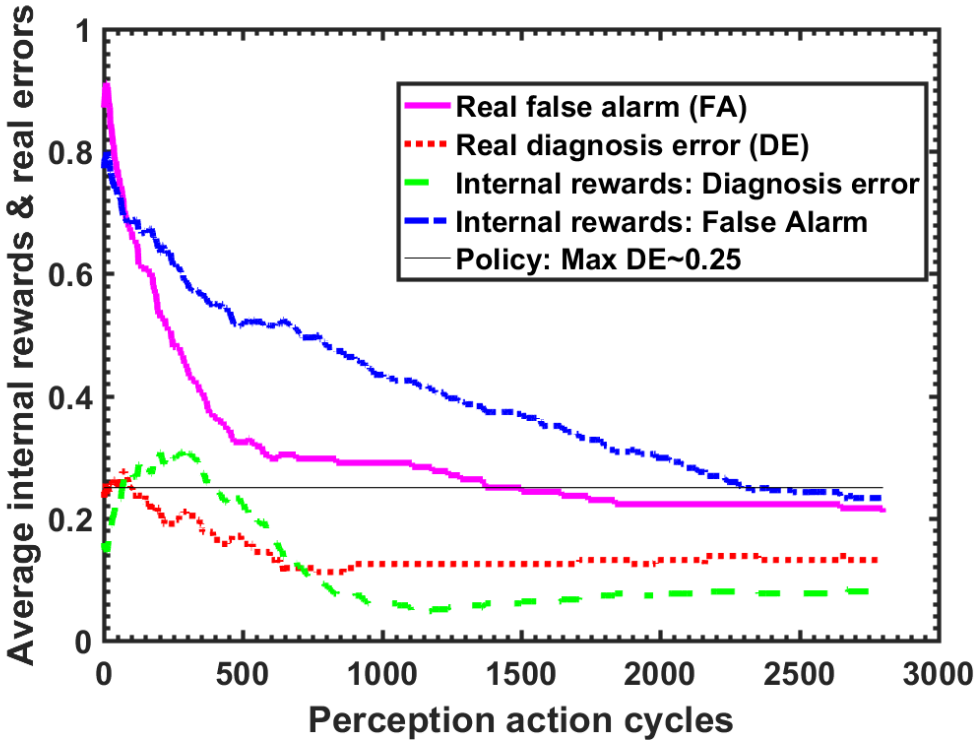


Figure 7.10: Average internal rewards vs. average real diagnosis error and false alarm, for desired diagnosis error (DE)≤25%.

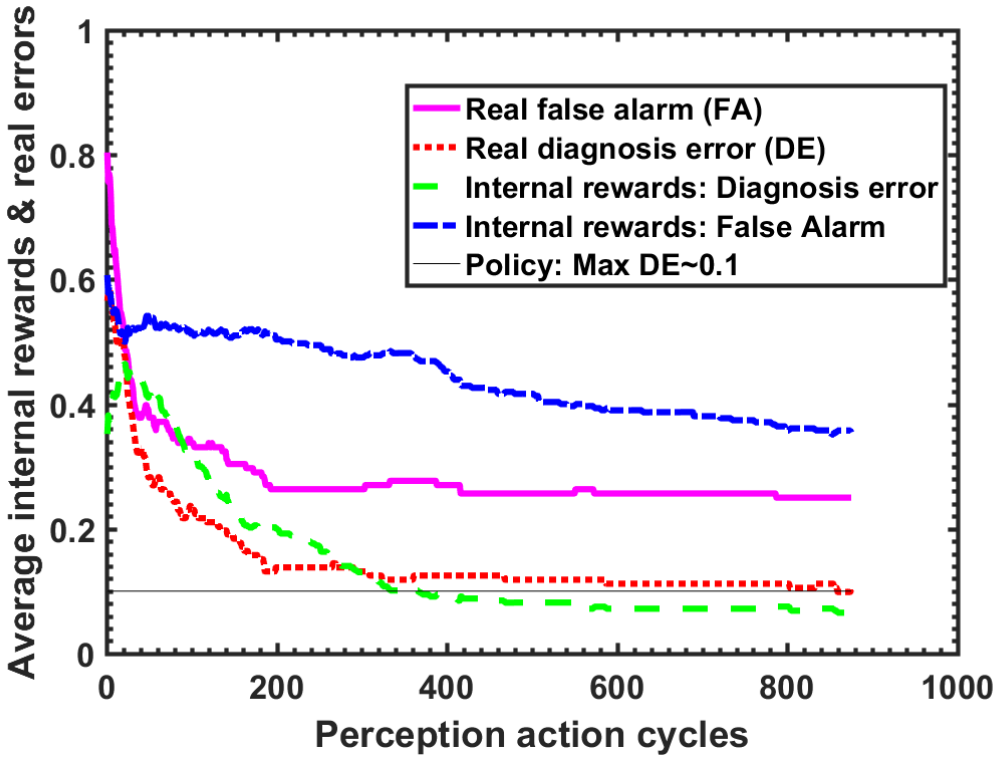


Figure 7.11: Average internal rewards vs. average real diagnosis error and false alarm, for DE≤10%.

