PARTICLE SWARM OPTIMIZATION APPLICATIONS TO MPEG-4 TRANSMISSION OVER ZIGBEE

By

Iman Samizadeh

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

London Metropolitan University

School of Computing

October 2012

PAGE

NUMBERING

AS ORIGINAL

Declaration

I, Iman Samizadeh confirm that the work presented in this thesis is solely my work, and that to the best of my knowledge the work is original. Where the information has been driven from other sources, I confirm that this has been indicated in the thesis and the source is always given by reference to respective authors.

Abstract

The IEEE 802.15.4 standard commonly known as ZigBee is a wireless sensor targeted at applications that require low data rate, low power and less expense. IEEE 802.15.4 standard is limited to a through-rate of 250kbps maximum providing supports for small packet file transitions and is design to provide highly efficient connectivity. Hence, IEEE 802.15.4 is not designed and cannot be used to transfer large amounts of data. Therefore, in this research MPEG-4 video transmission over ZigBee is the aim as its bandwidth is too low and the limitation could become a real problem which makes the video transmission over IEEE 802.15.4 networks difficult to achieve. Due to the low bandwidth of the ZigBee any large amount of data needs to be optimized at the targeted bitrate. Optimization techniques are widely used in engineering and computer science as well as being used in real environment applications to overcome complex issues and in particular; an artificial intelligence technique known as Particle Swarm Optimization (PSO) has becoming very popular. PSO is a population-based stochastic optimization technique, inspired by the social behavior of flocks of birds, and colonies of ants and bees. Such intelligence is decentralized, self-organized and distributed throughout an environment and used by swarm to solve problems. In this research, the problem solving strategy decided on the use of PSO to optimize the transmission of MPEG-4 video over ZigBee, which requires a much lower computation and accordingly it can be executed faster.

A novel solution to transmit MPEG-4 over IEEE 802.15.4 has been developed and this research further utilizes a technique to regulate the quantization patterns and output an optimal frame rate by using an "Adaptive Scalar Quantization", which prevents excessive data loss of MPEG-4 video over IEEE 802.15.4 transmission. The computer simulation results confirm that adaptive scalar quantization video coding do improve the quality of picture and reduce data loss and prevents excessive data loss, and use of particle swarm optimization can improve QoS and empower video within the MPEG-4 compression technique to be transmitted over IEEE 802.15.4 standard compared to conventional MPEG video transmissions.

School of Computing

London Metropolitan University

Lead Supervisor: Professor Hassan Kazemian

Second Supervisor: Doctor Ken Fisher

Third Supervisor: Professor Karim Ouazzane

Acknowledgments

First, I would like to express my sincerest gratitude to my supervisor, Professor Hassan Kazemian, for his support, great insights and the encouragement he has given me throughout the course of this research. I am grateful to Doctor Ken Fisher for his kind support, extensive suggestions and valuable discussions, which greatly contributed and helped throughout my PhD. And last but not least, many thanks go to Professor Karim Ouazzane for thorough comments and the continuous support, encouragement and inspirational character throughout the past few years.

Finally, thanks to my mother who cared for me and made immeasurable sacrifices to ensure that I have all the support that I needed. She was patiently stood by me even when I could not put up with myself. Sincere gratitude and respect to my father, who despite wanting to protect me from making mistakes allowed me to find my own way, and has faith in me even when I fail. He has always guided me towards the paths that I would succeed in; and gave me the greatest gift anyone could give another person. I would like to deeply thank my brother Ali and my sweet sisters Sorousheh and Souphiyeh, for all their love and care they showered upon me all my life.

I dedicate this PhD thesis to my mother and father; I am who I am because of them.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

In recent years, wireless technology has undergone an amazing transformation shift, enabling multimedia distribution and communications between people and devices. As well as a rapid increase in demand for the development of advanced interactive multimedia applications such as video telephony, video games and TV broadcasting from any location. Therefore, the consumer market for remote wireless technologies that use less power consumption and are less expensive is growing rapidly. The IEEE 802.15.4 known as ZigBee is designed to fulfill these desires, being cost-effective, using a Low-Rate Wireless Personal Area Networks (LR-WPANs), as a short distance wireless communication and therefore using low power consumption. This makes IEEE 802.15.4 a very attractive proposition and one that warrants the introduction of a focused standard for everything from household application to a very efficient and low latency application.

IEEE 802.15.4 is a new frequency standard in wireless technology and it is designed to provide highly efficient connectivity. In many ways, it is similar to Wi-Fi and Bluetooth, operating in the same 2.4GHz Industrial Scientific Medical (ISM) bands worldwide at a maximum data-rate of 250kbps, 868 MHz band at a data rate of 20kbps in Europe and 914 MHz band at 40kbps in the USA and Australia. The IEEE 802.15.4 standard is aimed primarily at remote control and sensor applications and it developed extremely fast in smart homes and smart office networks with flexibility and seamless mobility. IEEE 802.15.4 wireless network provides only limited, time-varying Quality of Service (QoS) for the delay-sensitive, bandwidth-intense and loss-tolerant applications. This means it only supports small packet file transitions and is not meant to be used or provide QoS.

While other IEEE wireless standards focus on connectivity between large packet user devices, such as workstations, laptops, smart phones and personal area networking, these products have not yet been able to make a significant impact on the market. At the same time, some wireless standards like 802.11.x and Bluetooth are used more widely especially in the areas of computer and mobile peripherals. Nevertheless, current commonly used WPAN peripherals are not expandable in automation. This has led to the invention of the wireless low data rate personal area networking technology IEEE 802.15.4, to receive tremendous attention from industry leaders and researchers.

However, ZigBee devices are limited to a maximum through-rate of 250kbps, whereas the specified maximum range of operation for ZigBee devices is approximately 70m. For that reason, IEEE 802.15.4 is not designed to transfer large amounts of data or MPEG-4 video as its bandwidth is too low. On a positive note, due to IEEE 802.15.4's low power output, ZigBee devices can sustain themselves on a small battery for many months, or even years, and their self-organizing capability makes them ideal devices to introduce multimedia applications to transfer or live streaming. IEEE 802.15.4/ZigBee's potential as a cost effective easy-to-use product makes it highly likely that it will soon be used to transfer large amounts of data.

The objective of this research is to transmit video over the IEEE 802.15.4 wireless network. Video transmission over IEEE 802.15.4 networks is therefore difficult to achieve and its limitation could become a real problem, especially if the user wishes to transmit a large amount of data in a very short time. For example, the bandwidth required for a 320×200 colour video at 25 frames per second is $320 \times 200 \times 24 \times 25 = 38.4$ Mbps. Whilst with the IEEE 802.15.4 standard bandwidth limitation approximately one uncompressed greyscale (8-bit) frame 256×100 pixels per second, excluding the protocol overhead, and reliable communication with no interference at speed above 115200 bps can be transmitted. This means even the compressed stream that requires quite a lot of CPU power is far too much for the ZigBee data rate limits.

MPEG-4 video files need a large bandwidth in order to stream over wireless networks. However, this is not possible in the case of IEEE 802.15.4. Therefore, it is essential to study an efficient video compression technology where video data transmission is an important issue in many applications. While MPEG-4 already has a very good compression rate, it is not optimized enough to work in IEEE 802.15.4, and it is therefore impossible to fully ensure MPEG-4 video streaming over ZigBee, which will result in seamless content presentation in all circumstances.

MPEG-4 video compression techniques defined by Moving Picture Experts Group is commonly used to store digital video and digital audio streams and it can improve the speed in transmitting multimedia files. MPEG-4 media can be transported on existing transport layers such as Asynchronous Transfer Mode (ATM), Real-time Transport

Protocol (RTP), Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).The two most common protocols in video transmission are briefly described beneath:

- TCP is used if data needs to be exactly received, bit for bit, ensuring there will be no loss of bits. However, it will slow down the transmission, due to the need to re-transmit corrupted packets. In contrast, UDP does not care about reliability and is often used for various types of real-time traffic that does not need strict ordering. RTP is insensitive to packet loss, so it does not require the reliability of TCP, it uses the best effort transmission to transfer all available data in time, but does not attempt to re-transmit data that was lost or corrupted during the transfer. Thus, MPEG-4 transport is packet-based and the packets are transmitted bit-error free. The packet loss rate depends on the network conditions [1].

- Asynchronous Transfer Mode (ATM) is a concept defined by The American National Standards Institute (ANSI) and International Telecommunication Union (ITU) to carry a complete range of services including voice, video and data. It is designed to unify telecommunication and computer networks with high-speed [2]. It is a cell based switching and multiplexing technology and it can support any traffic type, including both delay-sensitive traffic, and non-delay-sensitive traffic. ATM has functional similarity with both circuit switched networking and small packet switched networking.

- ATM networks transmit their information in small, fixed length packets called “cells" [2]. The ATM cell networks is one of the most prominent research topics, specially within the mobile and wireless network, as cell loss can be considered a sub-form of packet loss, where cells are extremely small packets [1],[3]. In order to stream MPEG-4 video over ATM networks, guaranteed quality of service is needed. Quality of service is guaranteed through a few characteristics and service requirements, such as traffic control of call admission and congestion.

Before the streaming of MPEG-4 video is it impossible to know the optimum rate of transmission. Therefore, in order to ensure the best quality of service the quantization parameters must be varied during the encoding process. This will result in an increase of packets and possibly superfluous larger packets during the transmission. Moreover, the application environment, bandwidth or transport layers often does not allow large packets transmission or retransmission of damaged or lost video data because of the real-time constraints of broadcast transmission characteristics. Hence, Artificial Intelligence (AI) introduced to application of MPEG-4 to regulate the data that is transmitted, which can solve the problem and avoid congestion, whilst softening the data transmission when a problem occurs during communication.

In 1956, the field of artificial intelligence research was founded as an academic discipline and has been consistently studied since, and is still one of the most popular subjects in Computer Science. John McCarthy first coined the term artificial intelligence in 1956 when he held the first academic conference on the subject. But the journey to understand if machines can truly think, began long before that [4], [1]. Five years later Alan Turing wrote a paper on the notion of machines being able to simulate human beings and their ability to do intelligent things, such as play chess [4].

AI is widespread in advanced search algorithms, machine-learning algorithms, and many more applications and has become a significant player in problem solving and computation. In addition, computer hardware is becoming smaller and faster. The majority of the population in the world now uses personal portable communication devices. This is because, in recent years smart portable devices have become essential in our everyday life style and connecting more people to the online world is the goal of many service providers. Furthermore, mobile devices are rapidly becoming the dominant digital format as opposed to desk machines. The soulotions to the problem on streaming video over wireless networks has been addressed by many different pertinent pieces of research that have been conducted in AI and wireless communication.

Cheng and Chang, came up with a method based on fuzzy logic in order to control congestion whilst maintaining the quality of service [5]. They improved their model even further by introducing call admission control as well as congestion control on

ATM networks. They implemented the fuzzy traffic controller using a two-threshold congestion method, which uses both the classic mathematical formulation for the control and mimics traffic control, which produced the same capability of admission control on ATM networks. In 1997, in the research carried out by Ascia, et al., an application of fuzzy logic in traffic control in real-time ATM was developed. They designed an AI logic based system to achieve real-time traffic control in high speed networks using fuzzy logic [6].

In the seminal work by Ming-Chang Huang, at al., usage parameter control is presented to ensure that each source conforms to its negotiated parameters in ATM networks [7], [8]. To meet the requirements for the policing function, a fuzzy logic-based system is proposed to deal with the congestion control and policing problem in ATM networks [7]. In order to improve the performance of ATM networks, Ming-Chang and colleagues use a virtual leaky bucket with fuzzy logic control to manage the depletion rate in the bucket. Their simulation results show that the fuzzy leaky bucket system is effective in detecting source violation with low response time and the performance is significantly better than other mechanisms [8].

Shih-Lin proposed a fuzzy adaptive rate control, which would select the transmission rate for frame transmissions in Wireless Local Area Networks (WLANs). This fuzzy adaptive rate control considers the received signal strength indicator, the frame error rate and the medium access control delay to make a correct decision. Simulation results demonstrate that the proposed scheme enhances the network throughput and the access delay [9].

Kazemian and Meng published a solution in May 2006, adding a fuzzy control system introduced at the host controller interface [10]. This fuzzy control scheme was being developed to transmit MPEG-4 over a Bluetooth wireless network to improve QoS in video streaming using a fuzzy approach [11]. The format structure of approaching the video file was described in several methods. [11]. Furthermore, they have carried out research on MPEG and Bluetooth together. Their system uses a buffer to prevent excessive back-to-back cells. A fuzzy rule controller manages the output bit rate of this buffer. Another set of rules manages the input bit rate to optimize the loading of the buffer. The results showed a marked reduction of the data loss. The same authors developed another model, based this time on a neural approach in the design. This project included a buffer to prevent overflow. Neural fuzzy rules managed the input and output bit rate of the buffer to ensure that the multimedia stream from the host conforms to the traffic conditions of the Bluetooth channel during the communication time. Results on this simulation significantly reduced excessive delays during transmission [11]. In addition they studied the transmission of real-time MPEG-4 Variable Bit Rate (VBR) video sequences over an ATM network, using a self-organizing fuzzy controller [10]. The computer simulation results demonstrated that the use of a self-organizing fuzzy controller reduces excessive delay and data loss at the user-network interface compared with a conventional policing mechanism in ATM.

In this research, a new model is presented that is founded upon the Constant Bit Rate (CBR) model framework. VBR model ensures a constant quality of image with fix Quantization Parameter (QP) of the encoder. The CBR encoding scheme gives a wide control to the transfer of videos over wired networks and allows the transmission of video signals over narrow-band network such as ATM networks.

There are arguments for and against the use of both CBR and VBR. CBR service guarantees traffic at a constant rate and is commonly used in typical voice, video and audio, which require more bandwidth than other types of data files. The VBR service is for the applications that require buffering. VBR is typically used to support compressed voice and video. The limitation of IEEE 802.15.4 bandwidth is that the quality of service in CBR and VBR is a real problem if the user wishes to transmit a large amount of data in a very short time with the availability of only 250kbps.

The VBR model is designed to improve the consistency of video quality. It uses the given parameters in the beginning of encoding to make sure that the encoder, with an average bit rate, maintains the minimum, maximum and stay in between, whilst varying the quality of the video objects. High bitrates in VBR usually lead to larger file sizes than with CBR, yet pre-planning the needed bandwidth requirements is more difficult because the bit rate changes and more complex scenes will require greater bandwidth. In addition, VBR requires the use of either storage space or buffering. When storage space is not defined in VBR it requires a huge buffer size, therefore, it is very difficult to pre-set the limit of needed buffers. In order to stream multimedia at the limit of ZigBee's target bitrates a model that forces the application to stay within the bandwidth limit and buffer size was needed. Hence, CBR was chosen for this research project to be the framework of the model. It should be emphasized that ZigBee has a very limited bandwidth and what is considered acceptable is a consistent level of MPEG-4 video streaming and not perfect quality of video objects. Therefore, MPEG-4 needs to be optimized at the targeted bitrate.

Why use of particle swarm optimization? Optimization techniques are widely used in engineering and computer science as well as being used in real environment applications to overcome complex issues and in particular; an artificial intelligence technique known as Particle Swarm Optimization (PSO) has becoming very popular. Better optimization algorithms are always needed. In AI optimization techniques, one of the most used algorithms is Genetic Algorithm (GA) and in recent years, PSO has becoming very popular. In this research, the problem solving strategy decided on the use of PSO. Several studies in regards to measuring the performance of optimization algorithms in between GA and PSO have been carried out, which measure the quality and efficacy of the solutions.

Hassan, et al., research results shown that the computational efficiency superiority of PSO over the GA algorithm [12]. In seven out of the eight test problems investigated they have statistically proven a high confidence level that PSO is more efficient than GA. Further analysis shows that the difference in computational effort between PSO and the GA is a problem dependent. It appears that PSO outperforms the GA with a larger differential in computational efficiency when used to solve unconstrained nonlinear problems with continuous design variables and less efficiency differential when applied to constrained nonlinear problems with continuous or discrete design variables [12]. Furthermore, Yang, Zhang and Sun have published a research paper in which they carry out a comparison of PSO and GS for hidden markov model training, and they conclude that PSO is superior to GA [13] Azarkish, et al., have carried out research in comparing the performance of the PSO and the GA on the geometry design of longitudinal Fin, in their research PSO and the GA are used to minimize the error functions in the inverse design of convective-radiative fin profile. Their results show that particle swarm optimization was at least three times more efficient than genetic algorithm. Therefore, particle swarm optimization is recommended for geometry optimization, especially when the gradient base methods have failed [14]. Sivanandam and Deepa have compared PSO and GA for Lower Order System Modeling and their overall simulation results indicate that both GAs and PSO can be used in the search of parameters during system modeling. With respect to minimizing the objective function Integral Square Error, PSO determines a smaller value than GA. In terms of computational time, the PSO approach is faster than GA, although it noted that neither algorithm takes what is considered as an unacceptably long time to determine the results [15].

A few researchers have carried out studies on video transmitting over ZigBee. For example, in the research carried out by Burda, in MPEG-4 video over ZigBee networks, he demonstrates that it is possible to use an IEEE 802.15.4 network to broadcast voice messages (e.g. for instructions in case of emergencies), as well as, continuous sound streams (e.g. to create ubiquitous ambient sound environments). Furthermore, he has integrated a video image transfer to enable an optical feedback channel for auto-configuration of the system [16]. Another research carried out by Zainaldin, et al., in video transmission over wireless ZigBee networks, and has introduced the use of a single channel for data transmission even though multiple non-overlapped channels exist in the 2.4 GHz spectrum. The aggregate throughput of these networks can be improved by using multiple channels that are available in the radio spectrum allocated by the standard. The focus of the paper is on the performance improvement of ZigBee's networks under the interference of other 802.15.4 and 802.11 standards [17]. Arguably, Zainaldin and Burda's research introduces a dependency to coexist with other wireless standards or devices; this often cannot be possible and limits the type of the applications that can benefit from this approach.

Shilpa, et al., carried out a research on a performance evaluation of MPEG-4 video transmission under various scenarios. These scenarios included the effect of background traffics (CBRs) on videos and the behavior of multiple videos when transmitted simultaneously. Their simulation was carried out on both single and two hop communication networks and their results show that even though IEEE 802.15.4 is a low data rate communication standard, it provides an acceptable human video quality required for applications like video surveillance [18]. Shilpa's research proves that the CBR model is a good candidate for this research project and by applying PSO into this model it can achieve significant improvements.

Following this, Kazemian conducted a different research [19]. He applied AI to the video streaming technique in ZigBee wireless; and according to the research, a new Neural-Fuzzy (NF) scheme was developed to adjust the traffic-shaping buffer output rate, which eliminates unacceptable delay or loss of the VBR encoded video and conforms the data to the token-bucket's contract prior to entering the ZigBee channel [19]. The proposed idea in this research uses PSO, which requires a lot less computation, therefore, it can be executed faster as it is not using a hybrid system and consumes less power, which is what ZigBee is aims for. Additionally, using the PSO model to adaptively decide on quantization scale size and rate control, which was developed using the CBR model as a foundation instead of VBR, has proven that it can provide an acceptable video quality required for the applications based on ZigBee sensor network, according to the simulations results. Therefore, this research should be considered as potentially helpful and a possible direction for future research.

Particle swarm optimization has also been applied in MPEG bit rate optimization. In the research conducted by Arachchi and Fernando PSO-based bit rate optimization for MPEG-1/2 video coding has been studied and they have concluded that one of the significant problems in video compression schemes is the high fluctuation in the output data rate over the video sequence [20]. These compression schemes, in general, utilize a rate control algorithm in order to maintain the output data rate at a constant level, regardless of the properties of the video sequence and the differences in compression ratios of different picture types. Experimental results show that the proposed method can improve the average picture signal to noise ratio (PSNR) by more than 2dB [21], [22].

According to the above research and the similarity between ZigBee and other IEEE 802 standards, the algorithms and the control scheme developed specially for video transmission over wireless networks, which regulates the output bit rate from the MPEG-4 encoder according to the current condition of the wireless channel the problems with transmission should be fixed. As well as the successful studies into the application of fuzzy logic to many traffic control problems in wireless networks and the successful applications of Neural networks and fuzzy logic to MPEG video transmission over wired or wireless networks. However, IEEE 802.15.4 wireless networks standard reveals a number of issues that potentially could become performance bottlenecks and thus lead to serious performance degradation.

After reviewing all these papers on AI, a novel solution developed to transmit MPEG-4 over IEEE 802.15.4. The proposed idea in this research uses PSO, which requires a lot less computation and a new model is presented that is founded upon the CBR model framework to adaptively decide on the QP values, regulate the traffic and allows the transmission of video signals over ZigBee's narrow-band network. In addition this research demonstrates that the purposed technique can be used in transmitting JPEG or CCTV images over ZigBee, Bluetooth or other similar technologies.

1.1 Summary of Chapters

Chapter 1 describes the context of the research for this thesis and highlights the problems in this area of research. Chapter 2 is an overview of the next-generation of wireless communication systems. IEEE 802.15.4 ZigBee and MPEG video compression are described in this chapter. The IEEE 802.15.4 standard has been compared to other existing wireless standards and an extensive literature review has been carried out. Chapter 3 is an introduction to and an overview of particle swarm optimization as well as a background review of PSO's test functions. Chapter 4 explains the methodology of the research and the design of the proposed algorithm for both rate control and a mechanism to adaptive quantization. Furthermore, a novel adaptive quantization technique suggested to control and set an optimal or near-optimal value towards the quantization parameter in an ad-hoc way whilst encoding is in process. Chapter 5 discusses the experiment results, compares the proposed algorithm with other existing methods of streaming in MPEG-4 and presents the results in detail and finally Chapter 6 contains the conclusions drawn from the developed solution in this research and gives possible future directions for research.

Chapter 2: ZigBee and MPEG

2.1 IEEE 802.15.4 ZigBee

This chapter will focus on (the overview of) the ZigBee standard and the MPEG-4 video compression technique. ZigBee's advantages will be compared to other similar IEEE standards in order to justify the purpose of this research and to discuss the possibility of transferring MPEG-4, over IEEE 802.15.4.

IEEE 802.15.4 is a relatively new technology, therefore, very few simulations and implementations have been produced to test its new features. ZigBee is designed for short distance wireless communication, which is targeted at low data rates, and low-power consumption radio frequency applications, which have the potential to develop industrial and home sensors applications. IEEE 802.15.4 is used as sensor technologies to control and monitor and it's becoming more popular and are therefore in high demand.

The need for machine monitoring is not only present in industrial settings but also at home; thus the importance of ZigBee. Monitoring is crucial in many everyday situations, being used in all different household essentials such as, the refrigerator, power consumption, water, gas, fuel, heating and the list goes on, all of which need to be controlled regularly. However, unlike some sensors, security sensors need to stay active the whole time and cannot just 'wake up' at intervals; with ZigBee this is possible. In a similar way, if the sensor finds any abnormalities the appropriate action needs to be taken. Again, ZigBee is able to deal with any possible scenario. The most important thing though, is that ZigBee is able to do all this whilst remaining low on cost and power consumption but with a range of transmissions and bandwidths. The first version of the standard, introduced in 2003, specifies two physical layers based on Direct Sequence Spread Spectrum (DSSS) techniques, one working in the 868/915 MHz bands with transfer rates of 20 and 40kbps, and the other in the 2450 MHz band with a rate of 250kbps[21], [25].

The 2006 revision of the standard improved the maximum data rates of the 868/915 MHz bands bringing them up to support 100 and 250kbps, and has defined the fourth physical layers. The fourth 868/915 MHz layer is optional and uses a combination of binary keying and amplitude shift keying [25].This division of IEEE 801.15.4 has introduced two standards, IEEE 802.15.4a and IEEE 802.15.4b [27]. In April 2009, IEEE 802.15.4c and IEEE 802.15.4d were released, expanding the available PHYs with several additional PHYs: one for 780MHz [26], beyond these three bands, IEEE 802.15.4c/IEEE 802.15.4d study group is considering the newly opened 314-316 MHz, 430-434 MHz, and 779-787 MHz bands in China. Whilst the IEEE 802.15 Task Group 4D in Japan is defining an amendment to the existing standard IEEE 802.15.4-2006 to support the new 950 MHz-956 MHz band. The first standard amendments by these groups were released in April 2009 [25], [27].

2.1.1 Who developed the 802.15.4 standard and ZigBee

The IEEE has predecessors who are the AIEE (American Institute of Electrical Engineers) and the IRE (Institute of Radio Engineers). From its earliest origins, the IEEE has advanced theory and application of electrotechnology and allied sciences, served as a catalyst for technological innovation and supported the needs of its members through a wide variety of programs and services. The IEEE 802 LAN/MAN Standards Committee develops and maintains networking standards and recommended practices for local, metropolitan, and other area networks, using an open and accredited process, and advocates them on a global basis [28]. The common standards are Ethernet, LANs, Wireless LAN/PA/MAN and Wireless Coexistence.

2.1.2 Why the name ZigBee?

The name ZigBee refers to the waggle dance of honeybees after their return to the beehive; the bees use a zigzag type of dance to communicate information to other hive members. This type of communication behavior (zigzagging) is what set the standard of ZigBee and is what engineers are trying to reach with this protocol [29].

2.1.3 Use of ZigBee

ZigBee technology is intended to be simpler and less expensive than other WPANs such as Bluetooth, UWB and other IEEE wireless standards. In addition, many engineers want to design self-organizing, ad-hoc networks of digital radios to achieve highly efficient connectivity, simultaneously with high data rates and very low costs. ZigBee's low cost allows its technology to be commonly used in wireless architecture; controlling and monitoring applications. The low power-usage allows longer battery life with smaller and lower powered batteries. For example, a battery life of six months to two years on an AA battery would be 0.1% duty cycle with a minimum power output rating of 1mW and no specified maximum. When compared to IEEE 802.15.1 Bluetooth, this is considerably lower [21], [25], [27].

The current developments of ZigBee also focus on mesh network topology. The ZigBee self-organizing feature of the mesh networking on low power consumption is unique and assures high data reliability as well as a larger range of wireless networks without a central node for routing using a mesh of nodes. This self-organizing feature of ZigBee is very advantageous as it reduces the central failure risks, and provides self-healing and better robustness than a wireless static network topology.

2.1.4 ZigBee application areas

There has been tremendous enthusiasm and interest in ZigBee over the past few years as it is a revolutionary new technology built to compliment or replace automation application and automate our household, buildings and industries' machinery. It also supports applications for which other standards are not appropriate. Some applications for this new technology are [30]:

- Home Entertainment and Control: Smart Lighting, Advanced Temperature Control, Safety and Security.
- Home Awareness: Water sensors, power sensors, Smoke and fire detectors, Smart Appliances and Access sensors.
- Mobile Services: M-payment, M-monitoring and control, M-security and Access Control, M-healthcare and Tele-assist.
- Commercial Building: Energy monitoring, Heating, Ventilation, lighting and Access control.