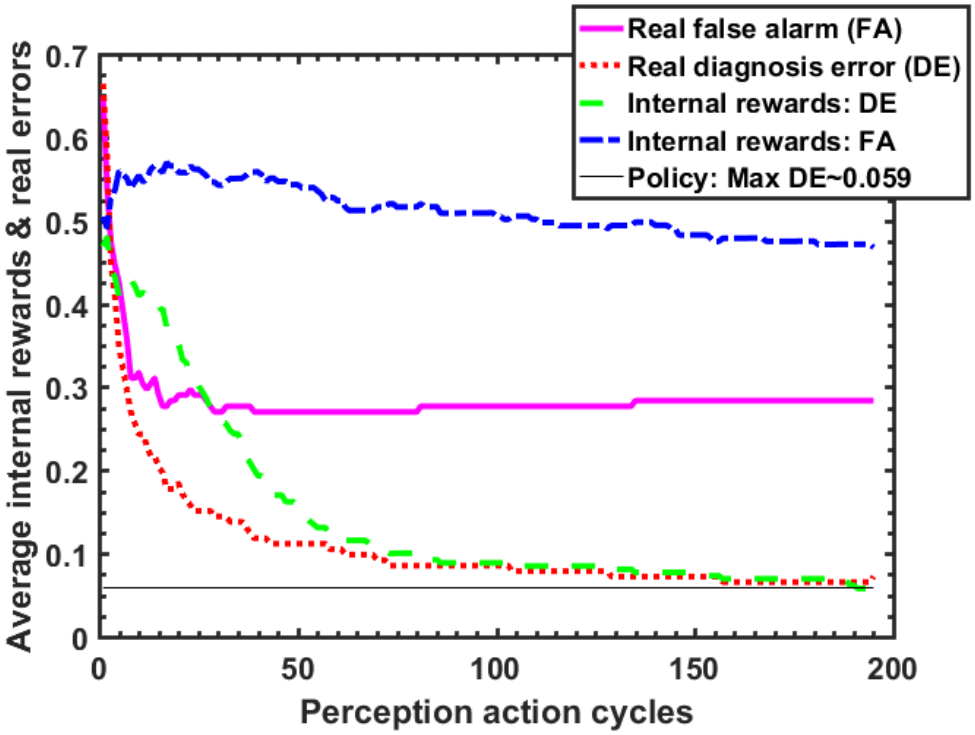


Figure 7.12: Average internal rewards vs. average real diagnosis error and false alarm, for DE≤5.9%.

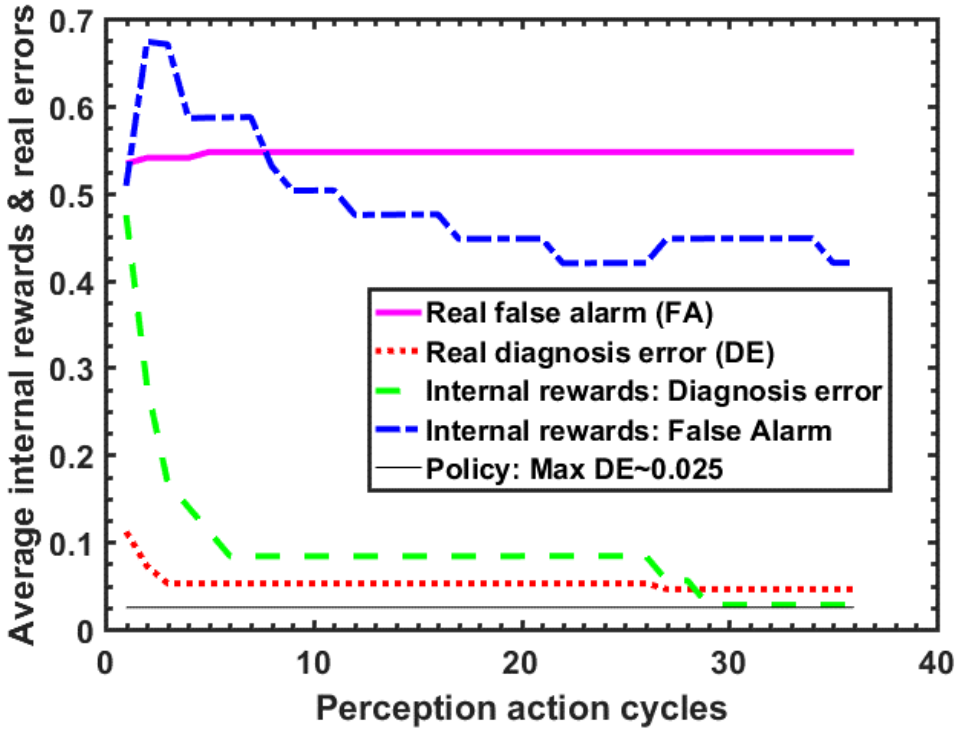


Figure 7.13: Average internal rewards vs. average real diagnosis error and false alarm, for DE≤2.5%.

7.5. Conclusions

In recent years, there has been a growing interest in developing smart interactive cyber-physical systems (CPS) such as smart home, and e-Health. An autonomic decision-making system (ADMS) is of paramount importance for the autonomic computing layer of such systems. The ADMS for a smart e-Health Home may include functionalities such as real-time dynamic training or decision-making, screening process, treatment, healing tracking as well as recommendations for healthy living. In this paper, we proposed a PMAC-based cognitive dynamic system (CDS) for an ADMS to enable automatic screening of human health with an acceptable level of false alarm rates. We also proposed a CoO-inspired decision-making algorithm that allows the proposed CDS to make a decision at a pre-defined level of confidence even when the training dataset itself is poorly labeled or unbalanced.

The system architecture and algorithms are developed to realize a health screening (i.e., healthy or unhealthy) application with an acceptable level of the false alarm policy. To illustrate the application of the proposed system, a proof-of-concept case study is performed on a defective dataset of ECG traces. The performance of the proposed CDS shows good agreement with the desired performance metrices. For the desired diagnosis errors equal or less than 25%, 10%, 5.9%, and 2.5%, the CDS achieved diagnosis errors of 13.2%, 9.9%, 6.6%, and 4.6%, respectively. These diagnosis errors are achieved with acceptable false alarm rates [7.1] of 20.1%, 25%, 28.4%, and 54.7%, respectively. Therefore, we could simulate the flexibility and reliability of the proposed CDS for screening purposes even with a training dataset that is defective or tempered through a cyber-attack that may disrupt the labeling or remove some training patterns from the dataset.