- Industrial Plant: Process control, Asset management, Environmental management, Energy management and Industrial device control.

Each of the named applications may look similar but their application each time is very different. For example, controlling the industrial lighting system is in no way the same as controlling home applications. A common example of how ZigBee can be used within the domestic setting is with plants in a glasshouse. Using a very simple monitoring device the water level and the temperature, for example, can be tracked by 'waking up' nodes at intervals. The nodes would then communicate to the controlling station, which would have the intelligence to increase or decrease the temperature and water the plants when they need it.

In a similar way, ZigBee can be used in industrial settings. Instead of one glasshouse, there could be numerous glasshouses; each with smaller nodes spread out around them. All the nodes could then report to the control station, which will take the appropriate action for each house. Furthermore, other appliances could be used in this situation, such as, thermostats and a motion detector, which could be controlled using nodes in a very similar way.

2.1.5 ZigBee Stack

The ZigBee stack architecture is made up of a set of blocks called layers. Each layer performs a specific set of services for the layer above. For example, as shown in Figure 2.1, a data entity provides a data transmission service and a management entity provides all other services. Each service entity exposes an interface to the upper layer through a Service Access Point (SAP), and each SAP supports a number of service primitives to achieve the required functionality [31].

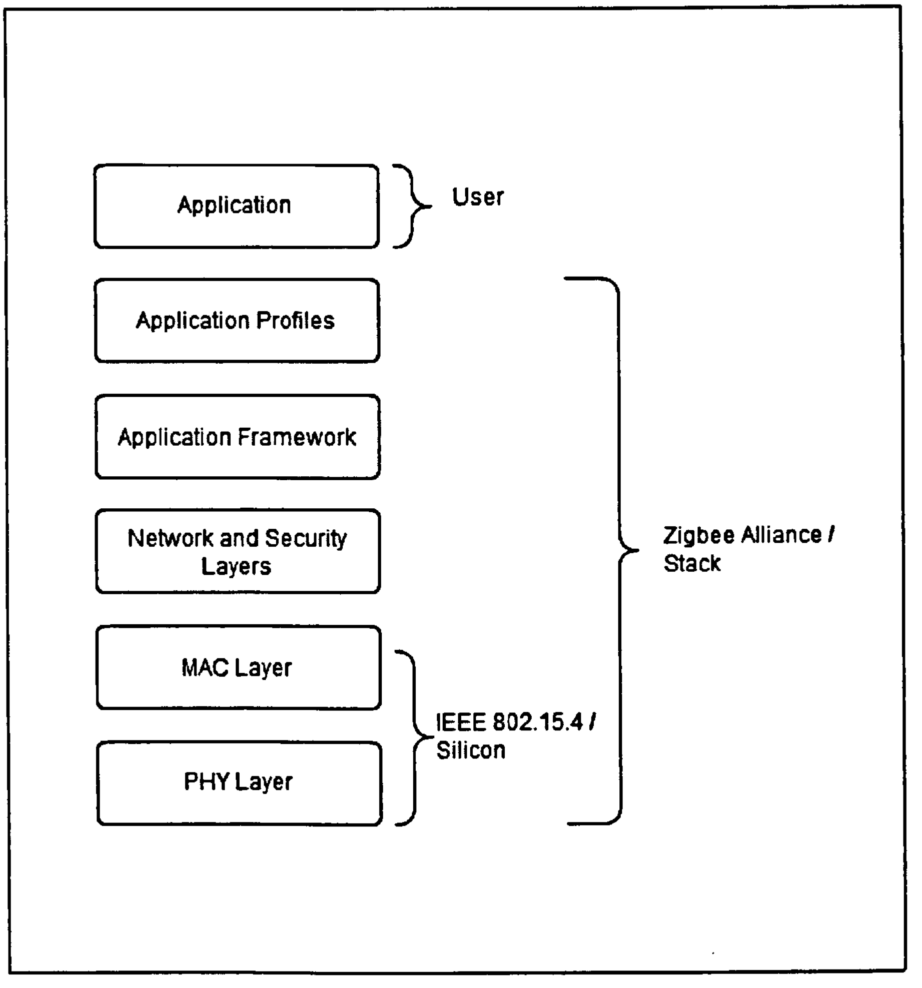


Figure 2.1: IEEE 802.15.4 Stack Layer

2.1.6 ZigBee Stack Components

The ZigBee stack architecture includes a number of layered components and each component provides an application with its own set of services and capabilities. The stack layer is listed as these components:

- Medium Access Control (MAC) layer
- Physical (PHY) layer
- Network (NWK) layer
- IEEE 802.15.4 Application Support (APS) Sub-Layer
- The ZigBee Device Objects (ZDO)

ZigBee's stack architecture is illustrated in Figure 2.2. It is based on the Open Systems Interconnection model with seven-layers but only the layers that function in the intended market space are defined.

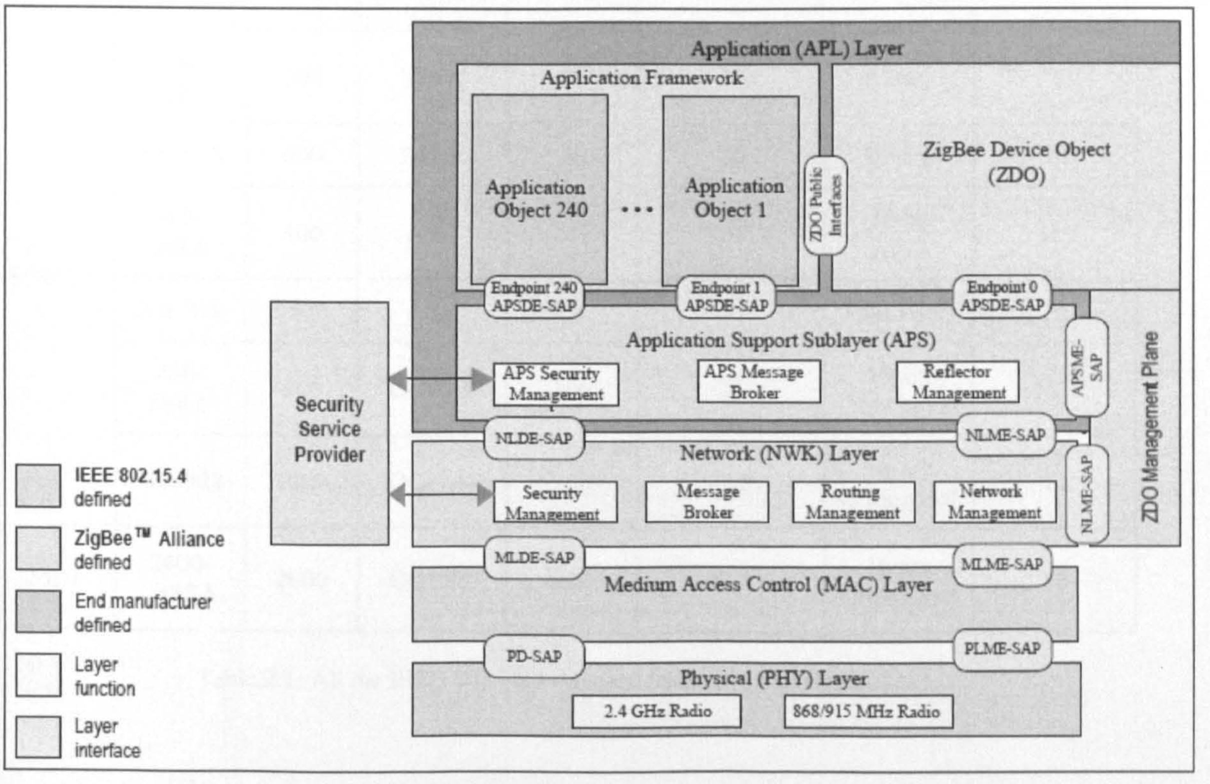


Figure 2.2: Outline of the ZigBee Stack Architecture [32]

2.1.7 Frequency bands and data rates

IEEE 802.15.4 requires that if a transceiver supports the 868 MHz band, it must support 915 MHz band as well, and vice versa. Therefore, these two bands are always bundled together as the 868/915 MHz frequency bands of operation [33]. Table 2.1 refers to details regarding all the frequency bands that are used in the IEEE 802.15.4 standard [32], [33].

PHY (MHz)	Frequency Band (MHz)	Spreading parameters		Data Parameters			
		Chip rate	Modulation	Bit rate (Kbits/s)	Symbol rate (ksymbols/s)	Symbols	Number of Channels
868/915	868-868.6	300	BPSK	20	20	Binary	1
	902-928	600	BPSK	40	40	Binary	10
	868-868.6	400	ASK	250	12.5	20-bit PSSS	1
	902-928	1600	ASK	250	50	5-bit PSSS	10
	868-868.6	400	OQPSK	100	25	16-ary Orthogonal	1
	902-928	1000	OQPSK	250	62.5	16-ary Orthogonal	10
2450	2400-2483.5	2000	OQPSK	250	62.5	16-ary Orthogonal	16

Table 2.1: All the IEEE 802.15.4 standard frequency bands [32], [33]

2.1.8 Coexistence in the 2.4 GHz ISM Band

The IEEE 802.15.4 standard makes use of the license free ISM band [34] at 2.4 GHz. This band is free for any device to use, and one of the most widespread standards for wireless networks. IEEE 802.11 Wi-Fi resides in this frequency band. Another widely used wireless technology, the IEEE 802.15.1 standard known as Bluetooth, also lies in this band. Additionally, other non-networking systems, e.g. microwave ovens, may emit electromagnetic waves in the 2.4GHz band.

A closer look at the interference from IEEE 802.11 stations can give insight into strategies to avoid some of the interference. Each frequency channel in the 802.11b standard spans for 22MHz, and there are 11 such channels from which three are non-overlapping. Therefore, the signal of the IEEE 802.11b interferer can be modeled as a band limited Additive White Gaussian Noise (AWGN) to the IEEE 802.15.4 standard signal [35]. This is due to the different bandwidths used by the two standards.

The IEEE 802.15.4 standard employs frequency channels of 2MHz bandwidth that is one eleventh of the IEEE 802.11b stations. Figure 2.3 shows an illustration of the frequency spectrum relationship of IEEE 802.15.4 and IEEE 802.11 [36], [37].

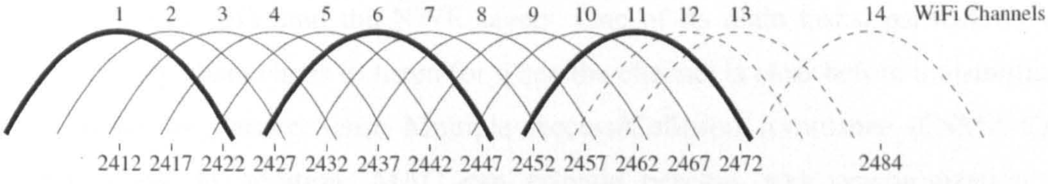


Figure 2.3: IEEE 802.15.4 and 802.11b/g and 802.15.4 frequency channels [37]

As shown in Figure 2.4 if the same carrier frequencies are selected, the impact of IEEE 802.11 and IEEE 802.15.1 with high traffic rate against IEEE 802.15.3 stations will be extremely critical, as there will be a major overlap.

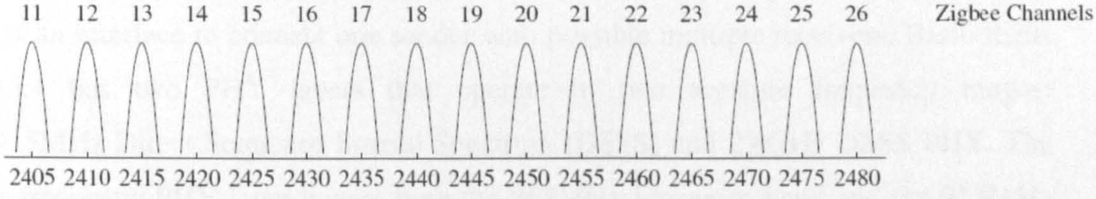


Figure 2.4: ZigBee channels

The interference can be avoided by selecting ZigBee channels shown in Figure 2.5 that use the free space between two neighboring 802.11 channels, plus channels #25 and #26 [38]. Hence, the WLAN interferer would cover the entire bandwidth of the IEEE 802.15.4 standard device, whereas, in the opposite case the interferer only affects parts of the bandwidth.

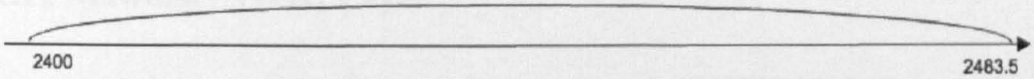


Figure 2.5: IEEE 802.15.4 and 802.15.1/Bluetooth frequency channel

2.1.9 Medium Access Control (MAC) Layer

The MAC layer provides services to enable reliable, single-hop communication links between devices, as well as reliable communications between a node and its immediate neighbors such as PHY and the NWK layers. One of its main tasks, particularly on shared frequency channels, is to listen for when the channel is clear before transmitting. This is known as Carrier Sense Multiple Access/Collision Avoidance (CSMA-CA) communications. In addition, MAC can provide beacons and synchronization to improve communications efficiency. The MAC layer also manages the packing of data into frames prior to transmission, and then the unpacking of received packets and the checking of them for errors [39].

2.1.10 Physical (PHY) Layer

The PHY layer provides the basic communication capabilities of the physical radio, as well as an interface to connect one sender with possible multiple receivers. Basic IEEE 802.15.4 has two PHY layers that operate in two separate frequency ranges: 868/915MHz Direct Sequence Spread Spectrum (DSSS) and 2.4GHz DSSS PHY. The lower frequency PHY layer covers both the 868MHz European band and the 915MHz band that is used in countries such as the United States and Australia [40]. The data rate is 250kbps at 2.4GHz, 40kbps at 915MHz and 20kbps at 868MHz. The higher data rate at 2.4GHz is attributed to a higher-order modulation scheme. Lower frequency provides longer range due to lower propagation losses. Lower rates can be translated into better sensitivity and larger coverage area. Higher rates mean higher throughput, lower latency or lower duty cycle [36], [37].

2.1.11 Network (NWK) Layer

The network layer ensures the correct operation of the IEEE 802.15.4 standard to provide a suitable service interface to the application layer. In order to interface with the application layer, the network layer conceptually includes two service entities that provide the necessary functionality. The NWK also provides routing and the multi-hop functions needed for creating different network topologies; for example, star, tree, and mesh structures. The application layer includes an Application Support (APS) sub-layer, the ZigBee Device Object (ZDO), and applications [33], [41].

2.1.12 Application Support (APS) Sub-Layer

The APS provides the interface between the network layer and the application layer through a general set of services for use by both the ZDO and the manufacturer-defined application objects [33]. The responsibilities of the APS sub-layer include:

- Maintaining tables for binding, that is; the ability to match two devices together based on their services and needs, and forwarding messages between bound devices [33].
- Group address definition, removal and filtering of group addressed messages, address mapping from 64 bit IEEE addresses to and from 16 bit NWK addresses. Fragmentation, reassembly and reliable data transport [33].

2.1.13 The ZigBee Device Object (ZDO)

The ZigBee Device Object supplies object interface layer to network and application framework. It fulfills all the requirements of any applications in the ZigBee stack, even the Routers, Coordinators and End Devices. The ZDO is responsible for the following [38]:

- Initializing the Application Support Sub-layer (APS), the Network Layer (NWK), and the Security Service Provider.
- Assembling configuration information from the end applications to determine and implement discovery, security management, network management, and binding management.
- Defining the role of the device within the network (e.g. coordinator or end device), discovering devices on the network and determining which application services they provide.

2.1.14 ZigBee Data Entity

The ZigBee data entity provides a data transmission service and a management entity provides all other services [44]. Each service entity exposes an interface to the upper layer through a Service Access Point (SAP), and each SAP supports a number of service primitives to achieve the required functionality [31].

2.1.15 Application Level

The top layer of the IEEE 802.15.4 standard is the Application Level, which runs on the network node, coordinator or router and makes the device functional. The Application layer is illustrated in Figure 2.6.

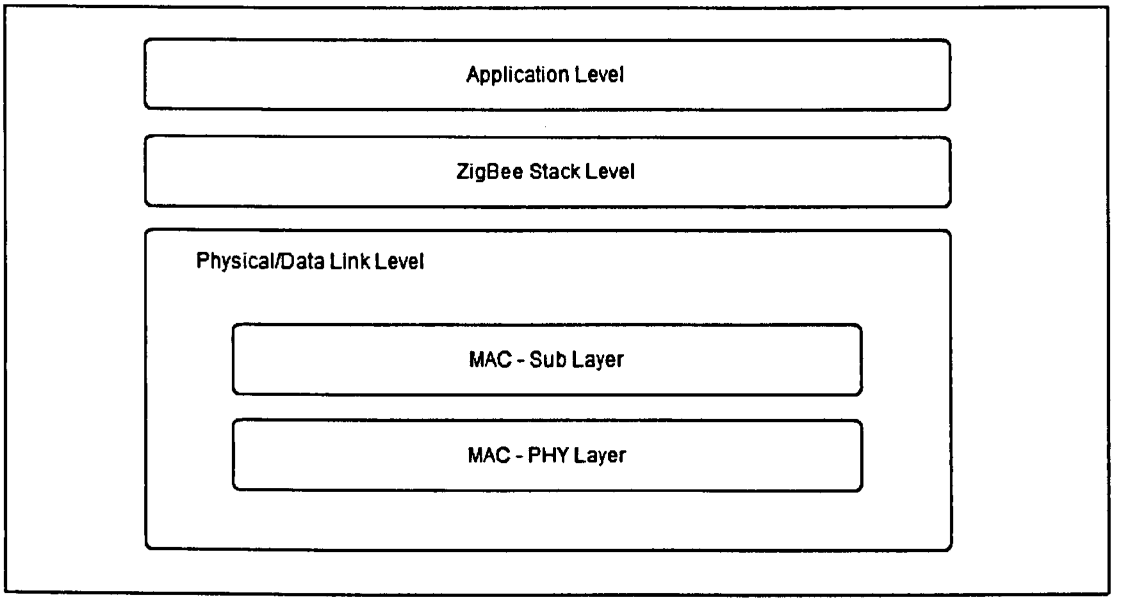


Figure 2.6: ZigBee basic layer model

2.1.16 Modulation

The IEEE 802.15.4 standard provides three modulation types: Binary Phase Shift Keying (BPSK), Amplitude Shift Keying (ASK), and Offset Quadrature Phase Shift Keying (OQPSK). The ASK and OQPSK are optional. In BPSK and OQPSK, the digital data is in the phase of the signal. In contrast, in ASK, the digital data is in the amplitude of the signal [33].

2.1.17 Binary phase-shift keying (BPSK)

Binary Phase-shift keying (BPSK) is a digital modulation scheme that conveys data by changing, or modulating the phase of a reference signal (the carrier wave). BPSK is appropriate for low-cost passive transmitters and the BPSK is simplest form of phase shift keying (PSK) [42]. Binary phase shift keying uses two-phase separated by 180 degrees [33], [42].

2.1.18 Amplitude-shift Keying

Amplitude-shift Keying modulates digital data as changes in the amplitude of a carrier wave takes place. The ASK transmission model is very simple and in ASK modulation, the information is embedded in the signal amplitude instead of the signal phase [33]. The IEEE 802.15.4 standard utilizes root-raised cosine pulses as defined in Equation 2.1 [45]:

$$
\begin{cases}
\dfrac{\left\{\pi(r+1).\sin\left(\frac{\pi}{4}\frac{(r+1)}{r}\right)+\pi(r-1).\cos\left(\frac{\pi}{4}\frac{(r-1)}{r}\right)-4r.\sin\left(\frac{\pi}{4}\frac{(r-1)}{r}\right)\right\}}{2\pi\sqrt{T_c}} \\
\dfrac{4r}{\pi\sqrt{T_c}}-\dfrac{(r-1)}{\sqrt{T_c}}, t=0 \\
4r\dfrac{\cos\left(\frac{(1+r)\pi t}{T_c}\right)+\sin\dfrac{\left(\frac{(1-r)\pi t}{T_c}\right)}{\left(\frac{4rt}{T_c}\right)}}{\pi\sqrt{T_c}\left(1-\left(\frac{4rt}{T_c}\right)^2\right)}, t \neq 0 \; and \; t \neq \left(\frac{\pm T_c}{(4r)}\right)
\end{cases}
$$

(2.1)

The pulses are multiplied by a PSSS code table, which effectively spreads the signals over the operating band [45].

2.1.19 Offset Quadrature Phase-shift Keying

The IEEE 802.15.4 standard can also use OQPSK modulation in the 868/915 MHz band. The form of OQPSK used in IEEE 802.15.4 transmits data in three steps [45].

- The first stage maps bits to one of the symbols from the 16-ary orthogonal constellation.
- The symbols are then phase-modulated over the band using a form of Direct Sequence Spread Spectrum modulation.
- The even indexed coded sequences are then modulated onto the in-phase (I) carrier wave, and the odd indexed coded sequences are modulated onto the quadrature carrier wave (Q).

For this modulation scheme, half-sine wave pulses are used as formulated in Equation 2.2 [45]:

$$(t) = \begin{cases} \sin\left(\pi \frac{t}{2T_c}\right), 0 \leq t \leq 2T_c \\ 0, otherwise \end{cases} \tag{2.2}$$

2.1.20 IEEE 802.15.4 Data Transport Basics

IEEE 802.15.4 primary data type has been defined as a unit of frame for data transfer. In IEEE 802.15.4 every frame type is distinct by the standard to facilitate data transfer. Four frame types are defined and known as; data, ACK, beacon and MAC. Data and ACK frames are bound together to ensure data transfer over a noisy channel where packets may drop. Data transfers to a coordinator require a beacon frame to synchronize the devices. The MAC frame acts as an addressing system to the network, just as MAC works for Ethernet [46].

2.1.21 Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA)

IEEE 802.15.4 implements CSMA-CA method to allow multiple devices to use the same frequency channel and listen to the network. In order to avoid collisions for its communication medium, before any real data is transmitted, it has to broadcast a signal onto the network in order to listen for collision scenarios and to tell other devices not to broadcast. With CSMA-CA, anytime a device wants to transmit, it first performs a clear channel assessment (CCA) to ensure that the channel is not in use by any other device, and then the device starts transmitting its own signal [37]. If the channel is active then the device will continue waiting until it is idle [45]. This algorithm is illustrated in Figure 2.7.

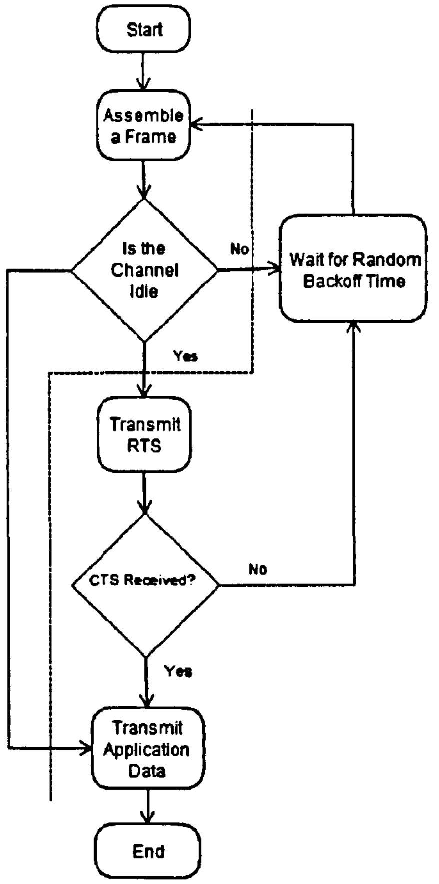


Figure 2.7: CSMA/CA Algorithm

The decision to declare a channel clear or not can be based on measuring the spectral energy in the frequency channel of interest or detecting the type of occupying signal [33]. If the channel is not clear, the device backs off for a random period and tries again. The random back off and retry are repeated until either the channel becomes clear or the device reaches its user-defined maximum number of retries [33].

2.1.22 ZigBee Operation Modes

ZigBee employs two methods of channel access beacon and non-beacon, which enables the to-and-fro data traffic. Beacon is a type of message with definite format that is used to synchronize the clocks of the nodes in the network [33]. A typical beacon frame is approximately fifty bytes long, with about half of that being a common frame header along with additional information. As with other frames, the header includes source and destination MAC addresses as well as other information regarding the communications process. The destination address is always set to all ones, which is the broadcast Medium Access Control (MAC) address [46].

2.1.23 Beacon-Enabled vs. Non-beacon Networking

Beacon mode allows both the coordinator and beacon to 'wake up' only at intervals. In beacon enabled PAN networks, a coordinator has the option to transmit beacon signals to synchronize the devices attached to it and thus has higher power consumption, as all the devices in the network must wake up on a regular basis, listen for the beacon, synchronize their clocks, and go back to sleep just as the coordinator switches into sleep mode when it is no longer in use. This means that many of the devices in the network may wake up only for synchronization and not perform any other tasks while they are active [33], [45]. The coordinator showing in Figure 2.8, sends out messages which the beacon looks for and once the message transmission is complete it goes back to sleep.

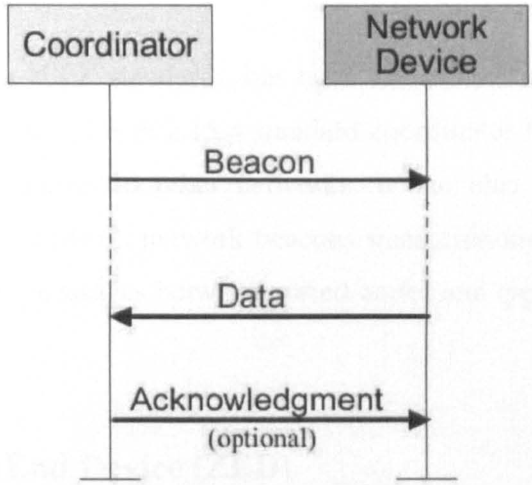


Figure 2.8: Beacon Data Transfer to a Coordinator in IEEE 802.15.4 [33]

The non-beacon mode shown in Figure 2.9 [33] finds favor when the coordinator is mains-powered. A network in which the PAN coordinator does not transmit beacons is known as a non-beacon network. The battery life in a non-beacon network can be noticeably better than in a beacon-enabled network because in a non-beacon network the devices wake up less often [41], [45].

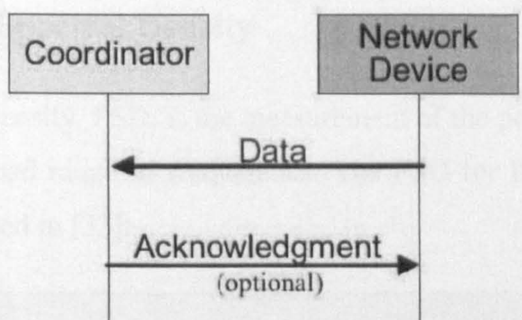


Figure 2.9: Nonbeacon Data Transfer to a Coordinator in IEEE 802.15.4 [33]

2.1.24 ZigBee coordinator (ZC)

ZigBee, like other IEEE standards, has been researched carefully and therefore is very well developed. The IEEE 802.15.4 standard coordinator forms the root of the network tree and bridges across to other networks. It can also store information about the network: set up a network, network beacons transmissions, managing the nodes within the network, route messages between paired nodes and typically operates in the receive state.