In summary, a CDS for health screening application is proposed and implemented. This CDS incorporates decision-making trees, non-monotonic reasoning, a decision-making approach inspired from humans in the case of conflict of opinions, prediction using the extracted model, and the characteristics of non-Gaussian and non-linear health features. The CDS checks only one feature in each perception-multiple-actions

cycle, making the proposed algorithm simple and fast. Finally, this work is the second step for designing the ADMS for screening applications in a smart e-Health system that can be extended for different healthcare policies such as diagnosing the disease class, prevention and early detection.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1. Conclusions

In recent years, some efforts were made to develop intelligent machines using cognitive dynamic system (CDS) for artificial intelligence (AI) applications. However, currently, typical CDS available algorithms are only applied to linear and Gaussian environment (LGE). These algorithms cannot be applied for a non-Gaussian and nonlinear environment (NGNLE) with finite memory. Therefore, we proposed novel CDS algorithms that could be applied on a NGNLE with finite memory. We redesigned the CDS in detail to make it suitable for NGNLE cognitive decision-making (CDM) applications. Also, the system architecture and algorithms are presented. Finally, the presented algorithms in this thesis are simple and fast because it used the Bayesian equation for calculating the posterior that has low complexity for the posterior extraction. Therefore, the low complexity algorithms makes the proposed CDS a perfect candidate for NGNLE applications such as long-haul fiber optic links or in healthcare.

In chapter 2, the basic problem for situation understanding in a NGNLE with finite memory is defined. Then, the background, research challenges, motivation and implementation of CDS for long haul fiber optic link as first example of NGNLE. Then, healthcare as the second example of a NGNLE is provided with the background and motivation of implementation of CDS for healthcare applications is presented. Also, similarities and difference of long-haul fiber optic link and healthcare applications are described.

In chapter 3, CDS v1 (simple CDS) is applied to a non-linear fiber optic communication system for the bit error rate (BER) improvement and the data rate enhancement. By cycling through the prediction mode and BER improvement mode

with appropriate actions by the executive, we have found that the data rate and Q-factor can be enhanced by 12.5% and 2.74 dB, respectively, as compared to the conventional fiber optic system. The key advantage of CDS is that it is intelligent, and software-defined, and it can automatically recognize and extract information about the fiber optic channel. In addition, it can enhance the data rate and improve the BER for different fiber optic link characteristics such as the span length, input power, and other system/signal parameters. These characteristics may change during the data transmission in a fiber optic network, but the CDS can still adapt to these changes. The computational cost of the proposed technique is much lower than the methods used to mitigate nonlinear impairments, such as digital back propagation (DBP).

In chapter 4, CDS v3 algorithms for NGNLE are applied to the long-haul non-linear fiber-optic link as a case study. By upgrading the conventional link to smart one, the achievable data-rate is increased from 208 Gb/s to 236 Gb/s with the help of CDS. Also, the CDS improves the Q-factor by 3.5 dB (the final BER $= 4.6\times10^{-3}$ at 236 Gb/s) as compared to the conventional system without CDS, which means that the required HD overhead can be reduced. As a result, the final data rate enhancement is 23.3%.

In chapter 5, a general algorithm of CDS v4 for a NGNLE with finite memory is applied on a long-haul non-linear fiber-optic link as a case study. By upgrading the conventional link with the CDS, the applicable data-rate increased from 208 Gb/s to 280 Gb/s. Also, the CDS Q-factor improved by ~7 dB (the optimum BER $= 4.4\times10^{-3}$ for 280 Gb/s with CDS). The data rate enhancement is ~35% without considering the HD overhead reduction. However, the CDS brings the BER under the new FEC threshold ($= 4.7\times10^{-3}$), and we do not need 8% of the HD-FEC overhead. Hence, the net data rate enhancement is ~43%.

In chapter 6, the system architecture and algorithms of CDS v2 are presented for health situations (i.e., healthy or unhealthy) diagnosis with low false alarm policy. To illustrate the application of our proposed system, a proof-of-concept case study is done on an Arrhythmia dataset. Our system provided an acceptable total accuracy of 95.4% that was achieved by increasing the focus level of the CDS. Also, for the proposed Arrhythmia case study, the run time of our algorithms is less than 80 ms for

a User of the smart e-Health home. This time included the time for the real-time modeling in two focus levels and for decision making.

The smart e-Health home can be implemented using IoT technology as a more generalized form of a cyber-physical system. In this scenario, the smart e-Health home using IoT technology must consider issues related to fog and edge computing and latency. Furthermore, the proposed, low complexity fast algorithms for the autonomic computer layer do not intensify these IoT technology issues because in the example provided, less than 80 ms was required time for training and prediction.

For the Arrhythmia case study as a proof-of-concept example, we used the database in which some key features such as heart rate were missing. It should be noted that missing key features can also be regarded as sensor(s) failure. Therefore, we could simulate the proposed CDS flexibility and reliability in the presence of the sensor(s) failure. Furthermore, the 95.4% accuracy shows that the proposed CDS can find alternative actions for the relevant diagnostic tests.

In chapter 7, CDS v5 is proposed as a PMAC-based CDS for an ADMS to enable automatic screening of human health with an acceptable level of false alarm rates. We also proposed a CoO-inspired decision-making algorithm that allows the proposed CDS to make a decision at a pre-defined level of confidence even when the training dataset itself is poorly labeled or unbalanced. The system architecture and algorithms are developed to realize a health screening (i.e., healthy or unhealthy) application with an acceptable level of false alarm policy. To illustrate the application of the proposed system, a proof-of-concept case study is performed on a defective database of ECG traces. The performance of the proposed CDS shows good agreement with the desired performance matrices. For the desired diagnosis errors equal or less than 25%, 10%, 5.9%, and 2.5%, the CDS achieved diagnosis errors of 13.2%, 9.9%, 6.6%, and 4.6%, respectively. These diagnosis errors are achieved with acceptable false alarm rates of 20.1%, 25%, 28.4%, and 54.7% [7.1], respectively. Therefore, we could simulate the flexibility and reliability of the proposed CDS for screening purpose even with a training dataset, which is defective or tempered through a cyber-attack that may disrupt the labeling or remove some training patterns from the dataset.