2.1.25 ZigBee End Device (ZED)

The ZigBee End Device can communicate with the parent node, the coordinator or router, but it cannot pass on data from other devices. This relationship allows the node to be asleep a significant amount of the time thereby giving longer battery life [33]. It is not just simplicity and low cost production that makes ZigBee an essential standard, it is also the unique feature of self-organizing in mesh networks (which can be used as mentioned above) which is outstanding and distinguishes it in comparison with any other low power wireless standard.

2.1.26 Power Spectral Density

Power Spectral Density, PSD, is the measurement of the power of a given packet of data spread over a broad range of frequencies. The PSD for IEEE 802.15.4 must meet the criteria summarized in [33]:

Frequency Band	Frequency Offset	Relative Limit	Absolute Limit
2.4GHz	$\lvert f - f_c \rvert > 3.5$MHZ	-20dBC	-30dB
915MHz	$\lvert f - f_c \rvert > 1.2$MHZ	-20dBc	-20dB
868	N/A	N/A	N/A

Table 2.2: IEEE 802.15.4 PSD Limits [33]

During PSD measurement, the resolution bandwidth must be 100 KHz. The peak power is the highest average power measured within 1 MHz of the carrier frequency in the

2.4GHz band and within 600 KHz of the career frequency in the 915 MHz band. For the 868 MHz band, since there is no adjacent channel, the only criterion is that a raised cosine filters the signal before transmission [33].

2.1.27 ZigBee Network Topologies

The network topology that best suits the application is an important factor and the supported topologies in IEEE 802.15.4 include: Peer to Peer (Ad-hoc), Star Configuration, Cluster Tree and Multi-hop network

2.1.28 Peer to Peer Topology (Ad-hoc)

ZigBee's peer-to-peer topology allows any device to communicate with any other device that is in range of it. In Figure 2.10 ZigBee's nodes connect directly to each other for peer–to-peer communication. Figure 2.11 illustrates Peer to Peer with a Pan Coordinator.

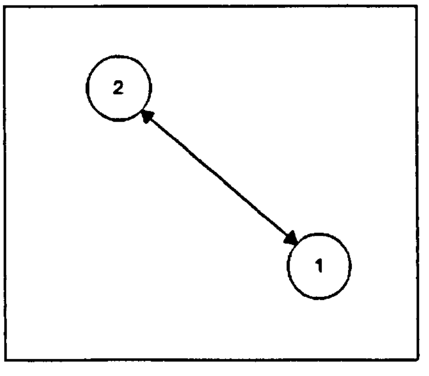


Figure 2.10: IEEE 802.15.4 Peer to Peer (Ad-hoc)

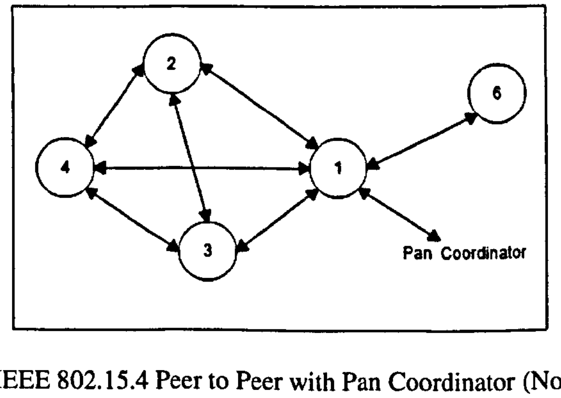


Figure 2.11: IEEE 802.15.4 Peer to Peer with Pan Coordinator (Node number 5)

2.1.29 Star Configuration

ZigBee supports star, mesh, and cluster-tree network topologies. In these topologies, when data reliability is crucial, star, mesh, and cluster-tree architectures provide the best shield against signal degradation and loss of data. On the other hand, few shorter range applications would be best suited to a hierarchical tree or star topology where the overheads of mesh network are not required [48], [49]. Figure 2.12 shows a PAN coordinator in star topology where each node is connected directly to the central coordinator and therefore all inter-node communications must pass through the coordinator.

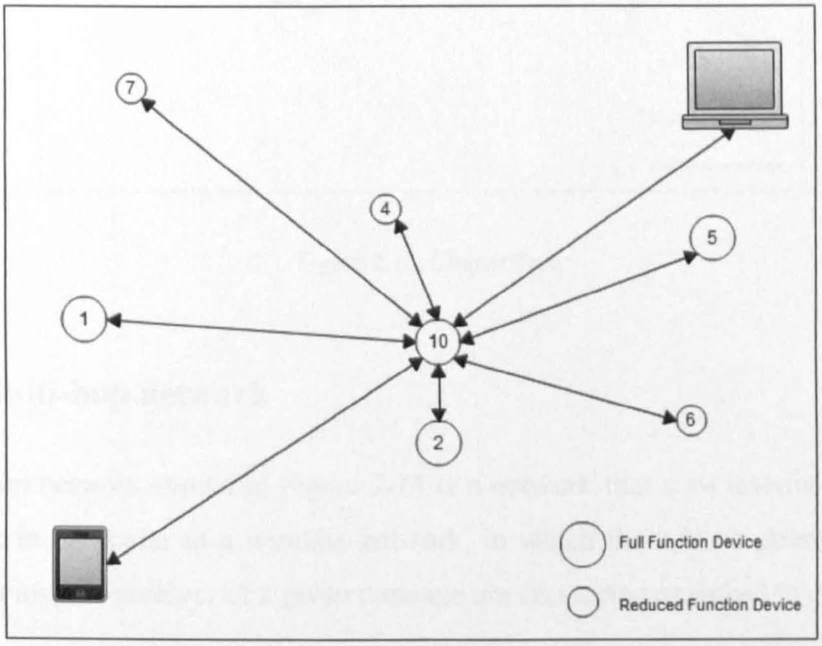


Figure 2.12: Star Configuration

2.1.30 Cluster Tree

A cluster tree network consists of a number of star networks connected together, whose central nodes are also in direct communication with the single PAN Coordinator [50]. Using a set of routers and a single PAN coordinator, the network is formed into an interconnected mesh of routers and end nodes that pass information from node to node using the most cost effective path. As Figure 2.13 demonstrates if any individual router becomes inaccessible then alternative routes can be discovered and used; providing robust and reliable network topography [50].

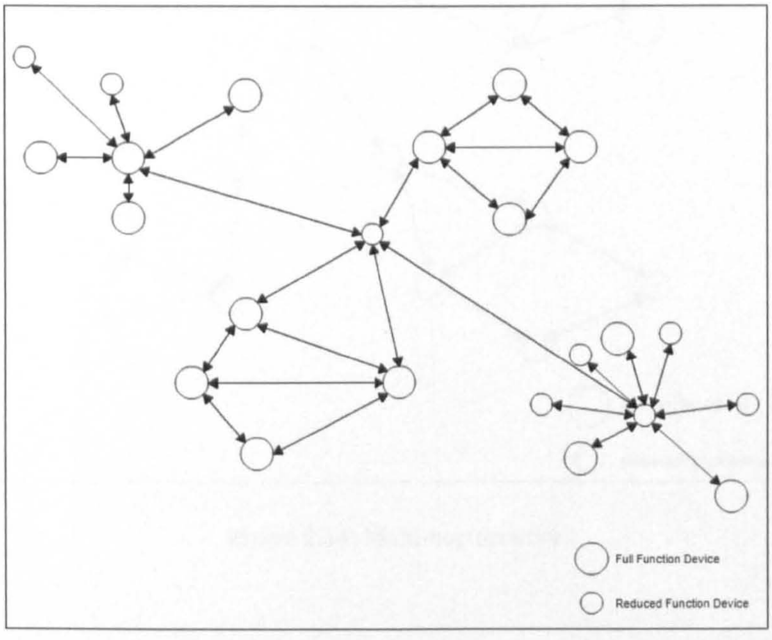


Figure 2.13: Cluster Tree

2.1.31 Multi-hop network

A multi-hop network shown in Figure 2.14 is a network that uses intermediate devices as routers, in particular in a wireless network, in which there is no guarantee that the transmitter and the receiver of a given message are connected or linked to each other.

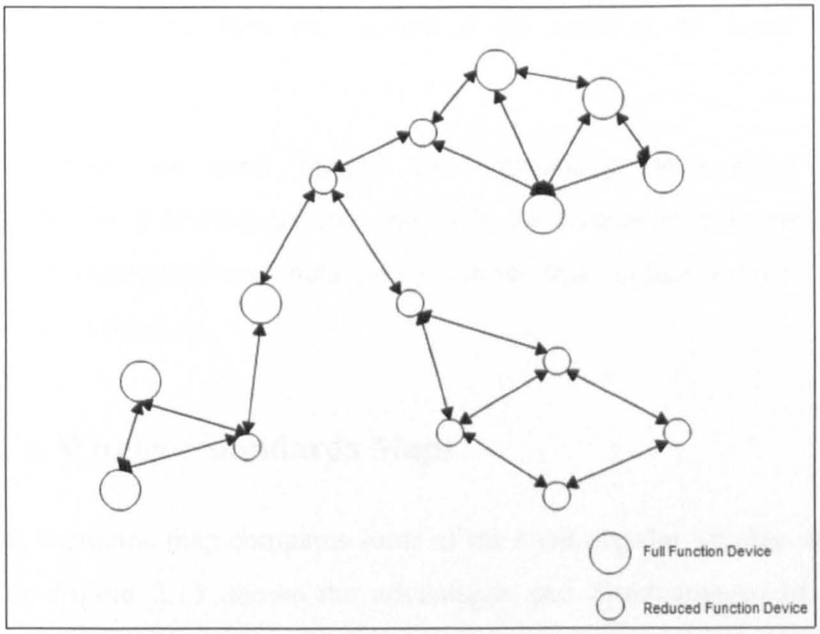


Figure 2.14: Multi-hop network

2.1.32 Encryption and Security

The IEEE 802.15.4 standard operates in the 2.4GHz band, performs medium access with the same MAC protocol and often co-exists in the overlapping coverage area. The IEEE 802.15.4 standard defines a protection mechanism to avoid frame collisions between itself and other IEEE 802 standards. IEEE 802.15.4 uses the frame-protection mechanism inherited from IEEE 802.11b [51]. IEEE 802.15.4 also makes a use of Advanced Encryption Standard in sub-clause with very low-power operation [32].

2.1.33 IEEE 802.15.4 Contenders

There are so many wireless technologies already; it seems strange that another is needed. However, no wireless technologies have fulfilled their potential of producing a low power and low data technology, something that ZigBee aims to do.

Existent technologies, such as WI-FI, UWB, Bluetooth and WIMAX, have been designed purely based on the need for high data rates. Power consumption has not been considered in these cases, which affects the size and therefore the cost. This is because the more complex the application the more expensive they are to produce and use. None of these technologies are, therefore, suited to the needs of the home or industrial automation.

These technologies are used mainly for networking, automation and sensor technologies. In the following section, they will be compared in terms of their data range, power consumption and data rate to show that, unlike ZigBee, they fail to provide a practical solution.

2.1.34 IEEE Wireless Standards Maps

This wireless standards map compares some of the most popular wireless standards and the following Figure 2.15 shows the advantages and disadvantages of the existent standards and their common application areas in terms of data range and data rate.

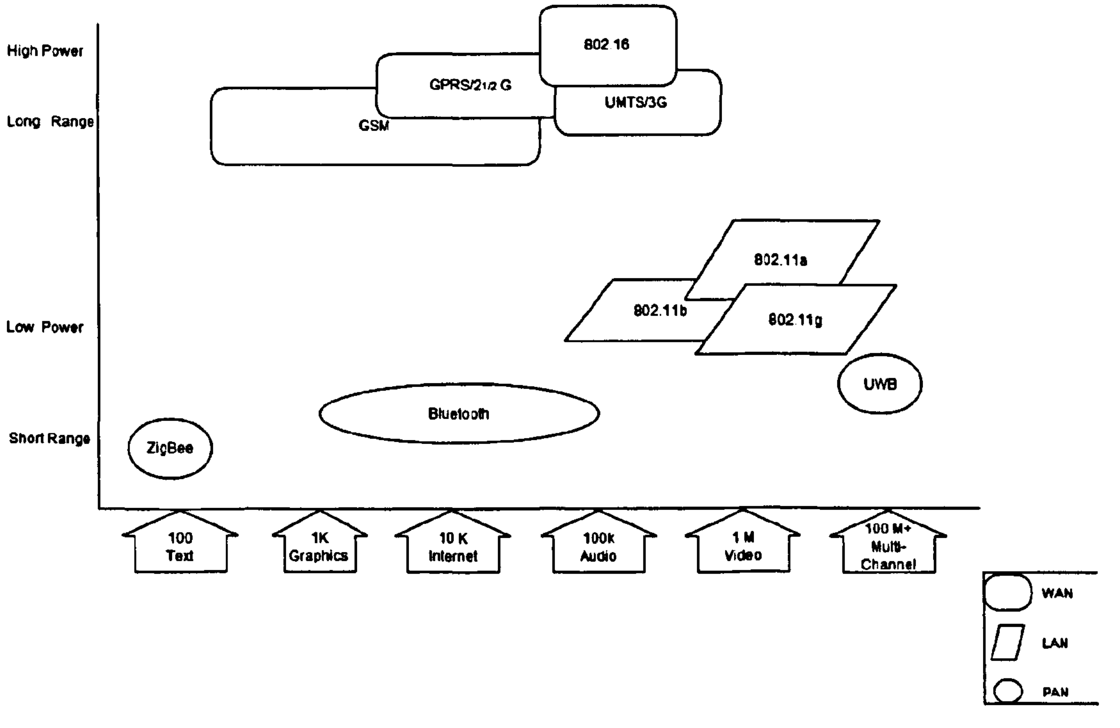


Figure 2.15: ZigBee versus others IEEE wireless standards data rates, data range and power consumption

The following Figure 2.16 shows the maximum data rates, data range and power consumption for various technologies.

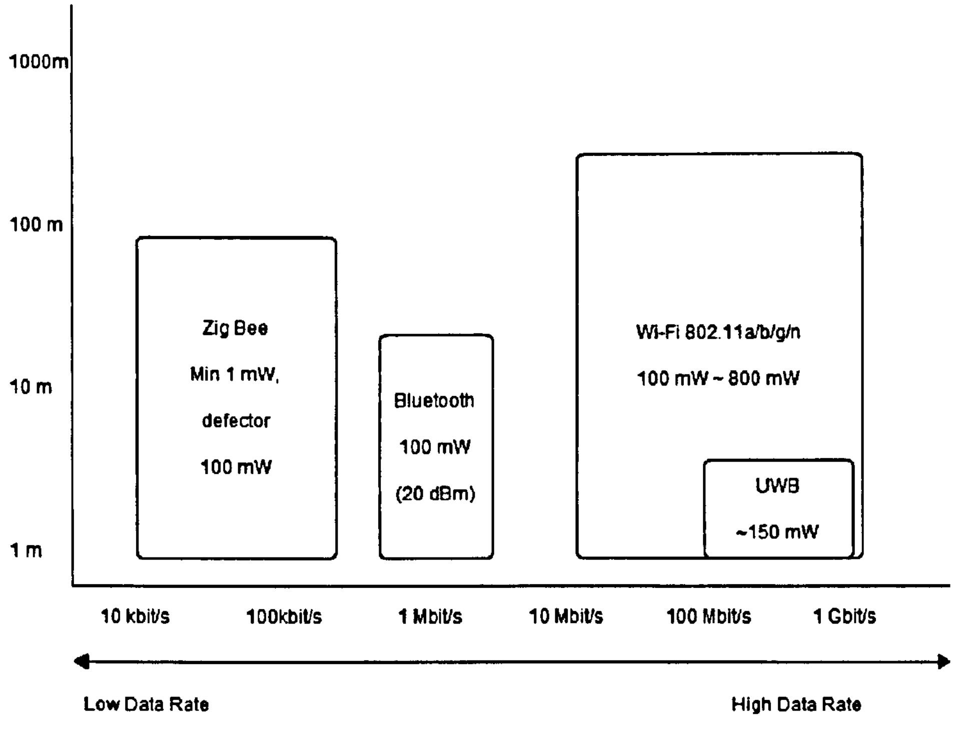


Figure 2.16: Data rates, data range and power consumption map versus other IEEE 802.xxx

2.1.35 ZigBee versus WI-FI

A WI-FI enables devices, such as PC, mobile phone or tablets to connect to the internet wirelessly. WI-FI 802.11 2.4GHz with 54Mbit/s to maximum of 600Mbit/s has a higher bandwidth than most of the other IEEE's standard. This makes the standard suitable for large file or Audio/Video transmissions. However, it is power thirsty and has never been aimed at use in home or industrial automation. Table 2.3 outlines some of the key characteristics of ZigBee compared to WI-FI wireless standards.

	ZigBee	802.11 (WI-FI)
Data Rate	20, 40, and 250 Kbits/sec (250kbps (peak information rate 128kbps))	11 & 54Mbits/sec
Range	10-100 meters	50-100 meters
Networking Topology	Ad-hoc, peer to peer, star, or mesh	Point to hub
Operating Frequency	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	2.4 and 5 GHz
Complexity (Device and application impact)	Low	High
Power Consumption (Battery option and life)	Very low (low power is a design goal)	High
Security	128 AES plus application layer security	High
Other Information	Devices can join an existing network in under 30ms	Device connection requires 3-5 seconds

Table 2.3: The key characteristics of ZigBee versus WI-FI

2.1.36 ZigBee versus Bluetooth

Bluetooth is an open wireless protocol for short-range communication and data exchanging from fixed and mobile devices, creating Personal Area Networks (PANs), as well as being the first known wireless standard for low data rate applications. It was intended to replace cables connecting portable and/or fixed electronic devices alternative to RS232 data cables [52] as a wireless and with the several device connectivity features low power and low cost technology with frequency hopping, to overcome the problems of synchronization.

Bluetooth key features are less complex when compared to WI-FI and utilizes the unlicensed 2.4GHz ISM Band, with frequency hopping and avoids interference by hopping to a new frequency 1600 times a second by using small packet sizes [53]. It has a range of over 10 meters and can easily extend to 100 meters with a power boost. It can transfer data at a maximum range of 720kbps.

Bluetooth can be considered as a good contender for automation and sensorial based application as they are low power and low cost, therefore, it is relevant to compare

Bluetooth and ZigBee, as they are sometimes seen as competitors. In this section their differences in order to clarify which applications suit each of them is highlighted.

The data transfer capabilities are much higher in Bluetooth than in ZigBee. Bluetooth is capable of transmitting audio, graphics and pictures over small networks, and it is also appropriate for file transfers. ZigBee, on the other hand, is better suited for transmitting smaller packets over large networks; mostly static networks with many, infrequently used devices, such as; home automation, toys, remote controls, etc. While the performance of a Bluetooth network drops when more than eight devices are present, ZigBee networks can handle 65000+ devices [55].

Bluetooth aims to cover more applications and provides a quality of service (QoS) which has pushed its design goal away from the simplicity that was intended in the first place. The complexity of Bluetooth makes it expensive and unsuitable for some applications requiring low-cost and low power. In addition, Bluetooth is faced with a lack of flexibility in the topologies, as research shows scalability problems [55], which means its construct is not a perfect contender for ZigBee.

As mentioned before Bluetooth is a cable replacement for items like phones, laptops, computers and headset devices that expect regular charging or battery replacements. Whereas, the main feature of ZigBee is a low and limited power requirement, which as result would be a better option for devices where the battery is rarely replaced. The ZigBee's join time for a new slave is typically 30ms, and the time needed by a slave to change from sleeping to active, or accessing the channel is typically 15ms. Bluetooth devices need 3 seconds to both join a network or to change to active from a sleeping state, though they are much faster at accessing the channel around 2ms.

Table 2.4 outlines some of the key characteristics of ZigBee and how compared with Bluetooth it has many more advantages.

	ZigBee	Bluetooth
Data Rate	20, 40, and 250 Kbits/s	1 Mbits/s
Range	10-100 meters	10 meters
Networking Topology	Ad-hoc, peer to peer, star, or mesh	Ad-hoc, very small networks
Operating Frequency	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	2.4 GHz
Complexity (Device and application impact)	Low	High
Power Consumption (Battery option and life)	Very low (low power is a design goal)	Medium
Security	128 AES plus application layer security	64 and 128 bit encryption
Other Information	Devices can join an existing network in under 30ms	Device connection requires up to 10 seconds
Chip rate	11 chips/ symbol	1600 hops / second
Symbol rate (K symbols/s)	62.5 K symbols/s	1 M Symbol / second
Symbols	DSSS - 4 Bits/ symbol	FHSS (Frequency-hopping spread spectrum) - 1 bit/symbol

Table 2.4: ZigBee versus Bluetooth

2.1.37 Z-Wave

Z-Wave is a wireless communications proprietary standard designed for home automation, specifically for remote control applications in residential and light commercial environments. This technology, which was developed by Sigma Designs' Zensys, uses a low power RF radio embedded or retrofitted into home electronics devices and systems, such as lighting, home access control, entertainment systems and household appliances. The technology has been standardized by the Z-Wave Alliance; an international consortium of manufacturers that oversees interoperability between Z-Wave products and enabled devices [57]. ZigBee and Z-Wave are very similar as compared in Table 2.5, and both target the same industry and applications. ZigBee is more multipurpose and it can even be used for any short-range wireless task, but it has a very complex protocol; resulting in longer development times. Z-Wave uses a far simpler protocol, supporting a maximum of 40kbits/s, which is much less than ZigBee's 250kbps bandwidth.

	ZigBee	**Z-Wave**
Data Rate	20, 40, and 250kbps	9.6/40 kbps
Range	10-100 meters	Max 30 meters
Networking Topology	Ad-hoc, peer to peer, star, or mesh	Ad-hoc, peer to peer
Operating Frequency	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	908.42 GHz
Power Consumption (Battery option and life)	Very low (low power is a design goal)	Low
Security	128 bit AES plus application layer security	128 bit AES encryption
Modulation	OQPSK	GFSK

Table 2.5: ZigBee versus Z-Wave

2.1.38 ZigBee versus Ultra-wideband

Ultra-wideband (UWB) is a radio technology that can be used at very low energy levels for short-range, high-bandwidth communications [58], [59]. UWB uses orthogonal frequency division modulation and direct sequencing to send out very short, fast, low power pulses of energy spread over a wide range of frequencies. In contrast to ZigBee, UWB uses every frequency available to it at the same time. It provides an efficient use of limited radio bandwidth while enabling both high data rate PAN, wireless connectivity and longer-range, low data rate applications as well as radar and imaging systems. Nevertheless, this requires interference and time synchronization, which would not be ideal for the simple and low power conception applications and as UWB intends to provide QoS for its application this has pushed its design goal from a simple to a very complex structure. In addition, it lacks security making it unsuitable for various applications requiring high security. Table 2.6 outlines some key characteristics of ZigBee and UWB standards and highlights ZigBee's advantages.

	ZigBee	UWB
Data Rate	20, 40, and 250 Kbits/s	100-500 Mbits/s
Range	10-100 meters	<10 meters
Networking Topology	Ad-hoc, peer to peer, star, or mesh	Point to point
Operating Frequency	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	3.1-10.6 GHz
Complexity (Device and application impact)	Low	Medium
Power Consumption (Battery option and life)	Very low (low power is a design goal)	Low
Security	128 AES plus application layer security	N/A

Table 2.6: ZigBee versus UWB

2.1.39 ZigBee versus other Wireless Standards

Table 2.7 outlines some key characteristics of ZigBee and how it is distinguished from other common wireless standards.

	ZigBee	Wireless USB	IR Wireless
Data Rate	20, 40, and 250 Kbits/s	62.5 Kbits/s	20-40 Kbits/s 115 Kbits/s 4 & 16 Mbits/s
Range	10-100 meters	10 meters	<10 meters (line of sight)
Networking Topology	Ad-hoc, peer to peer, star, or mesh	Point to point	Point to point
Operating Frequency	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	2.4 GHz	800-900 nm
Complexity	Low	Low	Low
Power Consumption	Very low	Low	Low
Security	128 AES plus application layer security		
Other Information	Devices can join an existing network in under 30ms		

Table 2.7: ZigBee versus other Wireless Standards

2.1.40 Power Consumption in Common Wireless IEEE Standards

None of the compared standards as presented in Table 2.8, have the same reliability required for the automation applications as ZigBee. Robustness in terms of critical application scenarios as applicable to industrial needs and reliability is important for power usage and prompt response.

Type	Bit rate	TX Power
802.11a/b/g	11 ~ 54Mb	50 ~ 200mW
802.15.1	1Mb	1mW
802.15.3	55Mb	1mW

Table 2.8: Power consumption in common IEEE 802.xxx

2.2 MPEG

MPEG is a video CODEC standardized by the Moving Picture Expert Group (MPEG) [60] for low bit rate digital media applications. There are two primary standard organizations driving the definition of image and video compression standards. A) The International Telecommunications Union which is focused on telecommunication applications and has created the H.26x standards for video telephony and B) The Internal Standards Organization; which is more focused on consumer applications and has defined the JPEG standards for still image compression and MPEG standards for compressing moving pictures [60].

Compression involves a complementary pair of systems, a compressor (encoder) and a decompressor (decoder) as illustrated in Figure 2.17. The encoder converts the source data into a compressed form (occupying a reduced number of bits) prior to transmission or storage and the decoder converts the compressed form back into a representation of the original video data [61].

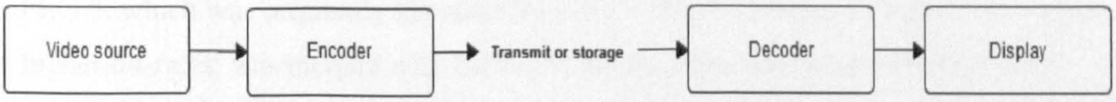


Figure 2.17: Encoder/Decoder

Data compression is achieved by removing redundancy, i.e. components that are not necessary for faithful reproduction of the data. Many types of data contain statistical redundancy and can be effectively compressed using lossless compression, so that the reconstructed data at the output of the decoder is a perfect copy of the original data. Unfortunately, lossless compression of image and video information gives only a moderate amount of compression [61].

2.2.1 MPEG-1

MPEG-1 is the first generation of the MPEG video compression algorithm. The motivation behind the application was storage and retrieval of moving pictures and audio on digital media such as video CDs using Source Input Format (SIF) resolution (352x240) at 30 fps. The targeted output bit rate was 1.15 Mbps, which produces effectively 25:1 compression. MPEG-1 is similar to H.261 but encoders typically require more performance to support the heavier motion found in movie content versus typical video telephony [62].

2.2.2 MPEG-2

After the development of MPEG-1, MPEG-2 was developed by the MPEG committee in order to further improve video and audio coding standards. The idea of this second phase of MPEG work came from the fact that MPEG-1 is optimized for applications at about 1.5 Mb/s with input source in SIF, which is a relatively low solution progressive format. Many higher quality, higher bit-rate applications require a higher resolution digital video source, such as CCIR601, which is an interlaced format [63].

2.2.3 MPEG-3

MPEG-3, which was originally intended for HDTV (High Definition Digital Television) at higher bit-rates, was merged with MPEG-2. Hence there now is no MPEG-3 [63].

2.2.4 MPEG-4

MPEG-4 has been developed in a similar way to MPEG-1 and MPEG-2. However, it provides far greater flexibility than is possible with other technologies and ensures the content produced is reusable. MPEG-4 offers transparent information, which can be interpreted and translated into the appropriate native signaling messages of each network with the help of relevant standard bodies [64]. The foregoing, however, excludes Quality of Service considerations, for which MPEG-4 provides a generic QoS descriptor for different MPEG-4 media [65]. The exact translations from the QoS parameters set for each media to the network QoS are beyond the scope of MPEG-4 and are left to network providers. Signaling of the MPEG-4 media QoS descriptors end-to-end enables transport optimization in heterogeneous networks [64].

MPEG-4's objectives are beyond "plain" compression. Instead of regarding video as a sequence of frames with fixed shape and size, with attached audio information, the video scene is regarded as a set of dynamic objects. Thus the background of the scene might be one object, a moving car another, the sound of the engine the third etc. The objects are spatially and temporally independent and therefore can be stored, transferred and manipulated independently [66].

2.2.5 MPEG Video Compression, Encoder and Decoder

The MPEG visual standards CODEC 'model' shown in Figure 2.18 uses block-based motion compensation, transform, quantization and entropy coding which known as MPEG encoder. The main components of this model, starting with the temporal model (motion estimation and compensation) and continuing with image transforms quantization, predictive coding and entropy coding [61]. The entropy is the constructive part while the redundancy is the rest, including the temporal, spatial and frequency side. The MPEG video-coding algorithm uses a Discrete Cosine Transform (DCT) that is block based and two-dimensional. A picture is first broken down to 8x8 blocks and to each block the DCT is then applied [67].

A quantizer is applied to the DCT coefficients, which sets many of them to zero. This quantization is responsible for the lousy nature of the compression algorithms in JPEG, H.261 and MPEG-1 video. Compression is achieved by transmitting only the coefficients that survive the quantization operation and by entropy coding their locations and amplitudes [67].

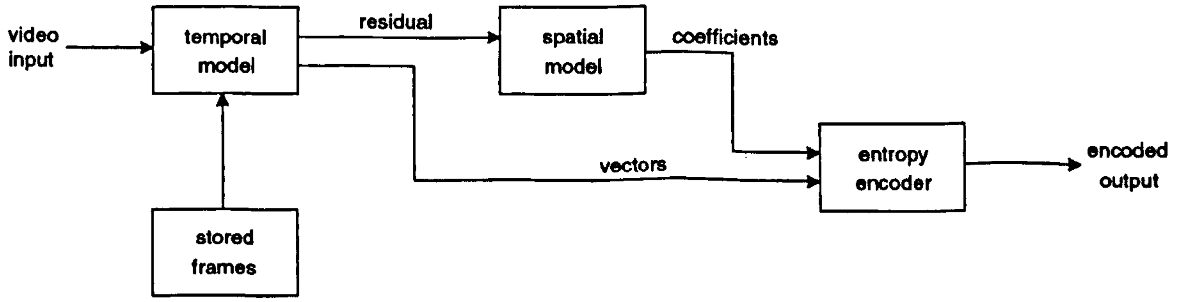


Figure 2.18: Video encoder model [61]

A video CODEC encodes a source image or video sequence into a compressed form and decodes this to produce a copy or approximation of the source sequence. If the decoded video sequence is identical to the original, then the coding process is lossless; if the decoded sequence differs from the original, the process is lousy. The CODEC represents the original video sequence by a model (an efficient coded representation that can be used to reconstruct an approximation of the video data) [61]. The model should be as close to the original video as possible whilst using the fewest bits possible. However, these two aims are often incompatible because the lower the compressed bit rate the worse the quality of the image at the end, once decoded. The decoder process is illustrated in Figure 2.19 as a block diagram:

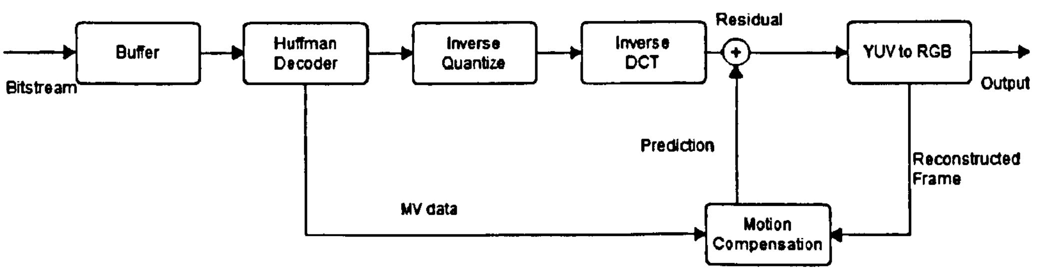


Figure 2.19: MPEG Decoder block diagram

2.2.6 Temporal Model

An uncompressed video sequence is put into the temporal model. The first frame is then fully encoded into a reference. The difference between the rest of the pictures and the reference frame is then encoded to determine the picture number, display order and picture type. The temporal model shown in Figure 2.20 is a compression method. The output result of temporal method will be used for Spatial Model as an input data. The temporal model is used to encode the first frame of a video sequence, which is uncompressed, as a reference and only encode the difference between the rest of the pictures; whilst determining the picture number, display order and picture types and adjacent the frames similarity. The temporal model attempts to reduce temporal redundancy by exploiting the similarities between neighboring video frames, usually by constructing a prediction of the current video frame [61].

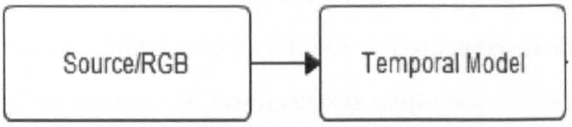


Figure 2.20: Temporal Model

The output of the temporal frame is a residual frame. This residual frame is created by subtracting the prediction from the actual frame and a set of model parameters, a set of motion vectors that describe the prediction process and how the motion was compensated.

2.2.7 Spatial Model

Spatial is picture size scalability where videos are coded at multiple spatial resolutions. Lower resolutions can predict data or samples of higher resolutions so that the bit rate can be reduced when coding higher resolutions. As illustrated in Figure 2.21 the output of the temporal model is a residual frame including a set of model parameters, typically a set of motion vectors describing how the motion was compensated.

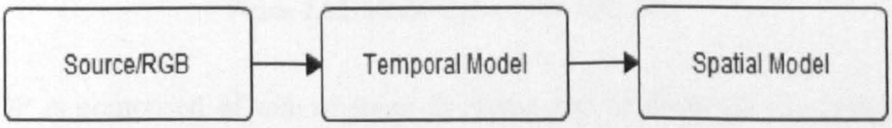


Figure 2.21: Spatial Model

The residual frame forms the input to the spatial model, which makes use of similarities between neighboring samples in the residual frame to reduce spatial redundancy [61]. The compression can be accomplished by taking advantage of the spatial and temporal redundancies inherent to video.

2.2.8 Entropy encoder

The parameters of the temporal model (typically motion vectors) and the spatial model (coefficients); are compressed by the entropy encoder. This removes statistical redundancy in the data (for example, representing commonly occurring vectors and coefficients by short binary codes) and produces a compressed bit stream or file that may be transmitted and/or stored. A compressed sequence consists of coded motion vector parameters, coded residual coefficients and header information [60].

2.2.9 MPEG Group of Pictures

Each video, in MPEG, is divided into one or more Groups of Pictures (GOP) that repeat the pattern of I, P, and B-frames for each video stream. There are three types of pictures defined in MPEG: I, P and B, pictures of which are shown in Figure 2.22:

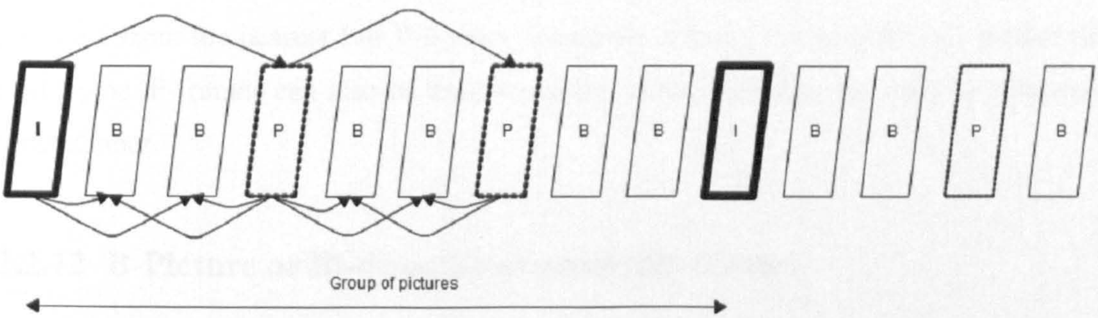


Figure 2.22: MPEG Group of Pictures [63]

Each GOP is composed of one or more pictures; one of these pictures must be an I-picture. Usually, the spacing between two anchor frames (I- or P-pictures) is referred to as M, and the spacing between two successive I-pictures is referred to as N [63], [68].

2.2.10 I-Picture or Intra frame (Key frame)

Intra frames or I pictures are the first picture. They use motion compensation and add difference data; known as intra-coding scheme and come up every ten to fifteen frames. I-frames only contain information presented within itself. The input I picture is converted from raster scan to blocks. The blocks are subjected to a DCT. The coefficients are then zigzagged, scanned, and weighted, prior to requantization (word length shortening) and subject to run-length coding. Figure 2.23shows the encoding process and the corresponding decoder.

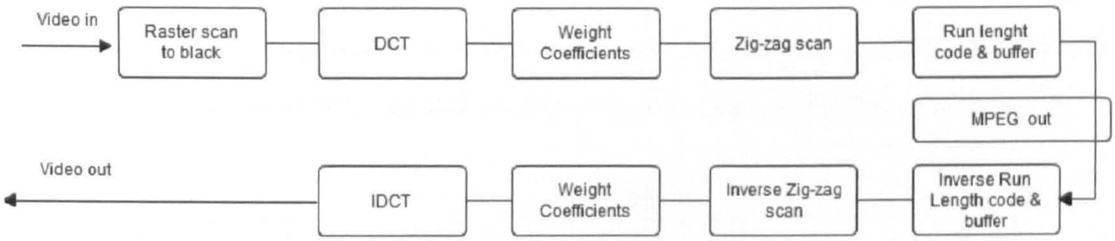


Figure 2.23: Basic encoder and decoder model

2.2.11 P-Picture or Predicted frames

P pictures or Predicted frames are decoded using the previous picture. They can be predicted from the nearest I or P-frames. Using the I-frame the encoder can predict the next frame. P-frames can also be used to predict other P-frames, but only in a forward time manner.

2.2.12 B-Picture or Bi-directional predicted frames

The last in the GOP is the Bi-directional interpolated prediction frame; the B-frame. B-frames can be decoded using vectors and prediction data from the nearest I or P-frame, before or after the B-frame [61]. The main advantage of using B-frames is coding efficiency. In most cases, B-frames will result in less bits being coded overall. Quality can also be improved in the case of moving objects that reveal hidden areas within a video sequence. Backward prediction, in this case, allows the encoder to make more intelligent decisions on how to encode the video within these areas. Also, since B-frames are not used to predict future frames, errors generated will not be propagated further within the sequence [69].

2.2.13 Macroblocks

The block layer is the lowest layer of the video sequence and consists of coded 8x8 DCT coefficients, shown in Figure 2.24. DC-coefficient is encoded when a macroblock is encoded in the Intra-mode, the. The differentials of DC values are categorized according to their absolute values and the category information is encoded using

Variable-Length Code (VLC) [63]. The VCL design follows the so-called block based hybrid video coding approach in which each coded picture is represented in block-shaped units of associated Luma and Chroma samples called Macroblocks [63],[61].

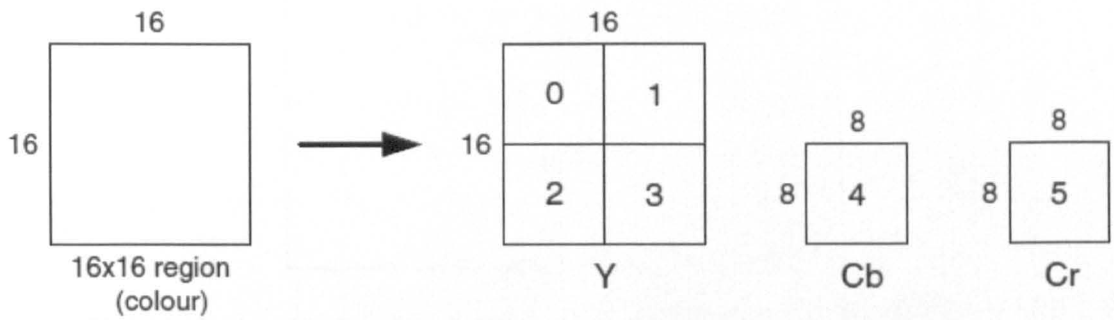


Figure 2.24: Macroblock [61]

A hybrid of inter-picture prediction that exploits temporal statistical dependencies and transforms coding of the prediction residual to take advantage of the spatial statistical dependencies is made into a basic source-coding algorithm.

2.2.14 Block Matching Algorithm

Motion estimation suggests that the objects and background of a frame in a video sequence have patterns that form corresponding objects in the next frame. The frame is divided into macro blocks; the previous frame's block are compared to the current frames in order to create vectors to specify the movement of the blocks in the frame. The movement is the motion estimate for the current frame. The search area for a good macroblock match is constrained up to 'P' pixels on all fours sides of the corresponding macroblock in previous frame. This 'P' is called the search parameter. Larger motions require a larger P, and the larger the search parameter the more computationally expensive the process of motion estimation becomes. Usually the macroblock is taken as a square of side 16 pixels, and the search parameter p is seven pixels. The matching of one macroblock with another is based on the output of a cost function [70]. This idea is represented in Figure 2.25.

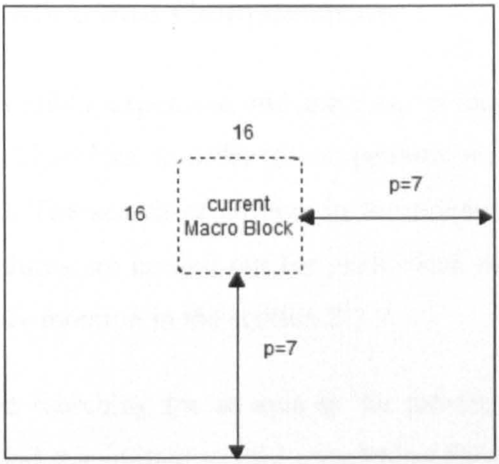


Figure 2.25: Macroblock of side 16 pixels and a search parameter p of size 7 pixels.

The macroblock that results in the least cost is the one that matches the closest to the current block. There are various cost functions, and Mean Absolute Difference (MAD) is the most popular and less computationally expensive given by Equation 2.3 [70].

$$MAD = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{i=0}^{N-1}|C_{ij} - R_{ij}| \tag{2.3}$$

Another cost function is Mean Squared Error (MSE) given by Equation 2.4 [70].

$$RMSE = \left[n^{-1}\sum_{i=1}^{n}|e_i|^2\right]^{1/2} \tag{2.4}$$

Peak-Signal-to-Noise-Ratio (PSNR) given by Equation 2.5 characterizes the motion compensated image that is created by using motion vectors and macro clocks from the reference frame [70].

$$PSNR = 10\log\left(255^2/_{\mathrm{MSE}}\right) \tag{2.5}$$

2.2.15 Motion Estimation and Compensation

Motion estimation is incredibly expensive and the most resource hungry operation in the compression process. Therefore, in order to compensate, movement of blocks in the current frame is lessened. The search of an area in the reference frame and a popular matching criterion procedures are carried out for each block of M × N samples in the current frame as previously mention in the section 2.2.9.

The first procedure is the searching for an area in the reference frame, past or future frame previously coded and transmitted to find a matching M × N -sample region. This is carried out by comparing the M × N block and finding the region that gives the 'best' match. In addition, the second procedure is to find a popular matching criterion is the energy in the residual formed by subtracting the candidate region from the current M × N block, so that the candidate region that minimizes the residual energy is chosen as the best match. This process of finding the best match is known as motion estimation [71]. Motion compensation is when a chosen candidate region is made the predictor for the current M × N block and is subtracted from the current block to make a residual M × N block. Block-based motion compensation is popular for a number of reasons. It is relatively straightforward and computationally tractable, it fits well with rectangular video frames and with block-based image transforms and it provides a reasonably effective temporal model for many video sequences [61].

The algorithms that have been implemented for motion estimate are:

- Exhaustive Search (ES)
- Three Step Search (TSS)
- New Three Step Search (NTSS)
- Simple and Efficient (SES)
- Four Step Search (4SS)
- Diamond Search (DS)
- Adaptive Rood Pattern Search (ARPS).

2.2.16 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) helps to separate the image into parts or spectral sub-bands of differing importance with respect to the image's visual quality [72]. Like other transforms, the DCT attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency [73].

The DCT operates on X, a block of N × N samples (typically image samples or residual values after prediction) and creates Y, an N × N block of coefficients. The action of the DCT (and its inverse, the IDCT) can be described in terms of a transform matrix A. The forward DCT (FDCT) of an N × N sample block is given by Equations 2.6 to 2.12 [73]:

$$Y = AXA^T \tag{2.6}$$

The inverse DCT (IDCT) is given through Equation 2.7:

$$X = A^T YA \tag{2.7}$$

Where X is a matrix of samples, Y is a matrix of coefficients and A is an N × N transform matrix. The elements of A are explained in Equation 2.8:

$$A_{ij} = C_i \cos\frac{(2j+1)i\pi}{2N} \text{ where } C_i = \sqrt{\frac{1}{N}}\ (i = 0),\ C_i = \sqrt{\frac{2}{N}}\ (i > 0) \tag{2.8}$$

The above equations are expressed in Equation 2.9 and they can be written in summation form:

$$Y_{xy} = C_x C_y \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos\frac{(2j+1)y\pi}{2N} \cos\frac{(2i+1)x\pi}{2N}$$

$$X_{ij} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_x C_y Y_{xy} \cos\frac{(2j+1)y\pi}{2N} \cos\frac{(2i+1)x\pi}{2N} \qquad (2.9)$$

Example in 2.10: N= 4 the transform matrix A for a 4 × 4 DCT is:

$$A=\begin{bmatrix} \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) \\ \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{5\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{7\pi}{8}\right) \\ \sqrt{\frac{1}{2}}\cos\left(\frac{2\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{6\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{10\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{14\pi}{8}\right) \\ \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{9\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{15\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{21\pi}{8}\right) \end{bmatrix} \qquad (2.10)$$

The cosine function is symmetrical and repeats after 2π radians and hence A can be simplified as formulated in Equation 2.11:

$$A=\begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & c \end{bmatrix} \text{ where } \begin{cases} a = \frac{1}{2} \\ b = \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) \\ c = \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) \end{cases} \qquad (2.11)$$

Evaluating the cosines gives:

$$A=\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & 0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.653 & -0.653 & 0.271 \end{bmatrix} \qquad (2.12)$$

2.2.17 Quantization

Quantization operates on the output for DCT and used to reduce the precision of image data, is the main source of loss in the compression method. A value that is turned to 0 during quantization can never be returned to its original value after dequantization. The quantizer in an image or video encoder is designed to map insignificant coefficient values to zero whilst retaining a reduced number of significant, nonzero coefficients. The output of a quantizer is typically a 'sparse' array of quantized coefficients, mainly containing zeros [61], [62]. A quantizer maps a signal with a range of values X to a quantized signal with a reduced range of values Y. It should be possible to represent the quantized signal with fewer bits than the original since the range of possible values is smaller. A scalar quantizer maps one sample of the input signal to one quantized output value and a vector quantizer maps a group of input samples (a 'vector') to a group of quantized values [60], [61].

2.2.18 Scalar Quantization

Scalar quantization represents each data value (scalar) with a reconstruction level and hence is a one-to-one mapping. It can be showed that significantly improved performance could be achieved if groups of the input values or vectors of arbitrarily large dimension are coded. This many-to-one mapping is termed vector quantization [76], [77]. The equation of the quantization can be written as 2.13 [76], [77]:

$$QO = round\left(\frac{X}{QS}\right) \tag{2.13}$$

In this equation, X is the input value, QS is the "quantizer side step" which controls the output (QO).

2.2.19 Dequantization

Dequantization, which is inverse quantization, is written as 2.14 [61], [77]:

$$Y = QO \times QS \tag{2.14}$$

Again QO is the quantized value, QS is the step size and Y is the dequantized value. Table 2.9 is representing the example parameters.

$$\text{Example } Y = QS.round\left(\frac{X}{QS}\right) \tag{2.15}$$

	Y			
X	QS = 1	QS = 2	QS = 3	QS = 4
- 4	- 4	- 4	- 3	- 5
- 3	- 3	- 2	- 3	- 5
- 2	- 2	- 2	- 3	0
- 1	- 1	0	0	0
0	0	0	0	0
1	1	0	0	0
2	2	2	3	0
3	3	2	3	5
4	4	4	3	5
4	4	4	3	5
5	5	4	6	5
6	6	6	6	5
7	7	6	6	5
8	8	8	9	10
9	9	8	9	10
10	10	10	9	10
11	11	10	12	10

Table 2.9: Scalar Quantizers [61]

Figure 2.26 shows two examples of scalar quantizers, a linear quantizer (with a linear mapping between input and output values) and a nonlinear quantizer that has a 'dead zone' about zero (in which small-valued inputs are mapped to zero) [61], [63].

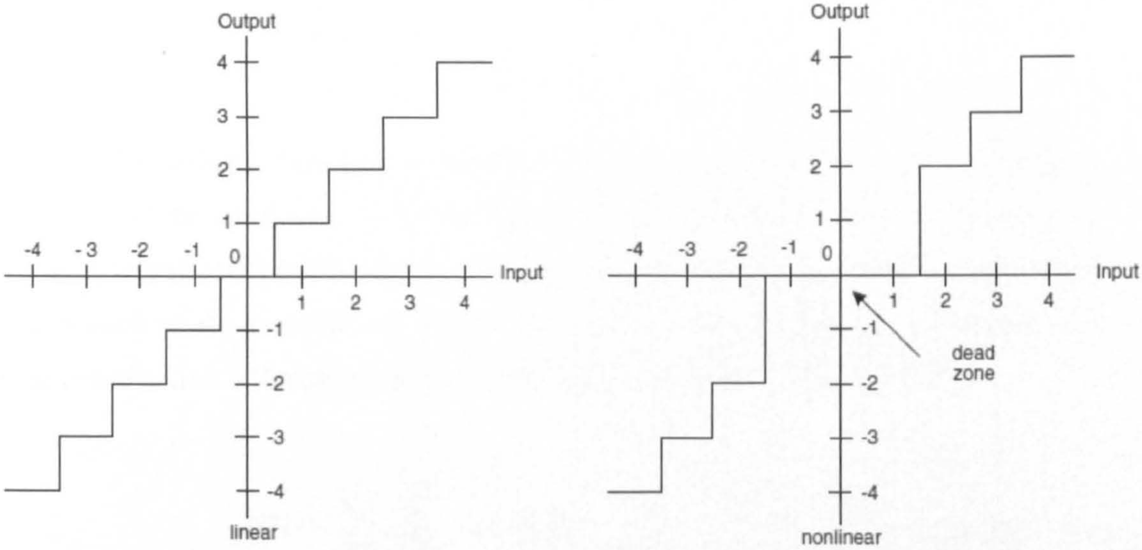


Figure 2.26: Scalar quantizers: linear; nonlinear with dead zone [61]

The quantization operation is made up of a Forward Quantizer (FQ) in the encoder and an Inverse Quantizer (IQ) in the decoder. The step size (QS) is crucial in the operation. The larger step size means smaller quantized values, which are highly compressed during transmission, but this means the rescaled values will only be a crude approximation of their original value. The smaller the step size the closer the value is to its original value once dequantized. However, the larger step size results in a larger range of quantized values, which reduces compression efficiency [61], [63], [76].

2.2.20 Discrete Wavelet Transform

A wavelet, in the sense of the Discrete Wavelet Transform (or DWT), is an orthogonal function which can be applied to a finite group of data [79]. Wavelet is based on the Fourier expansion; the idea that a signal can be expressed as a series of sinus and cosines. Wavelet transform explained as Equation 2.16 [79]:

$$X_w(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} x(t)\,\Psi_{a,b}(t)dt \tag{2.16}$$

where a mother wavelet Ψ is dilated to the scale parameter (a) and translated at the position parameter (b) to form the basis function Ψ defined by Equation 2.17:

$$\Psi_{a,b}(t) = \Psi\left(\frac{t-b}{a}\right) \tag{2.17}$$

If a one-dimensional function is transformed through a wavelet, a two-dimension function will be produced. Any number dimensional function that is transformed through a wavelet will double. For images, the discrete wavelet transform is used basing the approach on the fact that any square integral function $x(t)$ can be represented as a linear combination of functions as explain in Equations 2.18 and 2.19 [81]:

$$x(t) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \alpha_{mn} \Psi_{m,n}(t) \tag{2.18}$$

where α_{mn} are the wavelet transform coefficients:

$$\alpha_{mn} = \int_{-\infty}^{+\infty} x(t)\, \Psi_{m,n}(t) dt \tag{2.19}$$

The wavelet transform is a set of filters with coefficients equal to discrete wavelet functions [79], [80].

The different type of wavelets functions are, Haar wavelet, Daubchies wavelets, Shannon wavelet and Mexican hat wavelet.

2.2.21 Entropy Coding

The entropy encoder converts a series of symbols representing elements of the video sequence into a compressed bitstream suitable for transmission or storage [61]. Entropy encoding encodes data depending on the information content, possibly including motion vectors and quantized transform coefficients. The length of each code depends of the probability of occurring, the higher the probability the shorter the code and vice versa. The Huffman coding and Arithmetic coding are the most commonly used VLC algorithms.

2.2.22 Zigzag scanning and Huffman coding

Zigzag scanning is commonly used to reorder coefficients of a block. The scanning pattern tends to achieve long runs of consecutive zero coefficients. Figure 2.27 shows the zigzag scanning pattern. The serialization and coding of the quantized DCT coefficients exploits the likely clustering of energy into the low-frequency coefficients and the frequent occurrence of zero-value coefficients. The block is scanned in a diagonal zigzag pattern starting at the DC coefficient to produce a list of quantized coefficient values, ordered according to the scan pattern [75].