8.2. Future work

8.2.1. CDS

Five versions of a CDS are proposed in this thesis. However, CDS v4 is a more general purpose CDS that is proposed based on MAP rule decision making. For future studies, the proposed CDS for a NGNLE can be improved as a research tool. Here, suggestions for future works are provided:

1. The perceptor can be upgraded using SL algorithms to directly extract the posteriori.
2. The assurance factor concept is used for internal rewards by the CDS. However, further investigation and study are required to make sure this is the optimum choice for NGNLE in terms of accuracy and complexity.
3. Implementing of language for CDS can be done by designing a network of connected CDSs to exchange information, NGNLE models, and experiences using machine-to-machine (M2M) communications protocols.
4. Internal commands and adaptive modeling configuration, a simple executive is used that performs basic internal commands without considering permutation and probable complicated weighted relationships between environmental actions, type of extracted features and internal commands. So, these can be addressed in future works using a more advanced executive.

8.2.2. Fiber optic communications

Here, the suggestions are provided for future work on fiber-optic communication systems that are upgraded with CDS:

1- CDS can be applied to practical and experimental fiber optic communications systems using wave division multiplexing (WDM) or optical time division multiplexing (OTDM). Then, evaluation of CDS performance can help for future implementation of CDS in commercial fiber optic communications and improving CDS algorithms.

2- The network of CDSs mentioned in 8.2.1 can be applied to WDM fiber optic network and a cloud-based action and model libraries for a network of CDSs. The

network of CDSs may be a unique solution to address the inter-channel WDM nonlinear distortions in fiber optic networks.

3- In this thesis, the main objective of upgrading fiber-optic link with CDS is "controlling the quality of services and providing reliability of communication over the fiber optic link as well as guarantee maximum achievable data rate (or transmission distance)." Therefore, a detailed study is required to investigate which subsystem(s) of fiber optic communications system can be replaced by a CDS.

8.2.3. Smart e-Health systems

Design of a complex executive for more general healthcare applications and policies could be carried out in future. In this thesis, we implemented the first step of the ADMS for "*Diagnostic test* (*low false alarm policy) and screening process (acceptable false alarm policy) for someone who has a disease or no disease."* However, in future, the algorithms could be extended for other ADMS policies, step by step to include: *"Diagnosing disease class, i.e., someone has known diseases, but ADMS should diagnose the disease class"*, *"Search for and treat the cause of disease"*, *"Healthy life recommendations for illness prevention"*, and *"Tracking, recovery and healing after surgery or medical treatment"*.

For extending ADMS to other healthcare policies using the CDS, proper datasets are necessary for training the CDS and preventing blind trial and error by the executive reinforcement learning (RL). In addition, for *prevention policy* and *tracking recovery and healing after surgery or medical treatment*, the fundamental definition of prevention or tracking algorithms should be converted to engineering descriptions for implementation.

In addition, in prevention, the CDS should be able to predict the User's future health condition as the current CDS is designed to find health conditions at the present time. With predictive capabilities, the CDS can recommend to the User the required actions for the present time and help the User to maintain their good health condition. In addition, for the disease class diagnosis policy, the important question of: "*How many disease classes can be diagnosed by the CDS in the presence of unforeseen noise, disturbance and false information?"* should be answered.

Therefore, for reliable diagnoses and reduced risks, the CDS should be able to calculate the maximum number of possible disease classes that can be diagnosed at the present time. In addition, another open question to be studied is "*How the CDS can use a reduced dataset* (*e.g., some important sensors failed or are not functioning properly*) *with the ADMS and its policies to still make an accurate diagnosis?"* A preliminary form of this question is addressed in this thesis, but more study is needed. Finally, when these questions are answered, it will be possible to apply the CDS to healthy life recommendations, prevention and to search for and treat the cause of diseases.

Appendix I: VIRTUAL LONG-HAUL FIBER OPTIC LINK FOR CDS V3 AND V4

In this appendix, the implementation of eq. (5.16) for a fiber optic communication system is presented. In CDS v3 and CDS v4, the posteriori in eq. (4.10) and eq. (5.17) due to the prospective environmental action of c_{k+1}^{t} can be estimated as follows:

$$\hat{Y}_n^{(k+1),t,m} = \hat{Y}_n^{k+t,m} = \hat{Y}_n^{k,m} \times \left(ratio^{k+1,t} \right)^{\frac{(m+1)(t\times d)}{2(R_k - R_{k-1})}} = \hat{Y}_n^{k,m} \times \left(\max_{s=1,2,\ldots,S} \left(\frac{std(\hat{Y}_n^{k,m} \mid X_n^k = X_s)}{std(\hat{Y}_n^{k-1,m} \mid X_n^{k-1} = X_s)} \right) \right)^{\frac{(m+1)(t\times d)}{2(R_k - R_{k-1})}} \quad \text{(A.I.1)}$$

Here S is the constellation size, for example in QAM-16, S is equal to 16. Also, d is 4 Gb/s discretization step for data rate and R_k and R_{k-1} are data rate at PAC number kth and (k-1)th, respectively (see table 5.2 for other parameters and notations and section 5.2.3). The eq. (A.I.1) is developed based on the fact that it is expected that by increasing the data-rate, nonlinear distortions of received signal in fiber optic communications increases too.

Appendix II: COPYRIGHT PERMISSIONS

Dear Mahdi,

Thanks for your email.

Kind regards,

Ms. Fancy Chai
Managing Editor
Sensors Editorial Office
E-Mail: fancy.chai@mdpi.com

MDPI
St. Alban-Anlage 66, 4052 Basel, Switzerland
Tel. +41 61 683 77 34; Fax +41 61 302 89 18
http://www.mdpi.com/

Women in Sensors Award (1000 CHF)
https://www.mdpi.com/journal/sensors/special_issues/Women_Sensors

20th Anniversary of Sensors:
A series of special content and events at
https://www.mdpi.com/journal/sensors/anniversary

Kind regards,

Ms. Fancy Chai
Managing Editor
Sensors Editorial Office
E-Mail: fancy.chai@mdpi.com

Dear Mahdi Naghshvarianjahromi,

Your article was published in IEEE Access under a Creative Commons 4.0 license. You will need to see http://creativecommons.org/licenses/by/4.0/ for what you need to do.