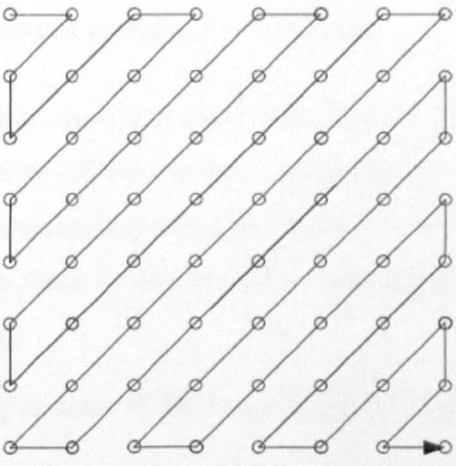


Figure 2.27: Order in which the coefficients are scanned for reordering in the zigzag [61]

The Huffman coding algorithm is used for lossless data compression with the aim of creating the shortest possible word code length. Every symbol has a variable length coding depending on its probability that run length of zeros followed by a non-zero level [60], [61], [77]. The relationship and bases of compression is illustrated in Figure 2.28:

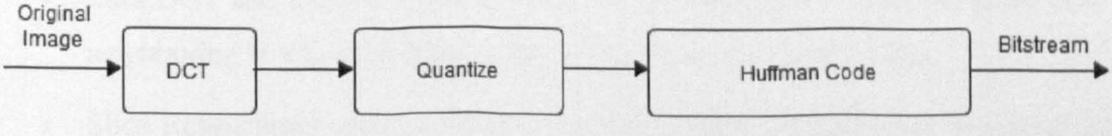


Figure 2.28: Basic compression process

2.2.23 MPEG-4 Video Coding

MPEG-4 adds new functionalities, which enable images to be compressed but still maintain high quality once decoded, additionally the same DCT and motion compensation techniques as MPEG-1 have been used with MPEG-4's video coding standard. However, MPEG-4 has new levels of interaction with video content or objects and network communications whilst using low bitrates and bandwidth. Furthermore, broad range of applications can make use of MPEG-4 technology to transport rectangular video frames and in most cases, the MPEG-4 transport happens over IP based network. In theory, MPEG-4 is designed primarily to support efficient, robust coding [82]. The main features included the following:

- Half pixel motion compensation, this feature significantly improves the prediction capability of the motion compensation algorithm by reducing the roughness in measuring best matching blocks with coarse time quantization and identifying where there is object motion that needs fine spatial resolution for accurate modeling.
- Arithmetic coding in place of the variable length (Huffman) coding.
- Advanced motion prediction mode including overlapped block motion compensation.
- A mode that combines a bi-directionally predicted picture with a normal forward predicted picture. In addition, H.263 supports a wider range of picture formats including 4CIF (704 × 576 pixels) and 16CIF (1,408 × 1,152 pixels) to provide a high-resolution mode picture capability.
- Supports prediction for objects when they partially move outside of the boundaries of the frame.
- Intra DCT and DC/AC Prediction that set the coefficients to be predicted from neighboring blocks whichever to the left or above the current block.
- Slice Resynchronization, quicker resynchronization after an error has occurred by creating slices within an image. The standard removes data dependencies

between slices to allow error-free decoding at the start of the slice regardless of the error that occurred in the previous slice [62].

- Reversible VLC code tables are designed to add to support recovery for packet loss. When an error is encountered, it is possible to synchronize at the next slice or start code and work back to the point where the error occurred [62].

2.2.24 System

MPEG-4 streams data in a very complex way, allowing all data to be streamed independently, it compresses all video and audio data so that it can be rebuilt at the other end. The systems part of the MPEG-4 addresses the description of the relationship between the audio-visual components that constitute a scene. The relationship is described at two main levels [82]:

- The arrangements of the objects in a scene can be described by a spatiotemporal sequence that is called Binary Format for Scenes (BIFS). BIFS is the compressed format in which the users can interact with the content by modifying the order of the objects in the scene [83].
- Elementary streams, is another level of the relationship that introduces the conceptual delivery pipes of MPEG-4. It is mapped to actual delivery channels by identifying the characterized of object descriptors. This includes the scene description, audio and visual objects data, as well as Object Descriptors themselves [84].

The Hybrid coding has been used at the MPEG-4 Visual standard. This coding can mix natural images and video coding with computer-generated scenes. This is a good trade-off between exploitation of spatial (intra-frame) and temporal (inter-frame) redundancies, which can be obtained by application of hybrid coding techniques [84].

Figure 2.29 shows a block diagram of a hybrid encoder. The image is divided into blocks each of which have a motion vector that describes the shift of a reconstructed block from a previously decoded image. This information is used to predict the next block. The difference between the actual block values and the prediction values is calculated, and a DCT is applied to this prediction error signal [84].

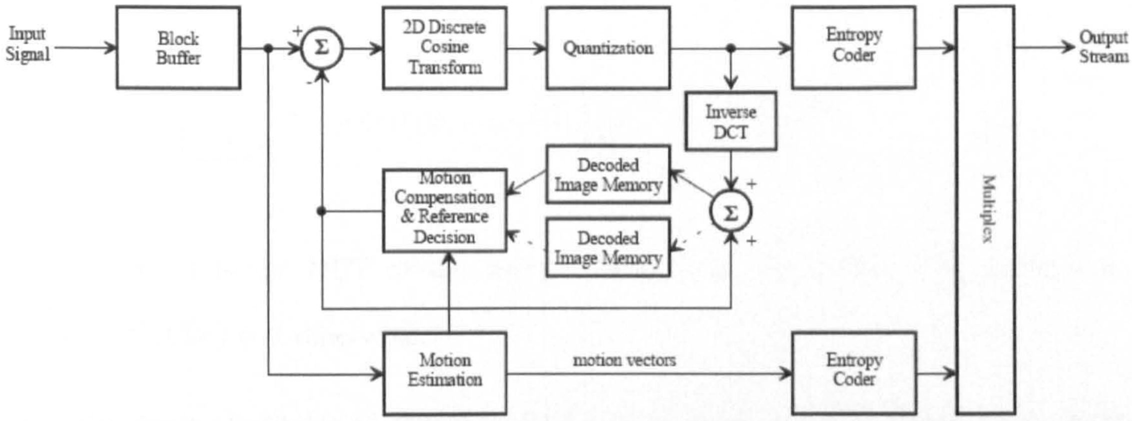


Figure 2.29: Structure of a hybrid video encoder according to an MPEG standard [84]

A smaller number of transform coefficients is a good representation of all the pixels within a block because of the de-correlation property of the transform. Using a combination of run-length and variable-length entropy encoding the dominant coefficients are quantized and encoded. Since only blocks from frames already transmitted are used for prediction, the inverse operation is possible at the decoder, such that a reconstruction can only be performed from the output stream [84].

2.2.25 Two Dimensional DCT/IDCT in MPEG-4

The 2-D DCT block calculates the two-dimensional discrete cosine transform of the input signal. The equation for the two-dimensional DCT is formulated in Equation 2.20 [74]:

$$F(m,n) = \frac{2}{\sqrt{MN}} C(m)C(n) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) cos \frac{(2x+1)m\pi}{2M} cos \frac{(2y+1)n\pi}{2N} \tag{2.20}$$

where $(m), C(n) = \frac{1}{\sqrt{2}}$, for $m, n = 0$ and $C(m), C(n) = 1$ otherwise.

The number of rows and columns of the input signal must be powers of two. The output of this block has the same dimensions as the input.

The 2-D IDCT block calculates the two-dimensional inverse discrete cosine transform of the input signal. The equation for the two-dimensional IDCT is formulated in Equation 2.21 [74]:

$$f(x,y) = \frac{2}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} C(m)C(n)F(m,n) \cos\frac{(2x+1)m\pi}{2M} \cos\frac{(2y+1)n\pi}{2N} \qquad (2.21)$$

where $F(m,n)$ is the DCT of the signal $f(x,y)$ and $C(m), C(n) = \frac{1}{\sqrt{2}}$ for $m, n = 0$ and $C(m), C(n) = 1$ otherwise.

DCT changes a frame f(x,y) into frequencies (u,v). If an image is processed by 8x8 blocks one at a time then the correlation between pixels from one 8x8 block to the next will be scattered but follow a straight line. This is because pixels near each other are only slightly different. The equation of the new system is explained in Equation 2.22 and can be written as [61], [74]:

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) cos\frac{(2x+1)u\pi}{16} cos\frac{(2y+1)v\pi}{16}$$

$$C(n) = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ 1 & n \neq 0 \end{cases} \qquad (2.22)$$

2.2.26 MPEG-4 GOP

The GOP length and structure in MPEG-4 or Group of Video (GOV) can be adjusted. Normally, it is a fixed repetitive pattern, for example:

GOV = 4, (IPPP IPPP ...)

GOV = 15, (IPPPPPPPPPPPPPP IPPPPPPPPPPPPPP ...)

GOV = 8, (IBPBPBPB IBPBPBPB ...)

The suitability of the GOP depends on the application. Less I-frames decreases the bit rate whilst fewer B-frames reduce the latency. Figure 2.30 shows a GOP where the second I-frame marks the start of the next GOP. The arrows indicate the frame dependency relationships. These demonstrate that the I-frame contains the most information, P-frames contain 50% of that information and B-frames contain only 25% of the information. Therefore, B-frames are only used between the current frames or before and after P reference frames, or I. P-frames are the second most important frames, but if an earlier reference frame is removed because of editing or dropping during transmission, a P-frame cannot be decoded and the GOP will not be processed or completed. The increased I-frames means less P and B-frames which would bring higher compression rates.

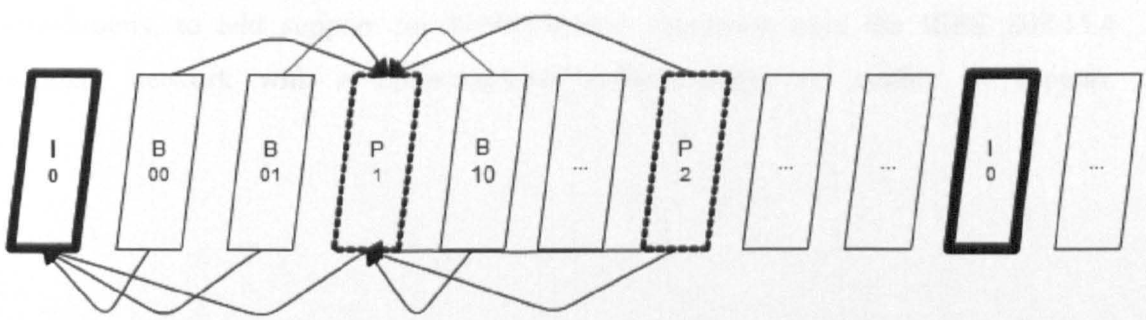


Figure 2.30: MPEG Group of Pictures Example

Let N_P represent the number of P-frames in a GOP and N_Bp represent the number of B-frames in between an I-frame and a P-frame or two P-frames. Using these two terms, a specific GOP pattern can be indented uniquely by G (NP, NBP). For example, GOP (2, 2) signifies the GOP pattern `IBBPBBPBB'. Let NB represent the number of B-frames in a GOP and NG represent the length of the GOP. Then $N_B = \left(1 + N_{Bps}\right) \times N_{BP}$ NBP and NG = 1 + NP + NB. As in Figure 2.30 the single I-frame of a GOP is referred to as I0, while P-frames are indexed as P_i, where $1 \leq i \leq$NP, and B-frames are strained data rate using media scaling [68],[86].

2.3 Summary

In this chapter, an overview of the IEEE 802.15.4 standard and MPEG video compression was introduced. IEEE 802.15.4 is considered the next generation of wireless communication systems and has been explained and compared to other existing wireless standards. The video coding tools, requirements and the technologies of the MPEG-1, MPEG-2 and MPEG-4 are described in this chapter; such as, motion compensated prediction, transform coding, quantization and entropy coding.

Given the discussed limitations in this chapter, this research will carry out experiments, which are motivated to find a novel streaming algorithm for ZigBee's open application technology and combining that with an intelligent system approach in MPEG-4 applications, to add support for MPEG-4video streaming over the IEEE 802.15.4 standard network with a optimum-level maintainability of quality of support.

Chapter 3: Artificial Intelligence and Mathematics

In the 1940s and 50s, a handful of scientists from a variety of fields (including mathematics, psychology, engineering, economics and political science) began to discuss the possibility of creating an artificial brain. The field of AI research was founded as an academic discipline in 1956 [88]. AI can be defined as a branch or computer and science that is interested in the mechanization of intelligent behavior principles. However, this definition suffers from the fact that intelligence itself is not very well defined or understood. "Although most of us are certain that we know intelligent behavior when we see it, it is doubtful that anyone could come close to defining intelligence in a way that would be specific enough to help in the evaluation of a supposedly intelligent computer program, while still capturing the vitality and complexity of the human mind" [89], [90].

3.1 Adaptive Systems

Adaptive systems are highly social and word adaptive implies that the organization and its subcomponents are capable of studying and analyzing the environment, taking semi-autonomous actions that internally adjust the organization and externally influence the environment in a manner that allows the organization to fulfill local and higher-level goals, while concomitantly adapting to environmental shifts and perturbations [90]. The adaptive systems often compete with one another and often join forces in cooperative communities. Adaptive systems provide a methodology to help solve the problem. Adaptive systems can be systematically investigated resulting in adaptive plans or strategies that can provide the basis for new and interesting algorithms. This social structure is perhaps the most important and difficult problem in understanding the natural history of adaptive systems [91]. In the 1970s, Holland emphasized that these systems should be able to handle uncertain and changing environments, and that the systems, via feedback from the environments in which they operate, should be able to self-adapt over time [92].

Holland proposed formalism in his seminal work on adaptive systems that provides a general manner in which to define an adaptive system, all of which have some basic properties, common to many complex adaptive systems [94]. Examples of adaptive systems are swarm intelligence, cities, the brain, the immune system, ecosystems, and computer models.

3.2 Swarm Intelligence

Swarm Intelligence (SI) is an AI technique, is concerned with the design of intelligent multi-agent systems taking inspiration from the collective and social behavior of schools of fish, flocks of birds and colonies of ants. This type of intelligence must be decentralized, self-organized and distributed throughout an environment and they are commonly used in nature to solve problems, including; colony relocation or foraging for food [93]. The SI problem solving has capabilities, formed by a quantity of relatively independent and very simple subsystems which do not show individual intelligence [95]. A swarm is able to solve problems that a single individual could not, hence individuals are candidate solutions. Individuals in a swarm only have local sensory information, have little or no memory and can only perform simple actions. SI problem solving techniques present several advantages over more traditional ones [93]. On one hand, they are cheap, simple and robust; on the other hand, they provide a basis with which it is possible to explore collective or distributed problem solving without centralized control or the provision of a global model [94]. However, even if some of the individuals of a swarm are lost or make mistakes the performance of the swarm as a whole will not be affected [95], [96]. In 1989, Beni and Wang described SI as systems of non-intelligent robots exhibiting collectively intelligent behavior evident in the ability to unpredictably produce 'specific' (i.e. not in a statistical sense) ordered patterns of matter in the external environment. Basically SI systems are "unpredictable" for their definition of unpredictable (which is thorough) and they produce results that are "improbable" so are in some way surprising or unexpected [97]. Two methods purpose and used by researchers in and they are Ant Colony and Particle Swarm Optimization.

3.3 The Ant Colony Optimization

Ant Colony Optimization (ACO), has been found to be both robust and versatile in handling a wide range of combinatorial optimization problems [107]. Ant Colony Optimization the first field investigates probabilistic algorithms inspired by the stigmergy and foraging behavior of ants. The ACO offers discovery of good solutions and it can be used in dynamic applications.

The basic idea of a real ant system is illustrated in Figure 3.1. Ants arrive at a decision point in which they have to decide whether to turn left or right. Since they have no idea about which is the best choice, they choose randomly. It can be expected that, on average, half of the ants decide to turn left and the other half to turn right. This happens both to ants moving from left to right and to those moving from right to left [108].

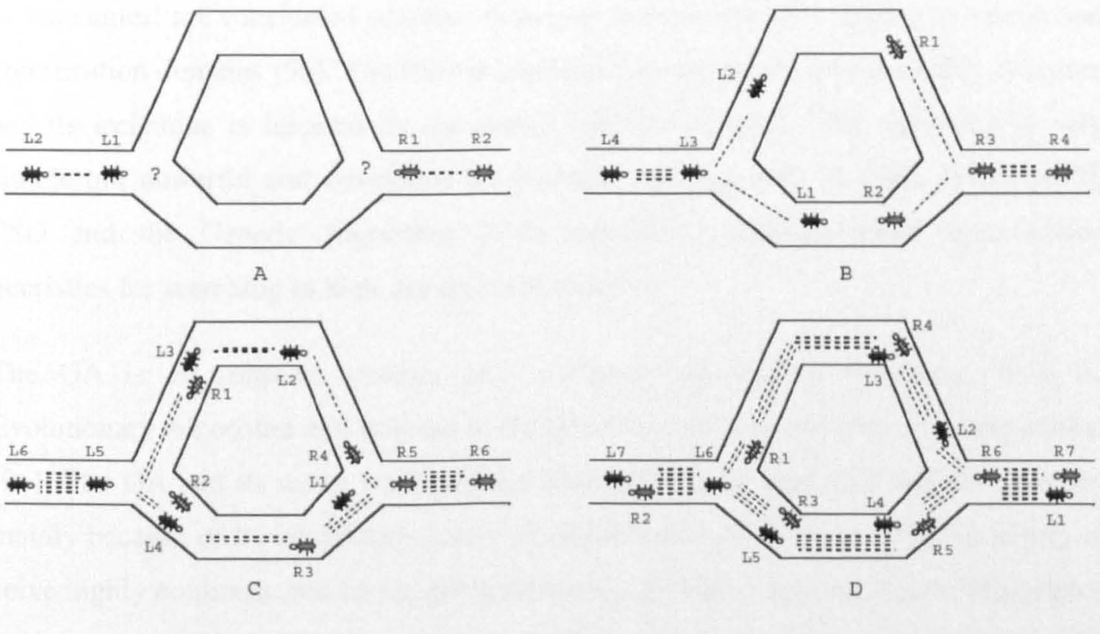


Figure 3.1: Illustrating the behavior of real ant movements [108]

Figure 3.1 shows what happens immediately following this. Since the lower path is shorter than the upper one, more ants will visit it on average, and therefore pheromone accumulates faster. After a short transitory period, the difference in the amount of pheromone on the two paths is sufficiently large so as to influence the decision of new ants coming into the system. From now on, new ants will prefer to choose the lower path, since at the decision point they perceive a greater amount of pheromone on the lower path. This in turn increases, with a positive feedback effect, the number of ants choosing the lower, and shorter, path. Very soon all ants will be using the shorter path [108].

3.4 Background of Particle Swarm Optimization

Optimization is another noble candidate at solving problems in science and engineering. In numerical optimization, a guess of what the optimum could be is made and then continually refined until the criterion is met. PSO is a population based stochastic and it is the second field that investigates probabilistic algorithms inspired by flocking, schooling and herding. Like evolutionary computation, swarm intelligence 'algorithms' or 'strategies' are considered adaptive strategies and are typically applied to search and optimization domains [96]. The PSO is applicable for problems that are fuzzy in nature and its technique is inspired by the social behavior of birds. The algorithm is very simple but powerful and developed by Eberhart and Kennedy in 1995, [101], [102]. PSO and the Genetic Algorithm (GA) represent population-based optimization heuristics for searching in high-dimensional spaces.

The GA is an adaptive strategy and a global optimization technique. It is an Evolutionary Algorithm and belongs to the broader study of Evolutionary Computation [96]. The GA and its many versions have been popular in academia and this industry mainly because of its intuitiveness, ease of implementation, and the effective ability to solve highly nonlinear, mixed integer optimization problems that are typical of complex engineering systems. The drawback of the GA is its expensive computational cost [103]. PSO is similar to the GA with much lower computational costs.

The idea of PSO is that populations of potential solutions are intended to move collectively through a problem search space, under their respective algorithmic strategies, towards 'fitter' regions (represented by better solutions) and ideally to a solution representing the global optimum [103]. PSO adapts behavior and looks for the best solution-vector in the search space. A solution is called a particle; each particle has a velocity that directs the "flying" of a particle as well as a cost value and fitness that is evaluated and minimized by the function. The particle searches for solutions to a continuous optimization problem by flying through the search space following optimum particles. The velocity of the particle is decided according to its flying experience and the experience from its neighboring particles. Specifically, the behavior of each particle is affected by either the local best or the global best particle to help it fly through hyperspace. Therefore, by observing the behavior of the flock and memorizing their flying histories, all particles in the swarm can quickly converge to near-optimal geographical positions [104]. PSO have no overlapping and mutation calculation. The search can be carried out by the speed of the particle [105]. The calculation in PSO compare to GA is simpler and it occupies the bigger optimization ability and it can be completed effortlessly [105], [106].

3.5 PSO Natural Model

To understand the PSO it is best to look at the coordination and collective behavior of animals in nature. For example, if a fish schools are searching for food in a defined area in which there is food to be found, the fish will not know where the food is but they know how far away the food is each time. Each fish therefore, will follow the fish that is nearest the food. Many biological creatures such as fish schools and bird flocks clearly display structural order, with the behavior of the organisms so integrated that even though they may change shape and direction, they appear to move as a single coherent entity. The main properties of the collective behavior can be pointed out as follows and is summarized by these principles [107]:

- Homology
- Locality
- Collision Avoidance
- Velocity Matching
- Flock Cantering

3.5.1 Homogeneity

Every bird in the flock has the same behavioral model. The flock moves without a leader, even though temporary leaders seem to appear [107]. Figure 3.2 illustrates the Homogeneity model.

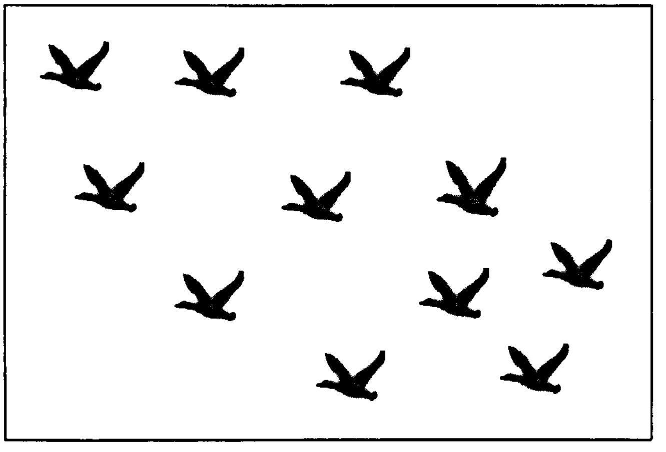


Figure 3.2: Homogeneity

3.5.2 Locality

Its nearest flock mates only influence the motion of each bird. Vision is considered to be the most important sense for flock organization [107]. Figure 3.3 illustrates the Locality model.

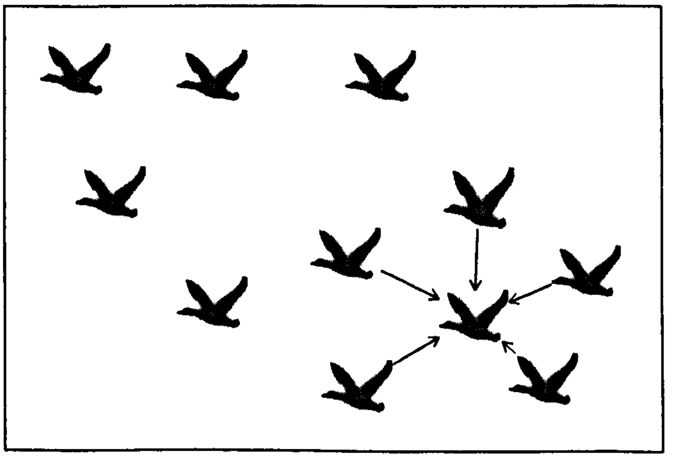


Figure 3.3: Locality

3.5.3 Collision Avoidance

Birds avoid colliding with their nearby flock mates [107]. Figure 3.4 illustrates the Collision Avoidance model.

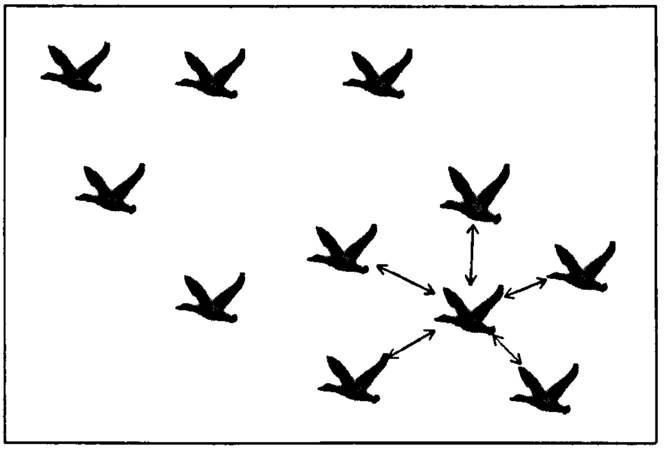


Figure 3.4: Collision Avoidance

3.5.4 Velocity Matching

Birds attempt to match velocity with their nearby flock mates [107]. Figure 3.5 illustrates the Velocity Matching model.

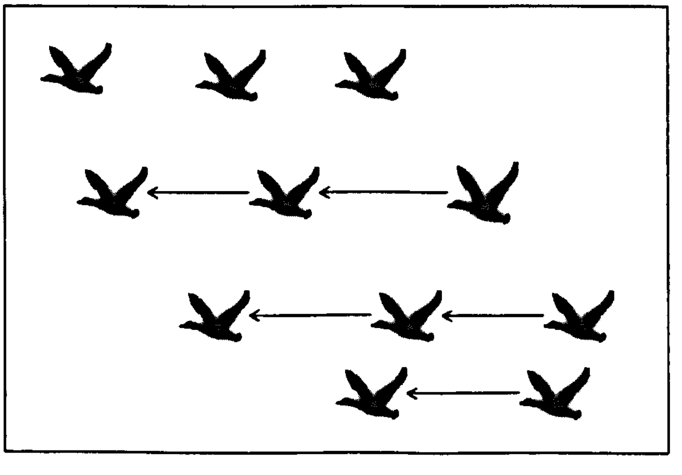


Figure 3.5: Velocity Matching

3.5.5 Flock Cantering

Birds attempt to stay close to their nearby flock mates. The individual birds attempt to maintain a minimum distance between themselves and others at all times [107]. Figure 3.6 illustrates the Flock Cantering model.