All that is necessary for IEEE is that you cite the original work.

Kind regards,

M.E. Brennan

Ms M.E. Brennan
IEEE
501 Hoes Lane
Piscataway, NJ 08854-4141 USA
me.brennan@ieee.org

+1 (732) 562-2660

Dear Mahdi,

Thanks. As open access journal, you don't need to apply for copyright.
You can cite them in your PhD thesis.

Best Wishes,

Daria
--
Ms. Daria Shi, M.Sc.
MDPI Branch Office, Beijing
Suite 305, Zhongjia Mansion, Building No.13, Taiyangyuan Community,
Dazhongsi East Road, Haidian District, Beijing
E-mail: daria.shi@mdpi.com
Skype: daria.shi.mdpi

MDPI
St. Alban-Anlage 66, 4052 Basel, Switzerland
http://www.mdpi.com/

Capacity Limits of Optical Fiber Networks

Author:

René-Jean Essiambre

Publication:

Lightwave Technology, IEEE/OSA Journal of

Publisher:

IEEE

Date:

Feb.15, 2010

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted chapter in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these chapters) users must give full credit to the original source (author, chapter, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original chapter is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted chapter in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, chapter title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted chapter can be used when posting the chapter or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

REFERENCES

1.1- S. Majumder, T. Mondal, M. J. Deen, "Wearable sensors for remote health monitoring," *Sensors*, vol. 17, no. 1, Jan. 2017.

1.2- H. Wang, et al., "A utility maximization approach for information communication tradeoff in wireless body area networks," *Pers Ubiquit Comput*, vol. 18, no. 8, pp.1963–1976, 2014.

1.3- H. Wang, et al., "Information-based energy efficient sensor selection in wireless body area networks," *IEEE international conference on communications—symposium on selected areas in communications e-Health Track (ICC2011—SAC EH)*, Kyoto, Japan, pp. 1-6, 5–9 June 2011.

1.4- M. J. Deen, "Information and communications technologies for elderly ubiquitous healthcare in a smart home," *Pers. Ubiquitous Comput.*, Vol. 19, pp. 573–599, 2015.

1.5-An architectural blueprint for autonomic computing (2015) IBM autonomic computing white chapter, 3rd Edition, June 2005. Available at https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Chapter%20V7.pdf [Accessed 20-06-2020].

1.6- J. O. Kephart, D. M. Chess, "The vision of autonomic computing. Computer", *Computer*, vol. 36, no. 1, pp. 41-50, Jan. 2003.

1.7- J. M. Fuster, *Cortex and Mind: Unifying Cognition*; Oxford University: Oxford, UK, 2003.

1.8- S. Haykin, *Cognitive Dynamic Systems: Perception-Action Cycle, Radar, and Radio*; Cambridge University: Cambridge, UK, 2012.

1.9- M. Fatemi and S. Haykin, "Cognitive Control: Theory and Application," *IEEE Access*, vol. 2, pp. 698-710, 2014.

1.10- M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain Inspired Dynamic System for the Quality of Service Control Over the LongHaul Nonlinear Fiber-Optic Link," *Sensors*, vol. 19, no. 9, pp. 2175-2195, 2019.

1.11- M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain-Inspired Cognitive Decision Making for Nonlinear and Non-Gaussian Environments," *IEEE Access*, *7*, 180910–180922, 2019.

1.12- S. Feng and S. Haykin, "Cognitive Risk Control for Transmit-Waveform Selection in Vehicular Radar Systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9542-9556, Oct. 2018.

1.13- M. I. Oozeer and S. Haykin, "Cognitive Risk Control for Mitigating Cyber-Attack in Smart Grid," *IEEE Access*, vol. 7, pp. 125806-125826, 2019.

1.14- S. Haykin. *Neural Networks and Learning Machines*; 3rd edition. Prentice-Hall: 2009.

1.15- C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, 27.3, pp. 379–42, 1948.

1.16- M. T. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons: 2012.

1.17- A. Raghu, et al., "Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach," *Proceedings of Machine Learning for Healthcare (MLHC) 2017*, pp. 1-17, 2017.

1.18- S. Haykin, "Cognitive Dynamic Systems: Radar, Control, and Radio," *Proceedings of the IEEE*, Vol. 100, No. 7, pp. 2095–2103, 2012.