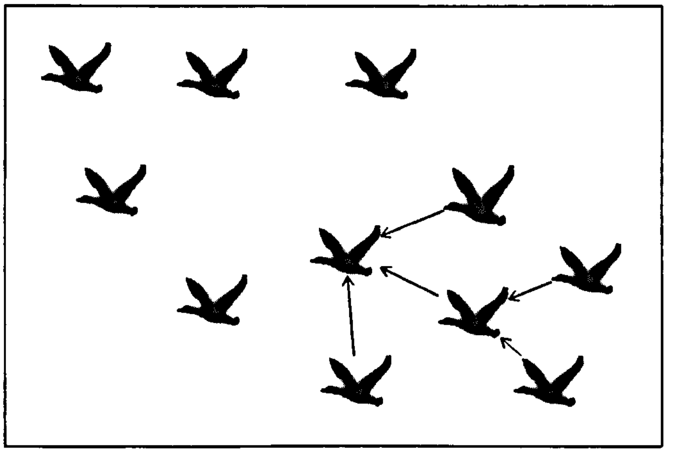


Figure 3.6: Flock Cantering

This rule is given the highest priority and corresponds to a frequently observed behavior of animals in nature. If individuals are not performing an avoidance maneuver they tend to be attracted towards other individuals to avoid being isolated and to align themselves with neighbors.

3.6 PSO Topologies

Particle topologies or neighborhoods refer to the grouping of particles into subgroups. A particular particle can communicate and exchange information about the search space only with other particles in its neighborhood. The performance of the PSO algorithm depends to some extent on the neighborhood topology, as discussed below [109], [110].

3.6.1 Star Topologies:

The star topology shown in Figure 3.7 has only one neighborhood and each particle has a link to every other particle. PSO algorithms using this topology are called "global best" or "gbest" algorithms [109].

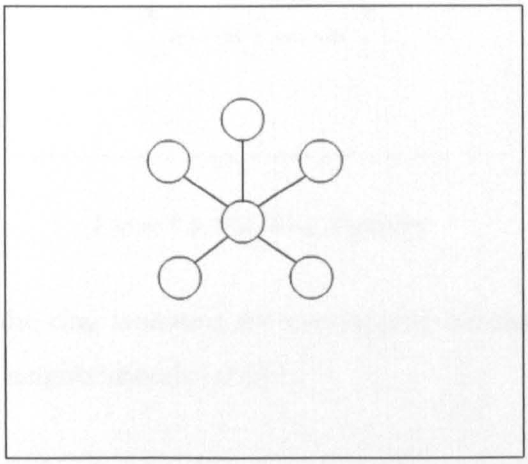


Figure 3.7: PSO Star Topologies

The star topology effectively isolates individuals from each other, since information has to be communicated through the central node. This central node compares the performance of every individual in the population and adjusts its own trajectory towards the best of them [111], [112].

3.6.2 Ring Topologies:

In the ring topology illustrated in Figure 3.8, the neighbors are closely connected and thus, they react when one particle has a raise an increase in its fitness, this reaction dilutes proportionally with respect to the distance [112].

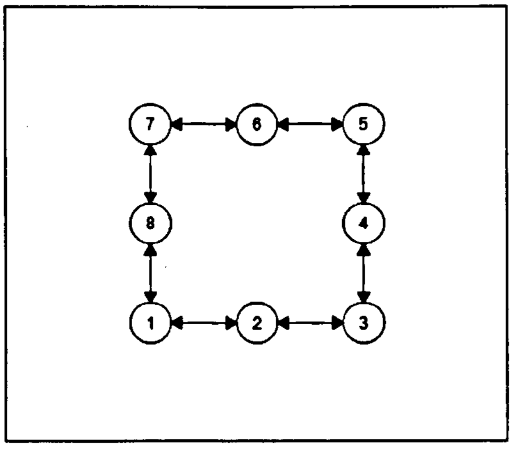


Figure 3.8: PSO Ring Topology

The neighborhoods in the ring structure are overlapping because each particle resides simultaneously in three neighborhoods [109].

3.6.3 Cluster Topologies:

In cluster topologies shown in Figure 3.9, the eight particles are placed in two neighborhoods, each containing four particles. PSO algorithms using the ring and cluster neighborhood topologies are called "local best" or "lbest" algorithms [109], [111].

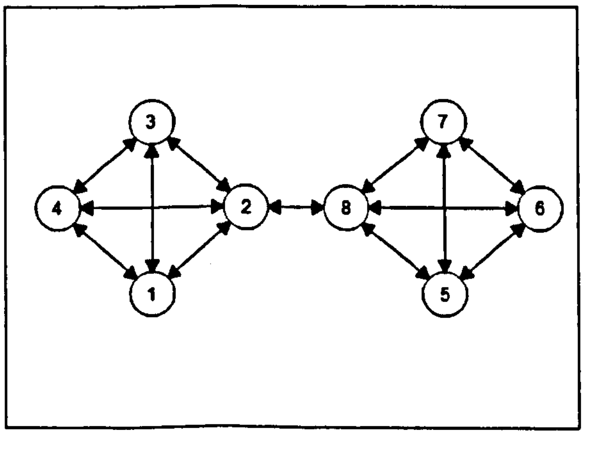


Figure 3.9: PSO Cluster Topology

By defining the particles' neighborhood based on their position in the search space, one can conceive a neighborhood for a particle by considering other particles that are moving to different regions of the search space. That is, particles that are spatially close – but only temporarily. In contrast, considering in the same cluster, only particles that are moving to similar regions on the search space seems much more appropriate than defining the neighborhood based simplistically on the particles' position [60].

3.6.4 Fully Connected Topologies:

Fully connected topology is also known as the full topology shown in Figure 3.10 where all nodes in this topology are directly connected to each other [109].

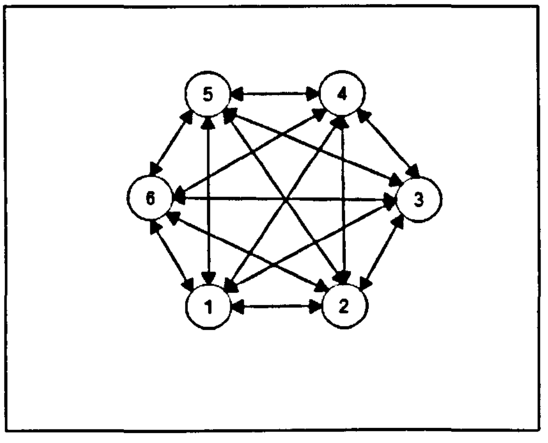


Figure 3.10: PSO Fully Connected Topologies

Kennedy et al. suggested (Kennedy and Eberhart, 2001; Kennedy and Mendes, 2002) that "gbest" populations tend to converge more rapidly to an optimum than "lbest" populations, but also, that they are more susceptible to converge to a local optima. However, this topology is the most commonly used [109].

3.6.5 Mesh Topologies:

In this type of topology, shown in Figure 3.11, one node is connected to several nodes; most commonly each node is connected to four neighbors. In the mesh topology, the particles in the corners are connected to its two adjacent neighbors [112].

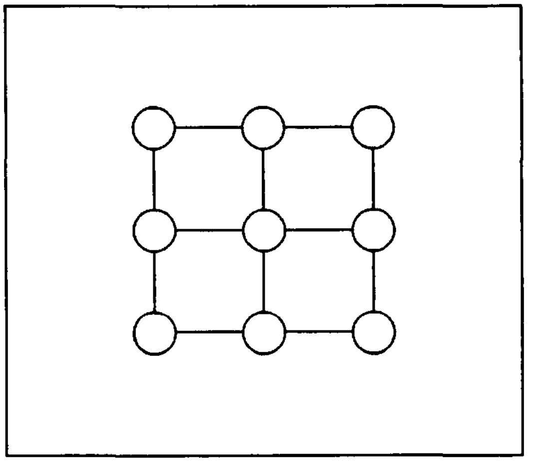


Figure 3.11: PSO Mesh Topologies

The particles on the mesh's boundaries will have three adjacent neighbors and the particles on the mesh's center will have four adjacent neighbors.[112].

3.6.6 Toroidal Topology:

Topologically, a torus is a closed surface defined as the product of two circles. This topological torus is often called a Clifford torus. The toroidal topology is similar to the mesh topology, except that all particles in the swarm have four adjacent neighbors [112]. As is shown in Figure 3.12 the toroidal topology connects every corner particle with its symmetrical neighbor. The same occurs with the toroid boundaries. The assignment from particles to nodes will be similar to the mesh topology assignment [112].

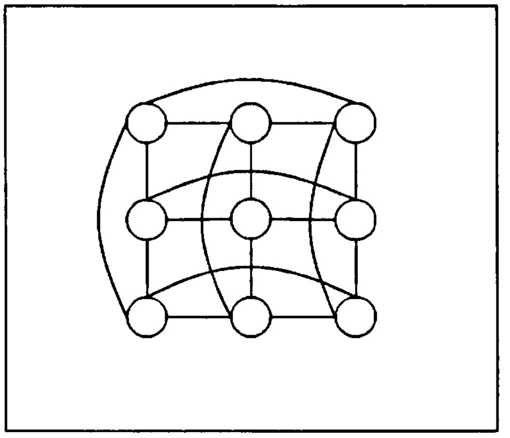


Figure 3.12: PSO Toroidal Topology

3.7 PSO Artificial Model

Particles in PSO are given random velocities and positions in order to solve real-valued and single-objective optimization problems. It works by having a swarm of candidate solutions called particles, each having a velocity that is updated recurrently and added to the particle's current position to move it to a new position [100],[103]. The PSO algorithm updates the velocity and position of each particle in a swarm by learning from the successful experience of other particles. PSO is a clustering algorithm in the areas of mutliobjective, dynamic optimization and constraint handling [105]. The basic model of PSO is shown in Figure 3.13.

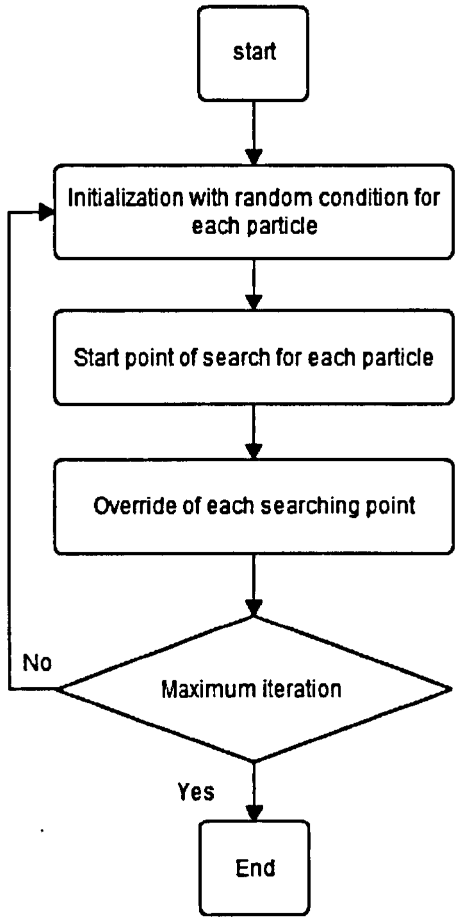


Figure 3.13: Particle swarm optimization model

In PSO, particles move over a specified D-dimensional search space at different random or heuristically velocities and positions. The algorithm updates the velocity and position of each particle in the swarm by learning from its neighboring particles. Its own fitness is then evaluated and a good experience is reached. The basic particle swarm model can be explained in Equation 3.1 [114]. In a D-dimensional search space, the position vector of the i-th particle is given by $X_i = \left(x_{i,1}, x_{i,2}, x_{i,D}\right)$ and the velocity of the i-th particle is given by $V_i = \left(v_{i,1}, v_{i,2}, v_{i,D}\right)$. Positions and velocities are adjusted and the objective function is to optimize i.e. $f(x_i)$ is evaluated with the new positional coordinates at each time-step. The velocity and position update equations for the d-th dimension of the i-th particle in the swarm may be represented as explained in Equation 3.1 [106], [114]:

$$v_{i,d,t} = w \times v_{i,d,t-1} + C_1 \times rand_1 \times (p^{t}_{i,d,t-1} - x_{i,d,t-1}) + C_2 \times rand_2 \times (p^{g}_{i,d,t-1} - x_{i,d,t-1})$$

$$x_{i,d,t} = x_{i,d,t-1} + v_{i,d,t} \qquad (3.1)$$

where $rand_1$ and $rand_2$ are random positive numbers uniformly distributed in (0, 1) and are drawn anew for each dimension of each particle.

P^l_i is the personal best solution found so far by an individual particle while P^g_i, i represents the best particle in a neighborhood of the its particle for the lbest PSO model. Note that in PSO, a neighborhood is defined for each individual particle as the subset of particles, which it is able to communicate with. The gbest for PSO may be regarded as a special case of the lbest model where the entire swarm acts as the neighborhood of any particle and P^g_i, i simply becomes the globally best position found so far by all the particles in the population. In lbest PSO, if at any iteration a particle is the best in its neighborhood, then the velocity update formula as presented in Equation 3.2 for this particle will be [114]:

$$v_{i,d,t} = w \times v_{i,d,t-1} + C_1 \times rand_1 \times (P^l_{i,d} - x_{i,d,t-1}) \qquad (3.2)$$

The variables in the PSO system of equations are summarized in Table 3.1

V_i	The particle velocity.
X_i	The particle position (test solution).
t	Time
P^l_i	The particle's position (previous) that resulted in the best fitness so far.
P^g_i	The neighborhood position that resulted in the best fitness so far.
d	D-dimensional search space

Table 3.1: The PSO Variables

3.7.1 Global Best (gBest)

In Star and Cluster topologies, every particle aims to improve its position by comparing positions with any particle it encounters during the exploration process. The best position of the entire process is called the "gbest". This changes as each particle's "gbest" reference value is stored during the process. This reference is used to update the position and velocity of each of the particles in the population. At any instant, the particle changes its velocity by comparing itself with that of the 'gbest' particle. Hence, the 'gbest' particle helps to update the position of each of the particles in the whole search process [113]. gBest can be formulated and explained as shown in Equations 3.3 and 3.4 [113]:

Velocity update per dimension:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 c_{2j}(t)\left[\hat{y}_j(t) - x_{ij}(t)\right]$$

$$v_{ij}(0) = 0\ (usually)$$

$c_1 c_2$ are positive acceleration coefficients:

$$r_{1j}(t), r_{2j}(t) \sim U(0,1)$$

$y(t)$ is the personal best position calculated as:

$$y_i(t+1) = \begin{cases} y_i(t) & if\ f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & if\ f(x_i(t+1)) < f(y_i(t)) \end{cases}$$

$\hat{y}(t)$ is the global best position calculated as:

$$\hat{y}(t) \in \{y_0(t), y_{ns}(t) | f(\hat{y}(t))\}$$

$$= \min\{f(y_0(t)), f(y_{ns}(t))\} \tag{3.3}$$

or

$$\hat{y}(t) = min\{f(x_0(t)), f(x_{ns}(t))\} \tag{3.4}$$

where n_s is the number of particles in the swarm.

3.7.2 Local best (lbest)

PSO in other topologies uses a Local Best. The Local Best, or "lbest", is where every particle is stochastically attracted to the particle in its topological neighborhood that has found the best solution. This is dependent on the particular neighborhood topology that is used for selecting the global best for each particle. Local best can be formulated and explained, as is shown in Equation 3.5 [107], [109], [100], [111]:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[\hat{y}_{ij}(t) - x_{ij}(t)\right]$$

$\hat{y}_i$ is the neighborhood best, defined as:

$$\hat{y}_i(t+1) \in \left\{N_i \middle| f(\hat{y}_i(t+1)) = \min f(x), \forall x \in N_i\right\}$$

With the neighborhood defined as:

$$N_i = \left\{y_{i-n,Ni}(t), y_{i-n,Ni+1}(t), y_{i-1}(t), y_i(t), y_{i+1}(t), y_{i+n,Ni}(t)\right\}$$

where N_i is the neighbourhood size. (3.5)

3.7.3 PSO Pseudocode

The framework of the original PSO is shown in Figure 3.14 (algorithm 1) as a Pseudocode. From the flow of the iterative process, it is noticeable that each particle flies to the global best particle in the swarm; this leads to a severe drawback of over learning from the best particle. Consequently, the diversity of the whole swarm will drop down dramatically [106].

Algorithm 1 Original Particle Swarm Optimizer

//Initialize swarm S
for $i := 1$ *to swarmsize* **do**
  for $d := 1$ *to* D **do**
    $V_i^d := rand[V_{min}, V_{max}]$; $X_i^d := rand[X_{min}, X_{max}]$;
  end for
end for
Compute the fitness value of each particle $F = (f_1, f_2, \ldots, f_{ps})$;
Set the $\boldsymbol{pbest} = (pbest_1, pbest_2, \ldots, pbest_{ps})$ and the $gbest$;
Set the acceleration constants c_1 and c_2;
Set the iteration counter t:=0;
while $t \leq Gen$ **do**
  for $i := 1$ *to swarmsize* **do**
    for $d := 1$ *to* D **do**
      //Update the velocity V_i^d of particle X_i using Eq.2
      $V_i^d := V_i^d + c_1 \cdot rand1_i^d \cdot (pbest_i^d - X_i^d) + c_2 \cdot rand2_i^d \cdot (gbest^d - X$
      //Update the position X_i^d of particle X_i using Eq.3
      $X_i^d := X_i^d + V_i^d$;
    end for
    Evaluate the fitness value f_i of the new particle X_i;
    if f_i is better than the fitness value of $pbest_i$ **then**
      Set X_i to be $pbest_i$;
    end if
    if f_i is better than the fitness value of $gbest$ **then**
      Set X_i to be $gbest$;
    end if
  end for
  if termination condition is met **then**
    break;
  else
    $t := t + 1$;
  end if
end while

Figure 3.14: Original Particle Swarm Optimizer [106]

3.7.4 Test Functions

The test functions are designed to deal with the optimization problems and procedures the testing. Almost the entire test functions that have appeared in the optimization literature are nonlinear least squares [115]. Table 3.2 presents the most common test functions of PSO. The test functions are used to find the adequate candidate as a solution.

Test Function	Formula
Ackley	$-20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{d=1}^{D} x_d^2}\right) - exp\left(\frac{1}{D}\sum_{d=1}^{D}\cos\left(2\prod x_d\right)\right) + 20 + e$
Rastrigin	$\sum_{d=1}^{D}\left(x_d^2 - 10cos\left(2\prod x_d\right) + 10\right)$
Step-Optimized	$0.5 + \frac{sin\sqrt{x^2+y^2} - 0.5}{\left(1.0 + 0.001(x^2+y^2)\right)^2}$
Eggcrate	$Minimize\ f(x) = x_1^2 + x_1^2 + 25(sin^1 x_1 + sin^2 x_2)$
Rosenbrock	$\sum_{d=1}^{D}(100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2)$

Table 3.2: PSO Functions

3.7.5 Rastrigin and Ackley Functions

Rastrigin's function as shown in Equation 3.6 is a scalable, separable, multimodal problem, it has a known global optimum on the origin and a huge number of local optima. The defined search space was $[-5,5]^D$ [116], [117].

$$\sum_{d=1}^{D}\left(x_d^2 - 10cos\left(2\prod x_d\right) + 10\right) \tag{3.6}$$

Ackley's function is a multi-modal, non-separable and scalable problem, it has a known global optimum on the origin and very small decreasing area around the optimum. The defined search space was $[-32,32]^D$, where D represents the number of dimensions of the search space [116], [117]. Ackley's function is formulated as shown in Equation 3.7 [117]:

$$-20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{d=1}^{D}x_d^2}\right)-exp\left(\frac{1}{D}\sum_{d=1}^{D}\cos\left(2\prod x_d\right)\right)+20+e \quad (3.7)$$

3.7.6 Step-Optimized Function

In Step-Optimized PSO (SOPSO), every particle has its own velocity weights. A particular setting of the velocity weights is referred to as the position of the velocity weights. An objective function for the velocity weights is used to quantify how well the positions of the velocity weights perform for solving the optimization problem [120]. Step-Optimized function is formulated as explained in Equation 3.9 [120]:

$$0.5+\frac{sin\sqrt{x^2+y^2}-0.5}{\left(1.0+0.001(x^2+y^2)\right)^2} \quad (3.8)$$

3.7.7 Eggcrate Function

In this problem, there are two design variables with lower and upper bounds of [-2π, 2π]. The Eggcrate function has a known global minimum at [0, 0] with an optimal function value of zero [120]. This function is described in Equation 3.10:

$$Minimize\ f(x)=x_1^2+x_1^2+25(sin^1x_1+sin^2x_2) \quad (3.9)$$

3.7.8 Rosenbrock Function

The Rosenbrock function is often used as a test problem for optimization algorithms. It has a global minimum of 0 at the point (1, 1) [118], [119]. A test function like Rosenbrock is useful because it test the ability of the algorithm to follow curved valleys [115]. The Rosenbrock function is formulated as explained in Equation 3.11 [115]:

$$\sum_{d=1}^{D}(100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2) \tag{3.10}$$

3.8 Summary

The optimization model using the Particle Swarm Algorithm has been introduced in this chapter. The technique mentioned uses AI to improve and optimize the application behavior and results. The PSO algorithm is very tolerant and it can use imprecision to adapt to the correct target. In the next chapter, the PSO algorithm will be applied to MPEG-4 transmission over ZigBee optimization problems.

Chapter 4: Methodology and Intelligent Technique Implementation

Simulation is fundamental in this research in order to make a detailed design of the various components in the ZigBee and MPEG-4 system, as well as to evaluate the system-level performance. To achieve the results, ZigBee's peer-to-peer topology including transmitter, reviser are simulated through operating on an existing IEEE 802.15.4 framework in Matlab and Simulink. The video streaming over ZigBee network simulated using MPEG-4 encoder and decoder in Matlab and the design of the system and its workflow with applied intelligent system technique (PSO) in MPEG-4 encoder application; explained.

4.1 Application workflow process

The system is composed of an MPEG-4 encoder to provide the data input to a network that organized into two nodes; transceivers and receivers, which is then passed onto a decoder. Figure 4.1 is block diagram of the complete system in a high-level overview of the proposed solution.

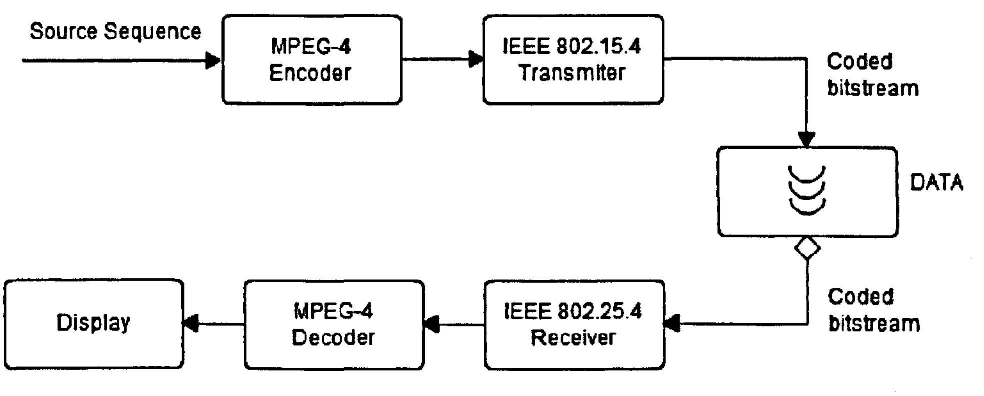


Figure 4.1: High-level application workflow and implementation of ZigBee network and MPEG-4

As shown in Figure 4.1, the system is made up of MPEG-4, and ZigBee's transmitter and receiver. The input of the system is an interactive audio and video file (AVI) for the encoder and the expected output file from the decoder is an AVI video file. The input data AVI is processed by the MPEG-4 encoder and streamed over the IEEE 802.15.4 standard network. The source data is then converted by the encoder into a compressed form, with a reduced number of bits, which is then transmitted and converted back into a representation of the original data.

The encoder consists of a temporal model, a spatial model and an entropy encoder; it processes and analyses each frame and GOP, which is composed of a sequence of three frame types; Intra coded frames (I-frames), Predicted frames (P-frames) and Bi- frames (B-frames).

Next, a loop filter implemented and placed it inside the MPEG-4 Simple Profile encoder for encoding and decoding. The filter is made of particle swarm optimization and the use of PSO has made an intelligent approach to the MPEG-4 encoder. An adaptive solution that introduced in this research than carried to finds the best quantization step size.

Lastly, the entropy encoder processes and converts the given elements of the video sequence after the filter, including GOPs, quantized transform coefficients, motion vectors and the side information that is used to correct the decoding, known as supplementary information into a compressed bit stream suitable for transmission. In addition, the same simulation process conducted by adding additive white Gaussian noise (AWGN) in a simulator of 802.15.4 device communication that comprises the transmitter radio channel as is shown in Figure 4.2.

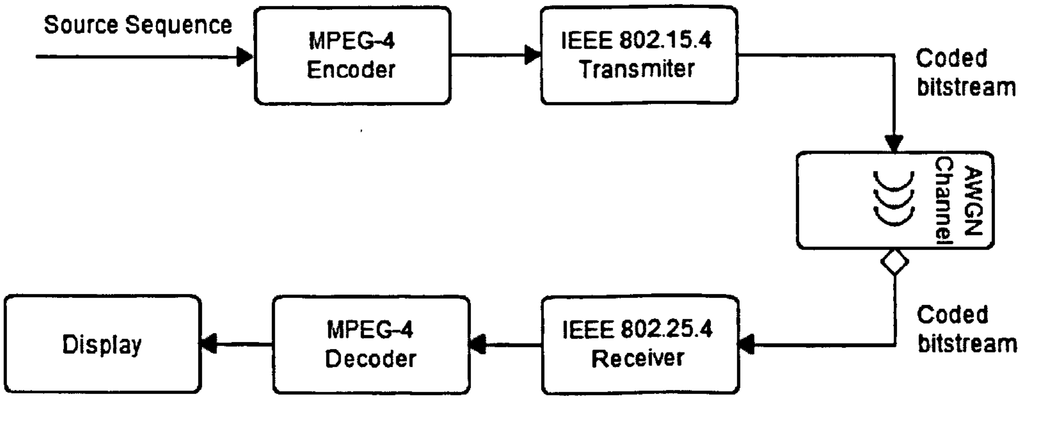


Figure 4.2: High-level application workflow and implementation of ZigBee network and MPEG-4 with noise (AWGN)

After the encoding part, the processed symbols during the encoding are stored as references in flat files systems to ensure that the decoders can successfully incorporate with the encoder by using those references. Furthermore, an evaluation of the different rates; Peak Signal-to-Noise Ratio (PSNR in dB), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and quality are measured before and after the transmissions to evaluate and determine the error level, address the data loss and assess the quality of service during the transmission. Each block diagram and how the system works in depth including the use of PSO in encoder whilst streaming MPEG-4 video continuously over the ZigBee network.