1.19- F. Khozeimeh and S. Haykin, "Brain-Inspired Dynamic Spectrum Management for Cognitive Radio Ad Hoc Networks," in *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3509-3517, October 2012
1.20- S. Haykin, "The Cognitive Dynamic System for Risk Control," *Proceedings of the IEEE*, Vol. 105, No. 8, pp. 1470-1473, 2017.
1.21- M. I. Oozeer, and S. Haykin, "Cognitive Dynamic System for Control and Cyber-Attack Detection in Smart Grid," in *IEEE Access*, vol. 7, pp. 78320-78335, 2019.
1.22- S. Feng, and S. Haykin, "Cognitive Risk Control for Anti-Jamming V2V Communications in Autonomous Vehicle Networks," in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9920-9934, Oct. 2019.
1.23- Mehdi Fatemi, Peyman Setoodeh, and Simon Haykin, "Observability of stochastic complex networks under the supervision of cognitive dynamic systems", *Journal of Complex Networks*, vol. 5, no. 3, pp. 433–460, 2017.
1.24- S. Feng, P. Setoodeh and S. Haykin, "Smart Home: Cognitive Interactive People-Centric Internet of Things," in *IEEE Communications Magazine*, vol. 55, no. 2, pp. 34-39, February 2017.
1.25- S. Haykin, "Artificial Intelligence Communicates With Cognitive Dynamic System for Cybersecurity," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 463-475, Sept. 2019.
1.26- R. Bono, M. J. Blanca, J. Arnau, and J. Gómez-Benito, "Non-normal Distributions Commonly Used in Health, Education, and Social Sciences: A Systematic Review," *Frontiers in Psychology*, pp. 1-4, 2017.
1.27- Y. Si and R. b. Liu, "Diagnostic Performance of Monoexponential DWI Versus Diffusion Kurtosis Imaging in Prostate Cancer: A Systematic Review and Meta-Analysis", American Journal of Roentgenology, vol. 211, No. 2, pp. 358-368, 2018.
1.28- M. H. Maurer, and J. T. Heverhagen, "Diffusion weighted imaging of the prostate-principles, application, and advances". *Transl Androl Urol*., vol 6, No. 3, pp. 490–498, 2017.
1.29- T. Micceri, "The unicorn, the normal curve, and other improbable creatures". *Psychol. Bull.*, vol. 105, No. 1, pp. 156–166, 1989.
1.30- L. Diaz-Serrano, "Labor income uncertainty, skewness and homeownership: a panel data study for Germany and Spain," *J. Urban Econ.*, vol. 58, No. 1, pp. 156–176, 2005.
1.31- S. N. Shahi, S. Kumar, and X. Liang, "Analytical modeling of cross-phase modulation in coherent fiber-optic system," *Opt. Express*, vol. 22, no. 2, pp. 1426–1439, 2014.
1.32- V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical principles of fuzzy logic*, Dordrecht: Kluwer Academic, 1999.
1.33- L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8, no. 3, pp. 338-353, 1965.
1.34- C. Cortes, and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273–297, 1995.
1.35- M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Natural Brain-Inspired Intelligence for Non-Gaussian and Nonlinear Environments with Finite Memory," *Appl. Sci.* 2020, *10*, 1150, 2020.
1.36- D. Silver, et al., "Mastering the game of go with deep neural networks and tree search," Nature, 529 (7587), pp. 484-489, 2016.
1.37- M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain-Inspired Intelligence for Real-Time Health Situation Understanding in Smart e-Health Home Applications," *IEEE Access*, vol. 7, pp. 180106–180126, 2019.
1.38- M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Brain Inspired Dynamic System for the Quality of Service Control Over the LongHaul Nonlinear Fiber-Optic Link," *16th Canadian Workshop on Information Theory (CWIT) 2019*, Hamilton, June. 2019.

1.39- M. Naghshvarianjahromi, S. Kumar, M. J. Deen, "Smart long-haul fiber optic communication systems using brain like intelligence," *16th Canadian Workshop on Information Theory (CWIT) 2019*, Hamilton, June. 2019.
1.40- W. Peng, H. Takahashi, I. Morita and T. Tsuritani, "Per-symbol-based DBP approach for PDM-CO-OFDM transmission systems," Optics Express, vol. 21, no. 2, pp. 1547-1554, 2013.
1.41- G. Vasu, et al., "Prediction and Classification of Cardiac Arrhythmia," *Stanford University*, Stanford, CA, USA, 2014.
1.42- A. Hilmy, et al., "Deep learning algorithm for Arrhythmia detection," *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, pp. 26-32, 2017.
1.43- M. Karandikar, G. Guidi, "Classification of Arrhythmia Using ECG Data," *CS229*, USA, Fall 2014.

2.1- S. Majumder, T. Mondal and M. J. Deen, "Smartphone sensors for health monitoring and diagnosis," *Sensors*, vol. 19, no. 9, pp. 2164, 2019.
2.2- S.L. Jansen, I. Morita, N. Takeda and H. Tanaka, "20 Gb/s OFDM transmission over 4,160-km SSMF enabled by RF-pilot tone phase noise compensation," *PDP15, NFOEC,* Anaheim, CA, March 2007.
2.3- G. P. Agrawal, Fiber-Optic Communication Systems. John Wiley & Sons, 1997
2.4- S. Kumar, and M. J. Deen, *Fiber optic communications: fundamentals and applications*; John Wiley & Sons: 2014.
2.5- J. P. Gordon and L. F. Mollenauer, "Phase noise in photonic communications systems using linear amplifiers," *Opt. Lett.*, vol. 15, no. 23, pp. 1351–1353, 1990.
2.6- R. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini and B. Goebel, "Capacity Limits of Optical Fiber Networks," in *Journal of Lightwave Technology*, vol. 28, no. 4, pp. 662-701, Feb.15, 2010.
2.7- M. Fujishima, "Key Technologies for THz Wireless Link by Silicon CMOS Integrated Circuits," *Photonics, 5*, 50, 2018. Yariv, A.; Fekete, D.; Pepper, D.M. Compensation for channel dispersion by nonlinear optical phase conjugation. *Opt. Lett.* **1979**, *4*, 52–54.
2.8- S. Watanabe, G. Ishikawa, T. Naito and T. Chikama, "Generation of optical phase-conjugate waves and compensation for pulse shape distortion in a single-mode fiber," in *Journal of Lightwave Technology*, vol. 12, no. 12, pp. 2139-2146, Dec. 1994
2.9- S. Kumar and D. Yang, "Optical backpropagation for fiber-optic communications using highly nonlinear fibers," *Opt. Lett.*, 36, pp. 1038–1040, 2011.
2.10- X. Liang, S. Kumar, J. D. Downie, W. A. Wood, and J. E. Hurley, "Transmission Performance of OFDM Signals Over 6,000 km Fiber Optic Links with Digital Back Propagation," in *Advanced Photonics 2018 (BGPP, IPR, NP, NOMA, Sensors, Networks, SPPCom, SOF)*, OSA Technical Digest (online) (Optical Society of America, 2018), chapter SpM4G.2, 2018.
2.11- X. Liang, S. Kumar, and J. Shao, "Ideal optical backpropagation of scalar NLSE using dispersion-decreasing fibers for WDM transmission," *Opt. Express*, 21, pp. 28668-28675, 2013.
2.12- S. Kumar and J. Shao, "Optical Back Propagation With Optimal Step Size for Fiber Optic Transmission Systems," *IEEE Photonics Technology Letters*, vol. 25, no. 5, pp. 523-526, March1, 2013.
2.13- B. Foo, B. Corcoran and A. Lowery, "Compensating XPM Using a Low-Bandwidth Phase Modulator," *IEEE Photonics Technology Letters*, vol. 29, no. 9, pp. 699-702, 1 May1, 2017.

2.14- E. Ip and J. M. Kahn, "Compensation of Dispersion and Nonlinear Impairments Using Digital Backpropagation," *Journal of Lightwave Technology*, vol. 26, no. 20, pp. 3416-3425, Oct.15, 2008
2.15- E. Mateo, L. Zhu, and G. Li, "Impact of XPM and FWM on the digital implementation of impairment compensation for WDM transmission using backward propagation," *Opt. Express*, 16, pp. 16124-16137, 2008.
2.16- X., Liang, X.; Kumar, S. Multistage perturbation theory for compensating intra-channel impairments in fiber optic systems. *Opt. Express* **2014**, *22*, 29733–29745.
2.17- X., Liang and S., Kumar, "Multi-stage perturbation theory for compensating intra-channel nonlinear impairments in fiber-optic links," *Opt. Express*, 22, pp. 29733-29745, 2015.
2.18- A. Hasegawa and T. Nyu, "Eigenvalue communication," *Journal of Lightwave Technology*, vol. 11, no. 3, pp. 395-399, March 1993.
2.19- M. I. Yousefi and F. R. Kschischang, "Information Transmission Using the Nonlinear Fourier Transform, Part I: Mathematical Tools," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4312-4328, July 2014
2.20- E. G. Turitsyna, and S. k., Turitsyn, "Digital signal processing based on inverse scattering transform," *Opt. Lett*. 38, pp. 4186–4188, 2013.
2.21- K. V. Peddanarappagari and M. Brandt-Pearce, "Volterra series transfer function of single-mode fibers," in *Journal of Lightwave Technology*, vol. 15, no. 12, pp. 2232-2241, Dec. 1997.
2.22- M., Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*; John Wiley & Sons: Hoboken, NJ, USA, 1980.
2.23- M., Nazarathy, et al, "V. Phased-array cancellation of nonlinear FWM in coherent OFDM dispersive multi-span links," *Opt. Express*, 16. pp. 15777–15810. 2008.
2.24- C. Lin *et al*., "Adaptive digital back-propagation for optical communication systems," *OFC 2014*, pp. 1-3, San Francisco, CA, 2014.
2.25- F. Zhang *et al*., "Blind Adaptive Digital Backpropagation for Fiber Nonlinearity Compensation," *Journal of Lightwave Technology*, vol. 36, no. 9, pp. 1746-1756, 1 May1, 2018.
2.26- A. Du, B. Schmidt and A. Lowery, "Efficient digital backpropagation for PDM-CO-OFDM optical transmission systems*," OFC/NFOEC*, OTuE2, 2010.
2.27- New StatsCan data reveals hundreds of 'excess' deaths in Canada amid pandemic, https://www.cbc.ca/news/canada/excess-deaths-covid-1.5619723. [Accessed 21-06-2020].
2.28- Man dies after CT scan report showing cancer goes unread for more than a year, https://www.healthimaging.com/topics/oncology-imaging/man-dies-ct-scan-cancer-unread-year. [Accessed 21-06-2020].
2.29- The third-leading cause of death in US most doctors don't want you to know about, https://www.cnbc.com/2018/02/22/medical-errors-third-leading-cause-of-death-in-america.html. [Accessed 21-06-2020].
2.30- J. Henry, *et al*., "Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015," *ONC data brief*, no. 35, pp. 1–9, 2016.
2.31- M. Ghassemi, et al., "Unfolding physiological state: Mortality modeling in intensive care units," *KDD*, pp. 75–84, 2014.
2.32- V. Gulshan, et al., "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.
2.33- A. Esteva, et al., "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, 542(7639), 115, 2017.

2.34- Y.-A Chung, W.-H Weng, “Learning deep representations of medical images using siamese cnns with application to content-based image retrieval,” *Machine Learning for Health (ML4H) workshop at NIPS 2017*, 2017.
2.35- K. Nagpal, et al., “Development and validation of a deep learning algorithm for improving gleason scoring of prostate cancer,” *npj digital medicine*, vol. 2, no. 1, pp. 1-11, 2019.
2.36- A. Raghu, et al., “Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach,” *Proceedings of Machine Learning for Healthcare (MLHC) 2017*, pp. 1-17, 2017.
2.37- W.-H Weng, et al., “Representation and reinforcement learning for personalized glycemic control in septic patients,” *Machine Learning for Health (ML4H) Workshop at NIPS 2017*, 2017a.
2.38- M. Komorowski, et al., “The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care,” *Nature Medicine*, vol 24, no. 11, pp. 1716-1720, 2018.
2.39- E. P. Lehman, et al., “Representation learning approaches to detect false Arrhythmia alarms from ECG dynamics,” *Proceedings of Machine Learning for Healthcare (MLHC) 2018*, pp. 1-15, 2018.
2.40- T.-M. Hsu, et al., “Unsupervised multimodal representation learning across medical images and reports,” *Machine Learning for Health (ML4H) Workshop at NeurIPS 2018*, pp. 1-15. 2018.
2.41- G. Liu, et al., “Clinically accurate chest x-ray report generation,” *Proceedings of Machine Learning for Healthcare (MLHC) 2019*, pp. 1-20, 2019.
2.42- S. Gehrmann, et al., “Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives,” *PLoS One*, vol. 13, no. 2, pp. 1-18, 2018.