4.2 IEEE 802.15.4 ZigBee Network

The ZigBee network with a transmitter and the receiver is simulated in Matlab and Simulink. The initial IEEE 802.15.4 simulations have been carried by modifying and extending to a ZigBee Modulation and Demodulation model created by Jitesh [122] to understand the physical layer blocks used in ZigBee Protocol. Furthermore, TrueTime framework [123], [124] is used and adopted to stream video over IEEE 802.15.4. TrueTime is a Matlab and Simulink-based simulator for real-time control systems and facilitates co-simulation of controller task execution in real-time kernels, network transmissions. In this research, focus is on the application layer of ZigBee but in order to stream MPEG-4 and achieve the result the physical layer has been simulated in Matlab/Simulink using IEEE 802.15.4 using worldwide channel 16. In the network simulation, two ZigBee nodes are used as is shown in Figure 4.3 with a data rate of minimum 80kb and a maximum frame size of 272 bits to prepare the network. The ZigBee nodes are created from TrueTime kernel Simulink blocks, clock offset (that is a constant time offset from the nominal time), clock drift, (known as time drift) 0.01 if the local time should run 1% faster than the nominal time for actual simulation time and power source or battery. Figure 4.4 and Figure 4.6 show the transmitter and receiver under the kernel blocks [123].

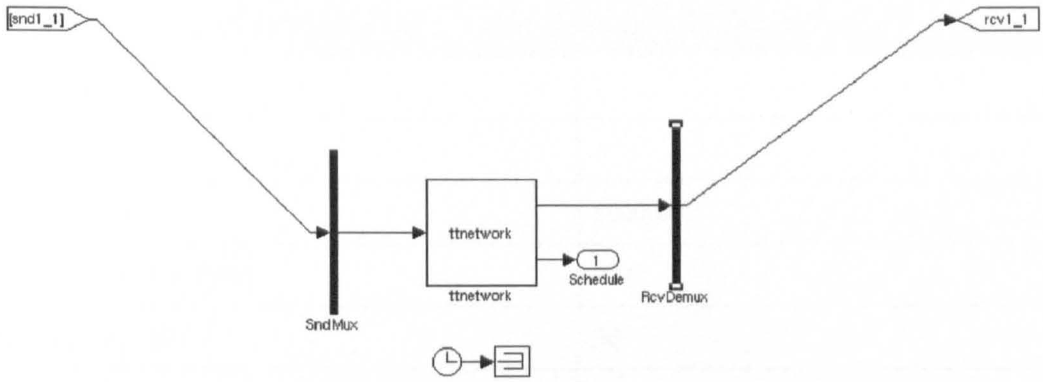


Figure 4.3: ZigBee network with two nods

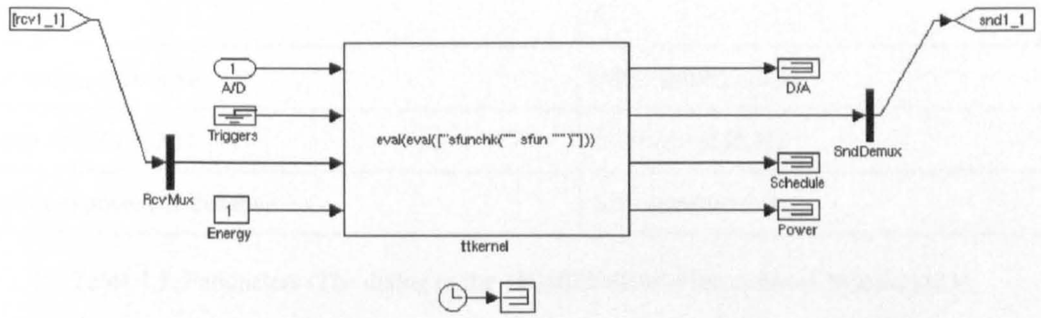


Figure 4.4: IEEE 802.15.4 Regular node (Coordinator)/Receiver

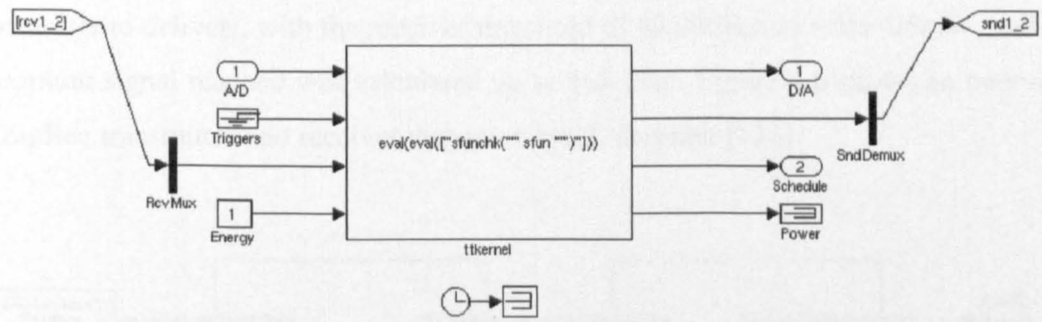


Figure 4.5: IEEE 802.15.4 Actuator/Transmitter

Table 4.1 shows the complete set of parameters.

Parameters	Value
Network number	1
Node number	2
Data rate (bits/s)	800000
Minimum frame Size (bits)	272
Transmit power (dB)	30
Receiver signal threshold (dB)	-48
ACK timeout limit (s)	0.00004
Retry Limit	5
Error coding threshold	0.03 - interval [0,1]
Loss probability (0-1)	0 - (interval [0,1])
Pathloss exponent (1/distance ^ x)	3.5 - (interval 2-4)

Table 4.1: Parameters (The dialog of the TRUETIME wireless network blocks) [123].

In this wireless network simulation transmit power has reached from 30.00dBm to a maximum of 1000.00mW (ZigBee requires at least -3dBm of output power for the transmitter to deliver), with the receiver threshold of 48.00dBm to 1.58e-05mW, and the maximum signal reached was calculated up to 168.27m. Figure 4.6 shows an overview of ZigBee transmitter and receiver through a block diagram [122].

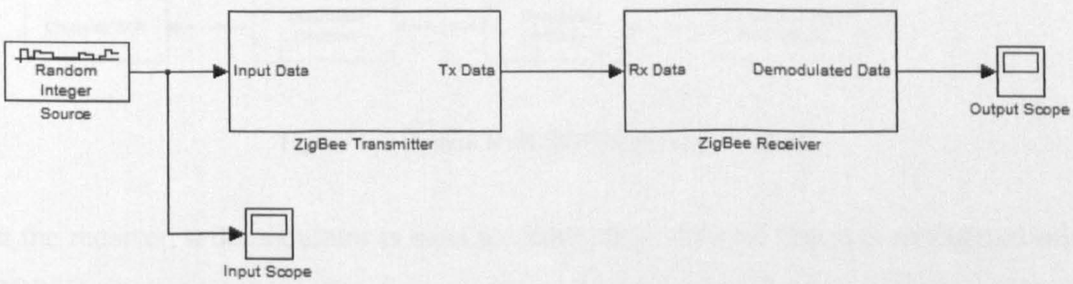


Figure 4.6: ZigBee transmitter and receiver high-level over view of block diagram in Matlab

The simulations in IEEE802.15.4 design is achieved by applying bit to symbol mapping, symbol to chip mapping, half sine pulse shaping, and modulation with high frequency carrier and by using direct spread spectrum technique. Figure 4.7 shows a diagram block of the modulator and demodulator in the simulation.

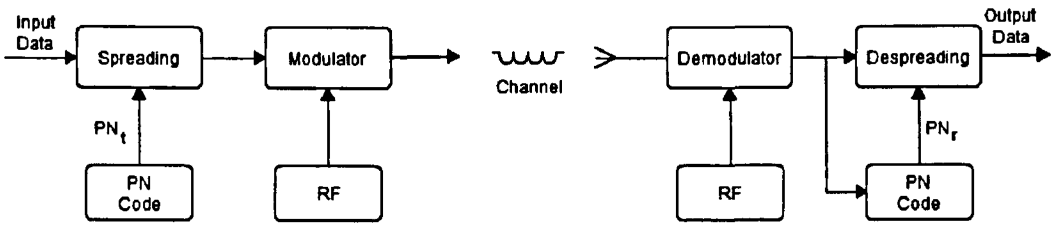


Figure 4.7: Modulator and Demodulator block diagram

The ZigBee transmitter simulation is based on spread bits, which are then, modulated using an OQPSK modulator and DSSS as mentioned in the chapter two for modulating radio-signals stream in physical layers. Figure 4.8 illustrates the process of QPSK in transmitters and receivers, with the added noise channel. DSSS modulates information before sending it on to the physical layer. It does this by converting the information into four different signals causing all the information to be transmitted through a large bandwidth but reducing the spectral power density.

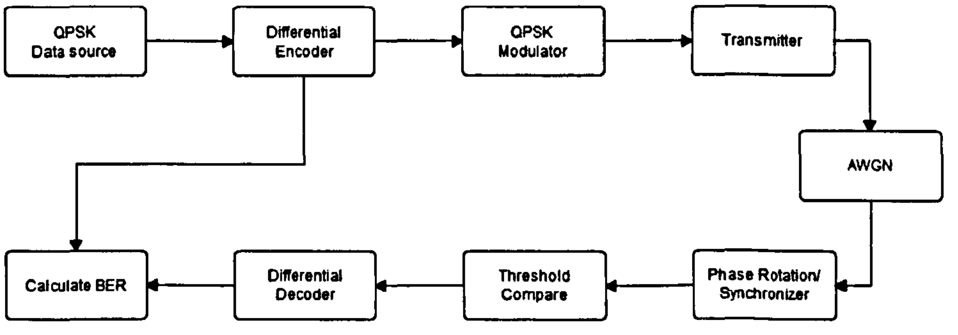


Figure 4.8: System level OQPSK process diagram

In the receiver, a demodulator is used to demodulate a signal that was modulated using OQPSK at transmission. The de-spreading method works with the delayed version of PN chip sequence in transmitted and received bits in order to demodulated the data contains chip rates.

4.2.1 ZigBee Transmitter and Receiver and Noise

Interferences and noise are very damaging to the quality of transmission; therefore, AWGN added to the network in simulation. The noise signal that is multiplied and added to the original signal that passes through is shown in Figure 4.9 [122], [123]:

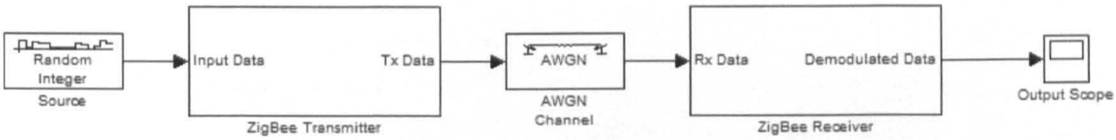


Figure 4.9: Additive White Gaussian Noise (AWGN)

The Gaussian noise ration input / noise value (Es/No) is 10 dB with an initial seed of 35, an input signal power of 0.001 ohm (watts) and symbol periods of 4e-6 to the AWGN channel, in order to measure the signal to noise ratio and the ratio of the energy per bit (Eb) to the spectral power noise density (No). These values and the integration bandwidth are then able to calculate the signal-to-noise ratio using the Equation 4.1

$$SNR = \frac{Eb}{No} - Reference\ Channel\ Power - 10 \times \log 10\,(Integration\ Bandwidth \div Bit\ Rate) \quad (4.1)$$

4.2.2 ZigBee network with Bit Error Rate

The Bit Error Rate ensures the digital link quality and is calculated by dividing the number of received bits with errors by the number of transmitted bits. The BER unit is known as a less performance measurement and its example is expressed as a percentage.

$$BER = \frac{Bits\ in\ Error}{Total\ Bits\ Recived}$$

Initial BER sampling result shows in Figure 2.1. Noise has a major impact upon the BER performance, especially when the data is transmitted; hence BER is measured in this simulation after adding the AWGN, the Figure 4.11 is showing the result. The

Figure 4.10, [122] shows the diagram block ZigBee network and BER measurement result.

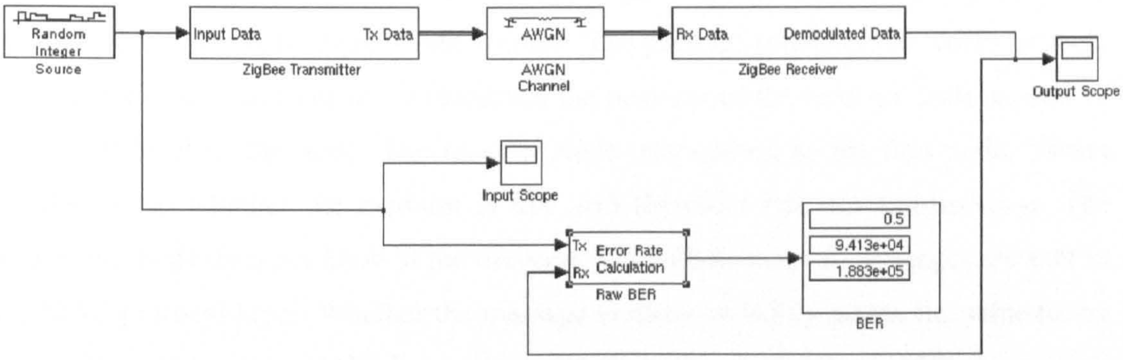


Figure 4.10: IEEE 802.4.15 BER [108]

Figure 4.11 shows the BER simulation result [122].

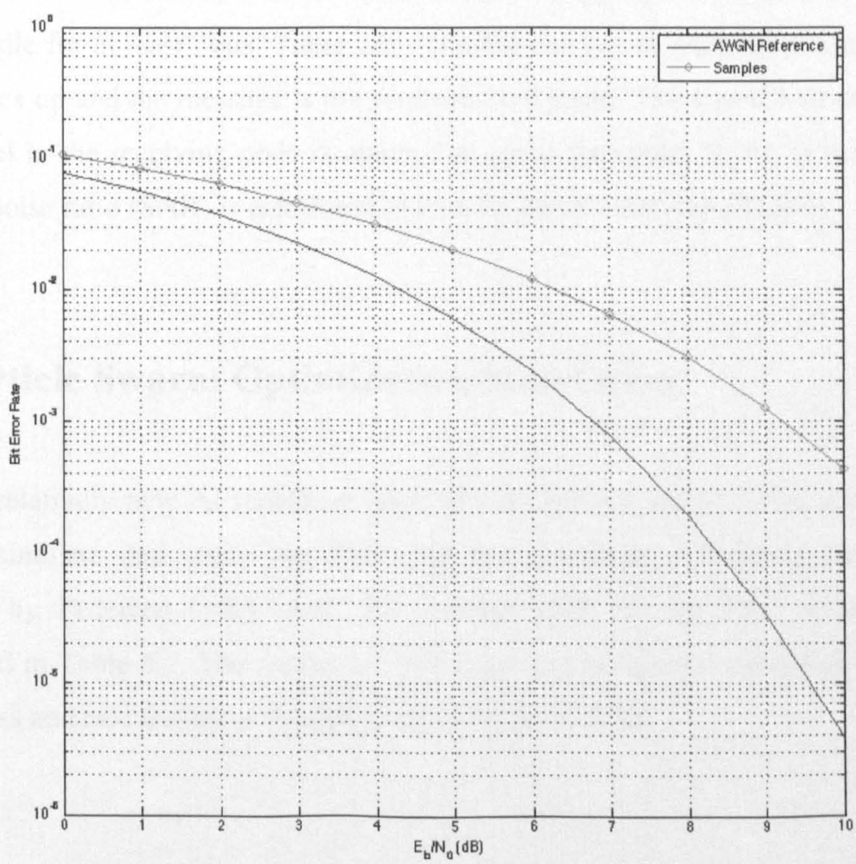


Figure 4.11: Bit error rate of white Gaussian noise in IEEE 802.15.4 simulation

4.2.3 ZigBee TrueTime Framework

Using the IEEE 802.15.4 (ZigBee) wireless network block from TrueTime framework, a realistic network created in simulation [123]. ZigBee network, in this simulation, takes the path-loss of the radio signals into account. The package transmission works as such: when a node starts to transmit, it calculates the position of the receiver node as well as the signal level in the node. The receiver node, responding to the first node, packet checks to see whether the medium is idle and therefore free for transmission. The transmitter node does not know if the message will collide so ACK messages are sent to the MAC protocol layer. Whether the message collides or is lost means the same to the transmitting node as no ACK is received. If this is the case and no ACK message is received, the node will wait a random period, within a contention window, until retransmitting the message.

The size of the contention window doubles each time the same message is retransmitted; which occurs if the back-off timer is stopped, the medium is busy or it has been idle for at least 50us. There are a limited number of retransmissions until the sender gives up and the message is not retransmitted again. The signal will detect if the signal level in the receiving node is above the signal threshold. If this is the case, the signal-to-noise ratio (SNR) is calculated to find the block error rate (BLER).

4.3 Particle Swarm Optimization Simulation

PSO is a relatively new AI technique, thus Matlab has not yet provided a toolbox. In order to simulate and study the PSO and test functions a Toolbox (SwarmOps) published by Pedersen [100] used. The settings used for the PSO simulation are represented in Table 4.2. The results are discussed in next chapter including the gBest, Best Fitness and calculation of desired Q-steps for each GOP.

Function	Rosenbrock
Max velocity divisor	2
Number of particles	240
Iterations /Maximum Evaluation	100
Lower Initialization (Q-step)	1
Upper Initialization (Q-step)	31
Test function	Rosenbrock
Dimension	4
Acceptable Fitness	1.0000e-03
Lower Bound	$-n$
Upper Bound	n

Table 4.2: Parameters used in PSO simulation

4.4 Multimedia

The high-level system design of MPEG-4 will be discussed and explained in this section, (apart from the three blocks that are of no importance in this research; MPEG-4 Decoder, IEEE 802.15.4 Transmission and IEEE 802.15.4 Reception). In addition, the relative parts of the MPEG-4 system will be simulated and the algorithm of the proposed intelligent system that uses AI will be described in block diagrams. The MPEG-4 system will be carefully looked at and decomposed into blocks and sub-diagrams. Each sub-diagram and the relative algorithm in this research will be explained in detail in separate subsections below.

The input of the system as shown in Table 4.3, are MPEG-4 video files used with the attributes of 176 pixels by 144 pixels and 64kbtis/s data rate. In this simulation, 240 frames (20 GOP) for a period of 20 seconds streamed using two video files, one with added Gaussians noise and other one as a basic MPEG-4.

File type/name	Wolf.avi	WolfWithGaussian.avi (Noise added)
Bitrate	64 kbps	64 kbps
Duration:	20 sec	20 sec
Size	148 kb	171kb
Resolution	176x144	176x144
Max objects	1 simple	1 simple
Frame rates	24 fps	24 fps

Table 4.3: Input files information

4.4.1 PSO implementations in MPEG-4 Encoder

The method that proposed to the applications of MPEG-4 using PSO explained in this section. Figure 4.24 illustrates a high-level overview of the implementation. In the Figure 4.12, block diagram, once the process starts from the first block, initialization including converting and resizing the video file from CIF (default) frame format to QCIF and sub QCIF and convert frame from RGB format to YCbCr (YUV) format are done. Then the second block becomes the encoder which PSO method implemented, which in its core foresees all the coding part. The third block is created to capture the data that is generated during the encoding and store them in a flat file database.

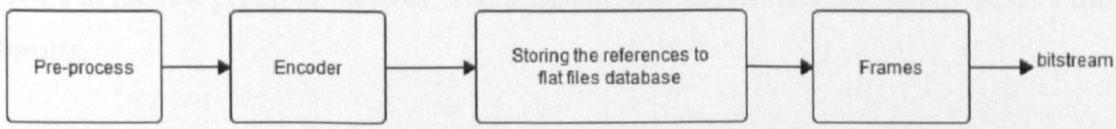


Figure 4.12: MPEG-4 Encoder diagram

The encoded frame is stored for further motion-compensated reconstruction. In order to ensure that the decoder uses an identical reference frame, the filter reference data and block data are stored in the flat file system database for each frame. The references in the database files are required for decoding. Then the decoder for further motion estimation and compensation uses the encoder references that are stored as flat file system database that contains the data and block data for each frame. Each reference file contains the parameters of both the encoder and the decoder including the Q-step sizes. The last block in the diagram processes the frames with a given size to correspond to the requirement of the transmitter in bitstream.

4.4.2 Decoder

The decoder works in reverse to the encoder. The diagram in Figure 4.13 shows the decoder process that introduced this research. The first block (initialization) is very important during the decoding. Initialization is the processing and fetching of the stored references in parallel with the video sequences data that are stored as flat file system during the encoding in order to use the parameters needed for decoding.

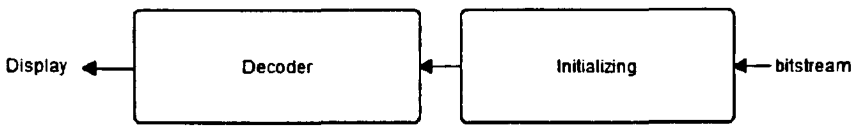


Figure 4.13: Decoder

The decoder, after the initialization, the than follows the decoding process, uses decoder uses the entropy decompression to decompress the picture using the coefficients obtained in the DCT decompression. The DCT is inverted and the entropy decompressed. Lastly, the frames are rendered from the first given group of picture GOPs to the last group of pictures. The program then writes the total clip in an AVI file format.

4.4.3 Encoder process

The MPEG-4 encoder is based on pre-sets of encoding parameters that are usually set in advance of encoding a video sequence. The encoding parameters have a major impact on the quality of service in decoded video sequences and its streaming. The initialization determines the parameters needed for the simulation at the start. The required parameters are bandwidth, Q-step for each frame and range of GOPs. The bandwidth in this research and simulation is the maximum value available in IEEE

802.15.4 ZigBee set at 250kbps. The next input data is the GOP's range that is from 1 to 20 GOPs, which means approximately 240 frames. The other parameters are for the quantization; for Variable Bit Rate (VBR) the values are 8 for I-frames, 10 for P-frames and 25 for B-frames. For Constant Bit Rate (CBR) the parameters have been set to a constant value of 15 for all the frames I, P and B. MPEG-4 uses the VBR and CBR rate control system to stream video. Rate control helps to ensure that the video coded bitstream can be transmitted successfully and whilst making full use of the limited bandwidth.

The adaptive system in combination with rate control also uses the initial value of 15 for each Q-set to start with. However, soon after running the simulation, it will override the value of its initial settings to decide adaptively on the required Q-sets to find the best fit for the purpose and transitions rates.

The MPEG-4 standard requires each video frame or object to be processed in units of a macroblock. MPEG-4 uses Variable Bit Rate (VBR) and the Constant Bit Rate (CBR) to set the control parameters of a video encoder. Encoding parameters are needed to control the output bitrate. The Quantizer Parameter (QP), or quantizer step size, is the most obvious parameter to vary, or rescale. This is because increasing the QP reduces the coded bitrate, whilst decreasing the QP will increase the coded bitrate. As mentioned in chapter two, motion estimation has a major role in the encoding performance, therefore if motion estimation search area and quantization step size are kept constant, then the number of coded bits produced for each macroblock will change depending on the content of the video frame. This causes the bit rate of the encoder output (measured in bits per coded frame or bits per second of video) to vary which will lead to a video with varying degrees of quality instead of producing a video with constantly excellent quality [61]. Hence, an intelligent system that consists of a number of steps used to find the balance between quality of service in combination with Rate Control Scalability (RCS). The proposed algorithm is the second process after the DCT coefficients, as much of the signal energy is at low frequencies and therefore the rest of coefficients have little energy. Hence, quantization helps to remove the unimportant values and divide the values by a non-zero positive integer known as a "quantization value" and round the quotient to the closest integer. The process starts of by being based on a constant bit rate as is described in Figure 4.14.

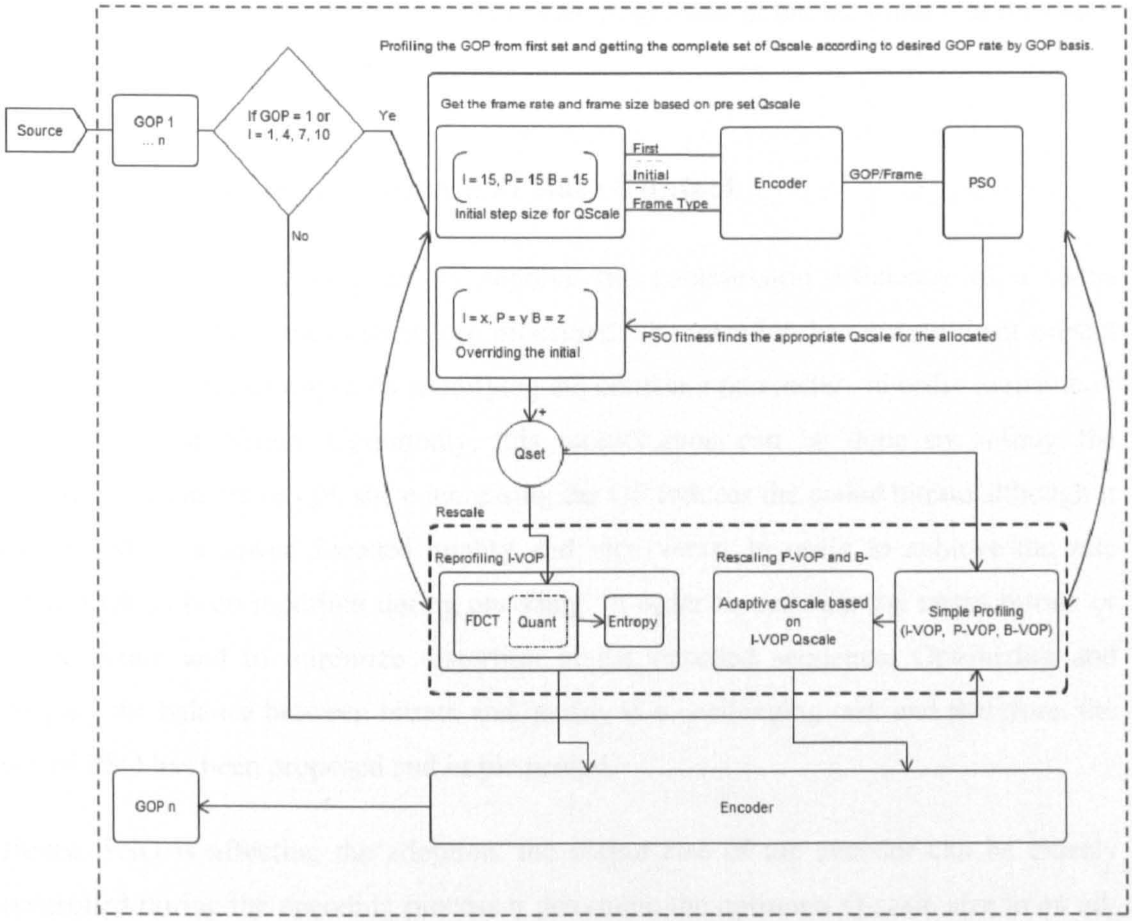


Figure 4.14 Overview of adaptive design implementation

As is shown in Figure 4.14, the VBR and CBR control methods are used to stream the video sequence. The proposed intelligent system, which is based on CBR and PSO, should minimize data loss and distortion whilst ensuring that the decoder does not suffer from underflow or overflow. These details will be explored further in the following section.