3.1- M. Ahissar, et al. “Reverse hierarchies and sensory learning,” *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* vol. 364,1515, 285-99, 2009.
3.2- A. S. Skidin, et al. “Mitigation of nonlinear transmission effects for OFDM 16-QAM optical signal using adaptive modulation,” *Opt. Expres*s, 24, pp. 30296–30308, 2016.
3.3- L. M. Zhang and F. R. Kschischang, "Staircase Codes With 6% to 33% Overhead," *Journal of Lightwave Technology*, vol. 32, no. 10, pp. 1999-2002, May15, 2014.
3.4- X. Liang and S. Kumar, “Correlated digital backpropagation based on perturbation theory,” *Optic Express*, vol. 23, no. 11, pp. 14655-14665, 2015.

6.1- S. Voss, "Fuzzy Logic in Health Care Settings: Moral Math for Value-Laden Choices," *Journal of Humanistic Mathematics*, vol. 6, no. 2, pp. 161-178, 2016.
6.2- J. H. T. Bates, M. P. Young, “Applying fuzzy logic to medical decision making in the intensive care unit,” *American Journal of Respiratory and Critical Care Medicine,* vol 167, no. 7, 2003.
6.3- D. DeMazumder, D.E. Lake, A. Cheng, T.J. Moss, E. Guallar, R.G. Weiss, S.R. Jones, G.F. Tomaselli, J.R. Moorman, “Dynamic analysis of cardiac rhythms for discriminating atrial fibrillation from lethal ventricular Arrhythmias,” *Circ Arrhythm Electrophysiol*, vol. 6, no. 3, pp. 555–561, 2013.
6.4- A. Petrėnas, L. Sörnmo, A. Lukoševicius, V. Marozas, “Detection of occult paroxysmal atrial fibrillation”, *Med Biol Eng Comput*, vol. 53, no. 4, pp. 287–297, 2015.

6.5- M. Macgill, "What should my heart rate be?" https://www.medicalnewstoday.com/articles/235710.php. [Accessed 22-06-2020].
6.6- Pediatric tachycardia algorithm (with a pulse and poor perfusion), https://www.uptodate.com/contents/image?imageKey=EM%2F67438&topicKey=EM%2F6392&source=see_link. [Accessed 22-06-2020].
6.7- T. Martin, The Effects of Carbon Monoxide From Cigarette Smoking, https://www.verywellmind.com/carbon-monoxide-in-cigarette-smoke-2824730, Updated 31 August, 2018. [Accessed 22-06-2020].
6.8- P. Harris, "Dimitrios Lysitsas; Ventricular Arrhythmias and sudden cardiac death," *BJA Education*, vol. 16, no. 7, pp. 221–229, 2016.
6.9- R. Mehra, "Global public health problem of sudden cardiac death," *Journal of Electrocardiology*, vol. 40, no. 6, (6 Suppl): S118–22.
6.10- National Heart, Lung, and Blood Institute, "Who Is at Risk for an Arrhythmia?", July 1, 2011. [Accessed 22-06-2020].
6.11- Arrhythmia Treatment Services, Brigham Women's hospital, https://www.brighamandwomens.org/heart-and-vascular-center/resources/Arrhythmia-treatment. [Accessed 22-06-2020].
6.12- S. Padhi S, et al., "Prevalence of cardiac Arrhythmias in a community-based chiropractic practice," *The Journal of the Canadian Chiropractic Association*, vol. 58, no. 3, pp. 238-245, 2014.
6.13- P. M. Steen M., et al., "Clinical value of transesophageal atrial stimulation and recording in patients with Arrhythmia-related symptoms or documented supraventricular tachycardia-correlation to clinical history and invasive studies," *Clinical Cardiology*, vol. 17, no. 10, pp. 528–534, 1994.
6.14- T. Hendrikx T, et al., "Intermittent short ECG recording is more effective than 24-hour Holter ECG in detection of Arrhythmias," *BMC Cardiovascular Disorders*. pp.14-41, 2014.
6.15- Q. Zhang, X. Chen, Z. Fang and S. Xia, "False Arrhythmia alarm reduction in the intensive care unit using data fusion and machine learning," *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Las Vegas, NV, 2016, pp. 232-235.
6.16- Matlab sample datasets, https://www.mathworks.com/help/stats/sample-data-sets.html. [Accessed 22-06-2020].
6.17- UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Arrhythmia. [Accessed 22-06-2020].
6.18- ECG interpretation Part 1: definitions criteria and characteristics of the normal ECG (EKG) waves, intervals, durations & rhythm, https://ecgwaves.com/ecg-normal-p-wave-qrs-complex-st-segment-t-wave-j-point/. [Accessed 22-06-2020].
6.19- C. Luan, G. Dong, "Experimental identification of hard data sets for classification and feature selection methods with insights on method selection," Data & Knowledge Engineering, Volume 118, 2018, Pages 41-51.AHA Releases 2015 Heart and Stroke Statistics, http://www.sca-aware.org/sca-news/aha-releases-2015-heart-and-stroke-statistics. [Accessed 22-06-2020].
6.20- A. Kumar, "Machine Learning: Validation Techniques," https://dzone.com/articles/machine-learning-validation-techniques. [Accessed 22-06-2020].

7.1- Drew, B. J. et al., "Insights into the problem of alarm fatigue with physiologic monitor devices: a comprehensive observational study of consecutive intensive care unit patients," *PLoS ONE*, 9, e110274 2014.

7.2- The top 10 causes of death: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. [Accessed 22-06-2020].
7.3- The PhysioNet Computing in Cardiology Challenge 2017, https://physionet.org/content/challenge-2017/1.0.0/ [Accessed 22-06-2020].
7.4- G. D. Clifford *et al*., "AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017," *2017 Computing in Cardiology (CinC)*, Rennes, 2017, pp. 1-4, doi: 10.22489/CinC.2017.065-469.
7.5- S. Datta *et al.* Matlab codes for feature extraction, https://physionet.org/content/challenge-2017/1.0.0/sources/shreyasi-datta-209.zip [Accessed 22-06-2020].