4.4.4 Quantization

Scalar quantization works with both the forward and inverse transform by dividing or multiplying the constant parameters. Different Scalar quantization (Q-scale) values influence the amount of compression; the more a video is compressed the worse the quality of the video and vice versa. Q-scale value can be set for P, B and I-frames separately with a scale of 1 to 31. The larger the number, the better the video will be compressed and therefore, the easier that video is to transmit but the quality of the video will have been greatly affected.

4.4.5 Adaptive Quantization and Rate Control

As mentioned earlier, in order to improve the compression efficiency of a video CODEC the video frames can be pre-processed. Quantization has a significant impact on rate control, for example, by modifying the encoding parameters in order to maintain a target output bitrate. Commonly, this modification can be done by setting the quantizer parameter or QP, since increasing the QP reduces the coded bitrate although it can introduce a lower decoded quality and vice versa. In order to achieve the rate control QP is been modified during encoding, in order to maintain the target bitrate or mean bitrate and to minimize distortion in the decoded sequence. Optimizing and keeping the balance between bitrate and quality is a challenging task and therefore, the use of PSO has been proposed and implemented.

Hence, PSO is affecting the adoption, the output rate of the encoder can be closely controlled during the encoding process it determine the optimum Q-scale size in an ad-hoc way. The QP will be set for each VOP and macroblock to target the bitrate restriction. This approach should eliminate any data loss and packet drops. Figure 4.29 shows the process of the implementation of proposed idea. Figure 4.15 illustrates a deep look into the main core functionality of this process. The concept of MPEG-4 compression is explained in chapter two, and based on the following algorithm each block has been used in combination with the adaptive Q-scale system, except for those blocks that are used for the MPEG-4 core system.

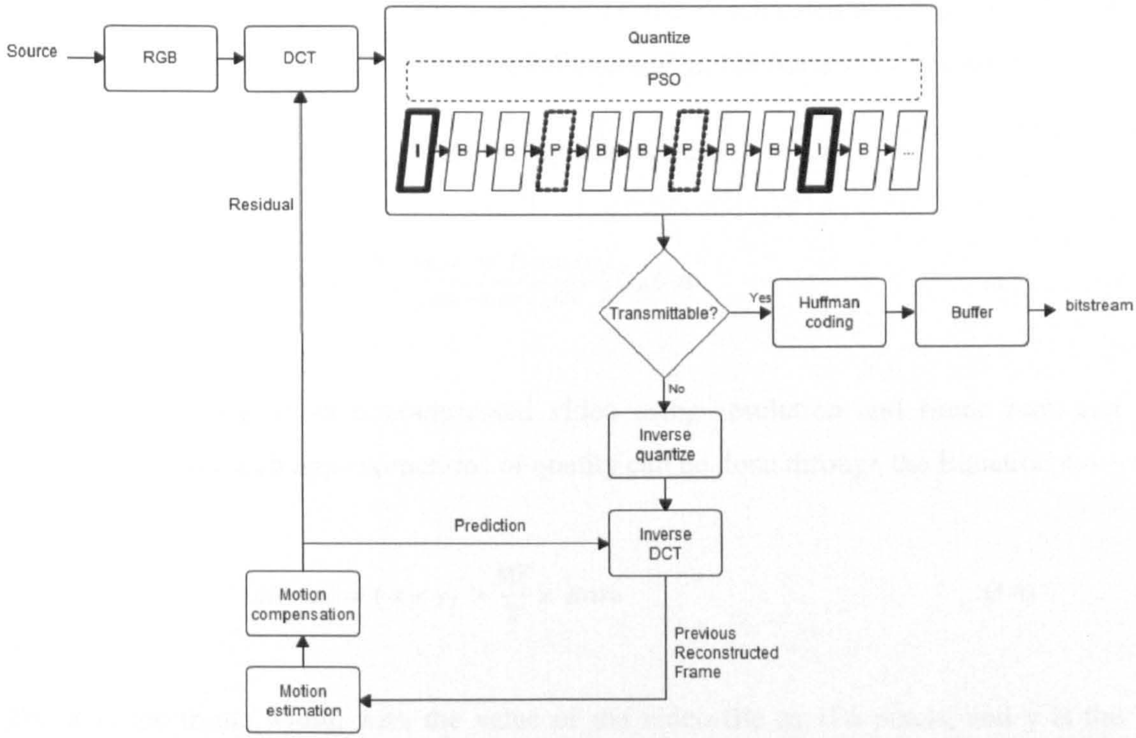


Figure 4.15: Adaptive quantization process

As is shown in Figure 4.15, during the processing of the first GOP, the I-frame bitrate and frame size are collected and given to the PSO system, in addition for each P-frames and B-frames the total bitrates and frame sizes are calculated to predict and to set the first QP steps to initialize the rest of the GOPs. The basic formula shown in Equation 4.2 is to determine the Scalar Rate Control (SRC) [61].

$$R = A\left(\frac{X_1}{Q} + \frac{X_2}{Q^2}\right) \tag{4.2}$$

Basic R is calculated by computing the quantizer step size (Q-step), which is applied to the whole frame, and A is the mean absolute difference of the residual frame after motion compensation for each frame I, P and B before encoding.

The target bit rate R, is calculated based on the number of frames in the GOP and the minimum and maximum level of bits that are available by calculating the prediction P-frame rate plus a virtual buffer. The buffer's maximum size is determined by estimating the complexity of each frame. If the previous frame is an I-frame, it is used as a reference to predict the next frame's complexity and is allocated a suitable number of bits, next the quantize step size Q for the following P and B-frames is calculated.

Find the desired bit rate or target rate example is expressed in 4.3:

$$Target\ rate = \left(\frac{Number\ of\ frames}{frame\ rate\ (24)}\right) ZigBee \quad (4.3)$$

Finding the bitrate of an uncompressed video using resolution and frame rate, and lossless video through approximations of quality can be done through the Equation 4.4:

$$Bitrate = (x \times y) \times \frac{\mathrm{MF}}{\mathrm{B}} \times Rate \quad (4.4)$$

The x is the frame width, with the value of the video file as 176 pixels, and y is the height with the value of 144 pixels. The number 4 is the value of MF (motion factor), which is divided by 8 bits, and therefore, 1000 is the value of rate which resulting bitrate of frame. The value of bitrate than is passed to the PSO for optimization. The optimization process is formulated as given in Equation 4.5 and 4.6 [100], [101]:

$$\vec{v} \leftarrow w\vec{v} + \varphi_p r_p(\vec{p} - \vec{x}) + \varphi_g r_g(\vec{g} - \vec{x}) \quad (4.5)$$

where velocity is:

$$\vec{x} \leftarrow \vec{x} + \vec{v} \quad (4.6)$$

The search-space boundaries are set as minimum and maximum Q-scales in between 1 to 31, as lower and upper boundaries. Instead of letting f map the entire n-dimensional real-valued space, it is often practical to use only a part of this vast search-space. The lower and upper boundaries constitute the search-space and enforce the optimization method to move the candidate frame back to the boundary value if it has exceeded the boundaries that are denoted as $\vec{b}_{lo}$ and $\vec{b}_{up}$ as formulated in Equation 4.7 [100], [103]:.

$$f : [\vec{b}_{lo}, \vec{b}_{up}] \longrightarrow \mathbb{R} \tag{4.7}$$

If optimization problems are f functions these are explained in Equation 4.8 of the following form [100]:

$$f : \mathbb{R}^n \rightarrow \mathbb{R} \tag{4.8}$$

Assuming that f is a minimization problem, meaning that it is searching for the candidate solution $\vec{x} \in R^n$ with the smallest value $f(\vec{x})$ using the following example: Find $\vec{x}$ such that $\forall\, \vec{y} \in \mathbb{R}^n : f(\vec{x}) \leq f(\vec{y})$

It is often not possible to find the exact optimum and a candidate solution of sufficiently good quality must be used instead [100]. The evaluation of frame rates is then passed into the Rosenbrock function. The first input argument is the frame rates to be evaluated. Instead of iteratively recalculating the number of particles from the dimensionality of the position matrix, the information is passed to the function through the second input argument. The output is the column matrix of function values in Matlab, which correspond to each frame row or rate that has been evaluated. Personal and global bests, including the best Fitness, are updated based on how well they minimize the following Equation 4.9 [100], [118], [119].

$$f\left(\vec{x}\right) = \sum_{i=-30}^{30} \sum_{j=j-1}^{n-1} \left(100\left(x_{j+1} - x_1^2\right)^2 + (1 - x_i)^2\right)$$
$$-30 \leq x_i \leq 30 \tag{4.9}$$

After training the data the result of the PSO then determines the Q-Step size for each GOP. The next process will be to Zigzag scan the DPCM for DC coefficients and AC coefficients, and Entropy code the coefficients, which includes Huffman encoding for DC, AC and Motion vectors.

4.4.6 Decoder

The decoder then uses pre-determined Q-scale parameters to initialize and proceed with Inverse Quantization and process the video sequences. Figure 4.16 is a block diagram representing the decoder algorithm.

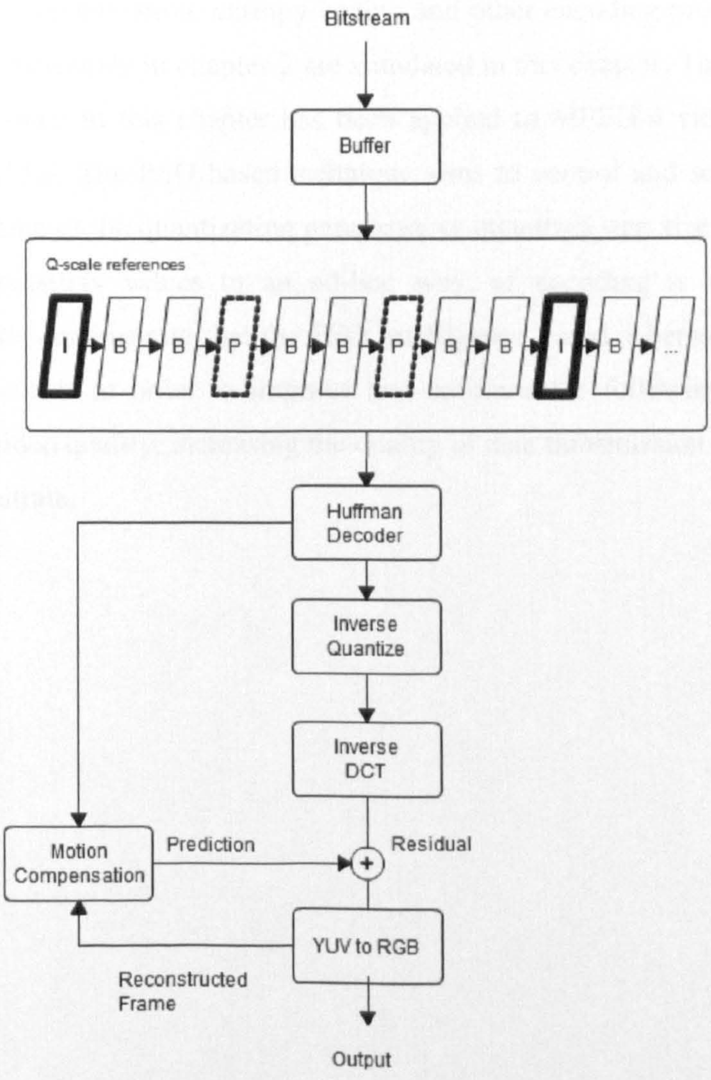


Figure 4.16: Decoding process after transmission.

4.5 Summary

The simulation presented in this chapter explains the implementation of a transceiver using OQPSK for the ZigBee wireless communication system using Matlab and Simulink. MPEG-4 compression techniques, such as, motion compensated prediction, transform coding, quantization, entropy coding and other encoding processes that have been explained previously in chapter 2 are simulated in this chapter. The proposed PSO technique mentioned in this chapter has been applied to MPEG-4 video transmission over IEEE 802.15.4. The PSO based technique aims to control and set an optimal, or near-optimal, value of the quantization parameter or quantizes step size by dividing and multiplying parameters values in an ad-hoc way, as encoding is in process. The simulation results demonstrate that the PSO intelligence based scheme is able to learn from previous events in order to improve and optimize the following results, whilst improving the video quality, increasing the quality of data transmission and transmitting at the targeted bitrate.

Chapter 5: Simulation Results

5.1 Streaming MPEG-4 over IEEE 802.15.4 without Intelligent System

In this section, streaming a video file over ZigBee by obtaining use of VBR and CBR transmissions have been simulated, and following this with a transmission simulation using PSO. The simulations are divided into two groups; A) streaming a video without added noise and B) streaming with added Gaussian noise using VBR, CBR and PSO in both groups. The work has been described the possibility of streaming over the IEEE 802.15.4 standard with the use of simulation in Matlab software. To simulate the streaming video over ZigBee in the closest possible real-life environment, Gaussian noise has been added into the video file. In order to simulate the transmissions, a peer-to-peer ZigBee network with throughputs of 250kbps and a video file with 24fps and 64kbps used. 20 GOPs are simulated that convert to 240 frames in this simulation. IEEE 802.16.4 peer-to-peer technology has an arbitrary transmission range of 50 meters. In this network each device can communicate directly with any other device so that there is a successful connection.

5.1.1 Video transmission over ZigBee with VBR

VBR is commonly used for video streaming as it tolerates a higher bit rate of MPEG-4 video to stream and has a higher amount of output data per time segment. For this reason, more storage space or buffer is needed for the more complex segments of the media file, while less space is needed for the less complex segments. The VBR method is used to stream a non-noisy video file with the specification that has been mentioned in the beginning of this chapter. Figure 5.1 shows the transmission result:

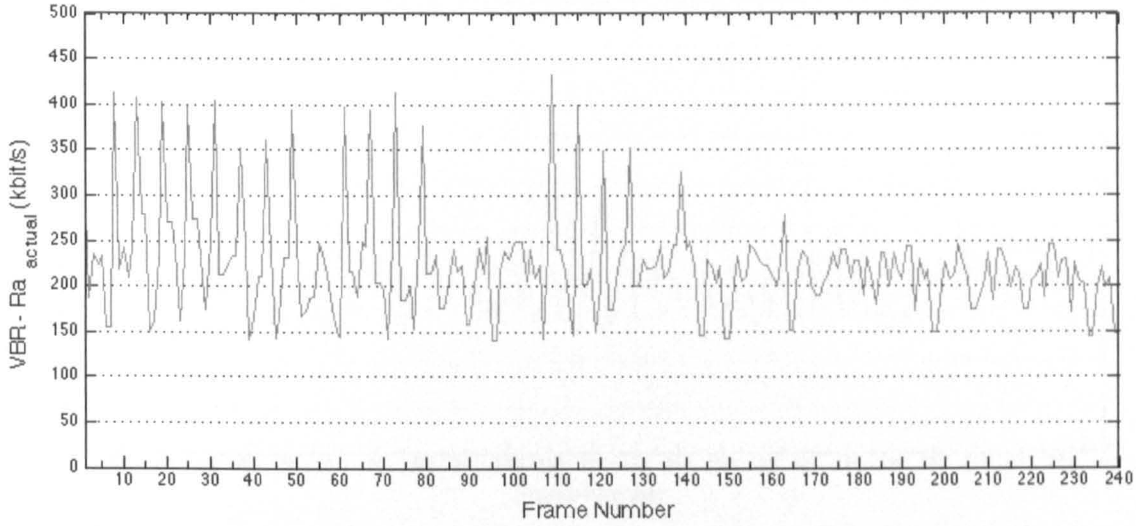


Figure 5.1 VBR transmit rates to stream 240 frames (20 GOPs)

According to the plot, a large amount of data is over the IEEE 802.15.4 standard throughputs limit, therefore, too much data has been dropped and the transmission with VBR has not been successful. For example, between the beginning frame and frame 180, the rate recorded before the transmission is mostly over the 250kbps maximum bandwidth all the time. Hence, lots of data has either not been transmitted or has been lost.

5.1.2 Video transmission over ZigBee with CBR

CBR, is mostly used when streaming a video on a limited bandwidth or limited capacity because it is the maximum bit rate that matters, not the average. CBR would not be the best option for storage but it is a good choice to stream MPEG-4 video as it would allocate enough data for complex sections but this is at the expense of destroyed or low quality.

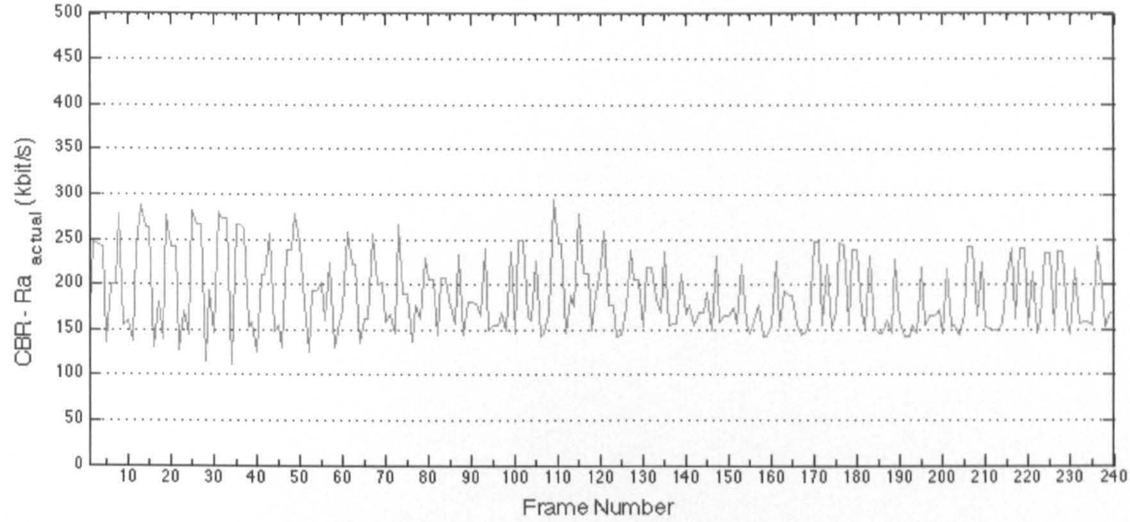


Figure 5.2: Video transmissions over ZigBee with CBR

Figure 5.2 illustrates the result of the video streaming simulation using CBR. Most of the numbers of data loss are from frame 1 to 120. The trend then (120-240) reverses and there is a steady downward tendency, but the frame rates have been in the given target bitrate limits. However, data loss has been significantly less than when using the VBR method.

5.1.3 VBR versus CBR in Video Transmission

In this simulation a comparison plot made for data frame rates between VBR and CBR methods. At the frame number scale, time series of VBR and CBR for transmission over ZigBee shows that VBR has a substantially greater data loss than CBR. As seen in detail in Figure 5.3 the differences between VBR and CBR indicate that in VBR from the beginning of transmission to frame 150 the majority of the transmitted data has surpassed the maximum of 250kbps. However, the CBR plot illustrates the greater part of our data loss is between the beginning frame-to-frame 120. The VBR model shows the most fluctuates in comparison to CBR and that is because the more complex regions of the video use more bitrate.

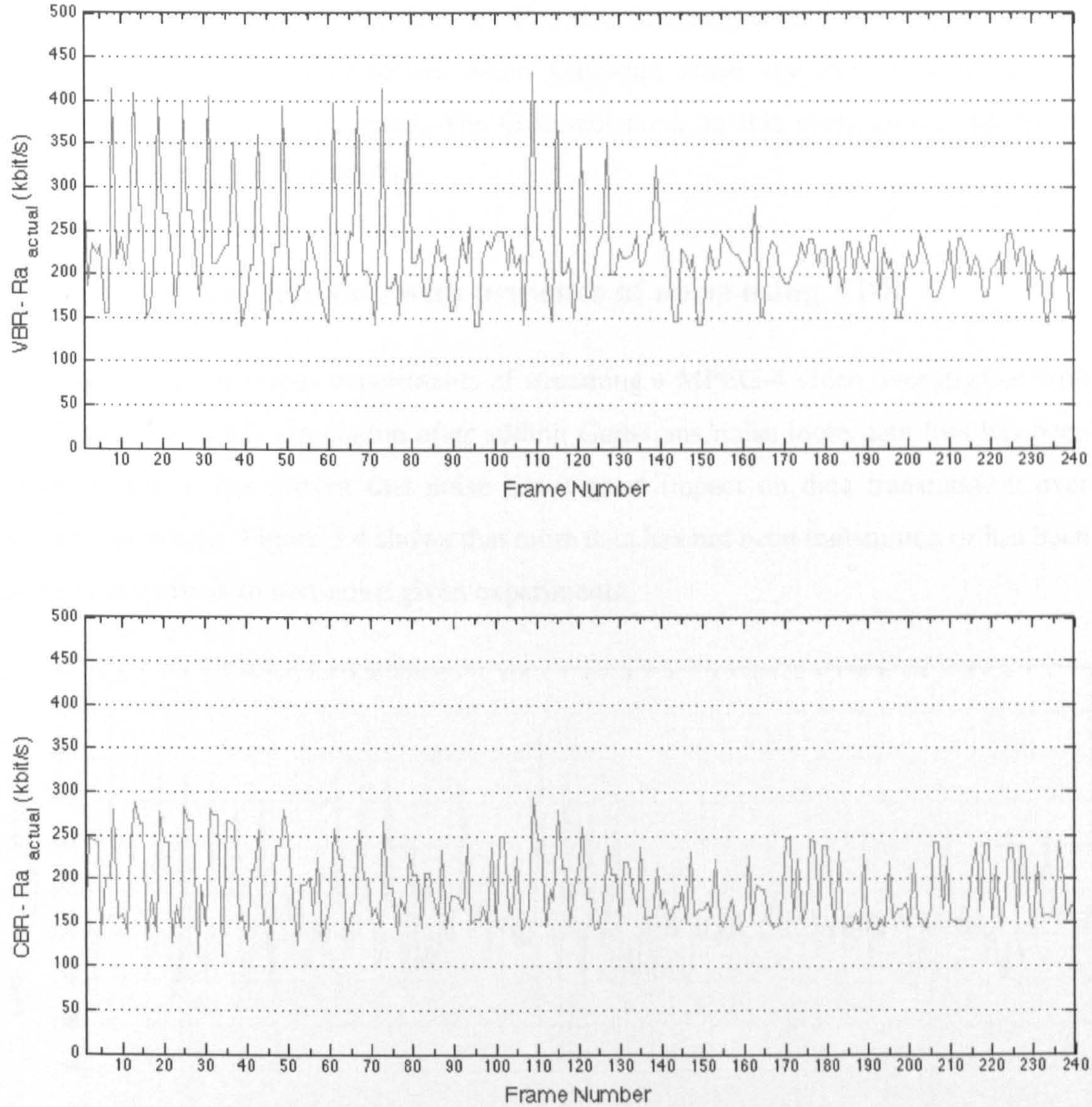


Figure 5.3: VBR and CBR sampling compression

5.2 Video transmission with presence of noise

White Gaussian noise added to the MPEG-4 video streams to simulate and measure the rates of data in the presence of interferences over the ZigBee wireless network. The purpose of adding noise is to simulate near to real world applications that are influenced by global noise, and to find out what could go wrong during the transmission. In general, any data transmission over a wireless network could be influenced by absorption, scattering or scintillations. The data throughput and the impact of data congestions with and without the white Gaussian noise and data loss results are collected and explained in detail. The Gaussian noise in this simulation is set by its rations input / noise value (Es/No).

5.2.1 Video transmission with presence of noise using VBR

Similarly, to the pervious experiments of streaming a MPEG-4 video over ZigBee with VBR method, in this simulation after adding Gaussians noise more data loss has been recorded and it has proven that noise has a great impact on data transmission over wireless networks. Figure 5.4 shows that more data has not been transmitted or has been lost in comparison to non-noise given experiments.

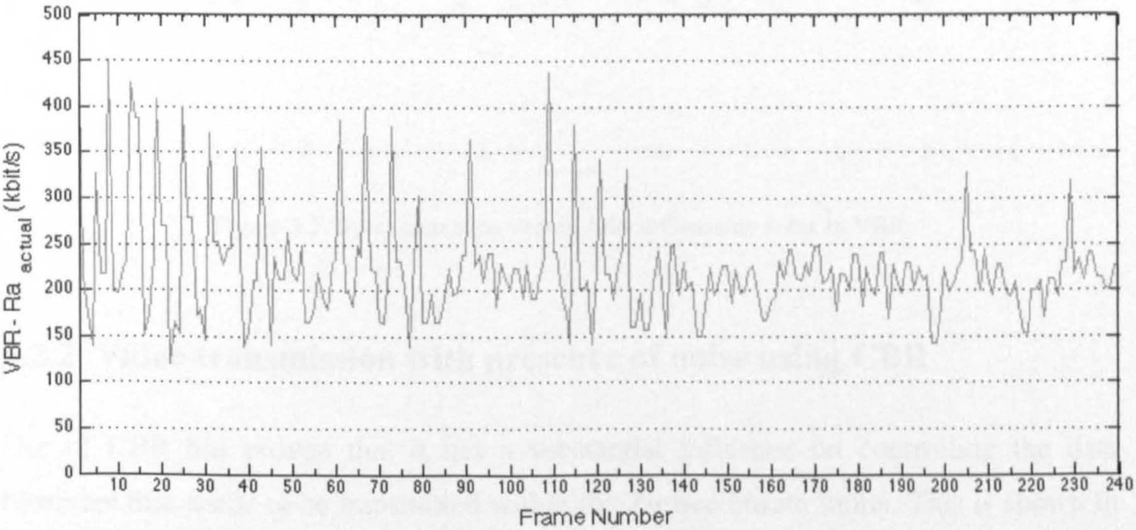


Figure 5.4: Video transmissions over ZigBee with the presence of noise using VBR

To see how the VBR method compares to the Gaussian noise influence on the transmission over ZigBee, Figure 5.5 provides and illustrates a plot in which the basic transmission is represented with a blue line with 'o' marks and the noisy MPEG-4 video is represented with ':' dated line and red triangle marks. In this plot, Gaussian noise shows a similar pattern to the transmission over ZigBee trade data. Both (VBR and VBR with Gaussian noise) indicate the high amount of data loss. In the VBR method, after frame 150, the trend is in our target bitrate and data trend is steadier. However, in VBR with Gaussian noise it is noticeable that the data loss even up to frame numbers

210 and 230. Streaming video with Gaussian noise over IEEE 802.15.4 indicates that noise causes more data loss and requires a higher bandwidth or buffer. The VBR method also shows that it may take more time to encode, as its process is more complex in order to produce a better quality of VOP. Furthermore, VBR poses problems during streaming when the instantaneous bitrate exceeds the data rate of IEEE 802.15.4.

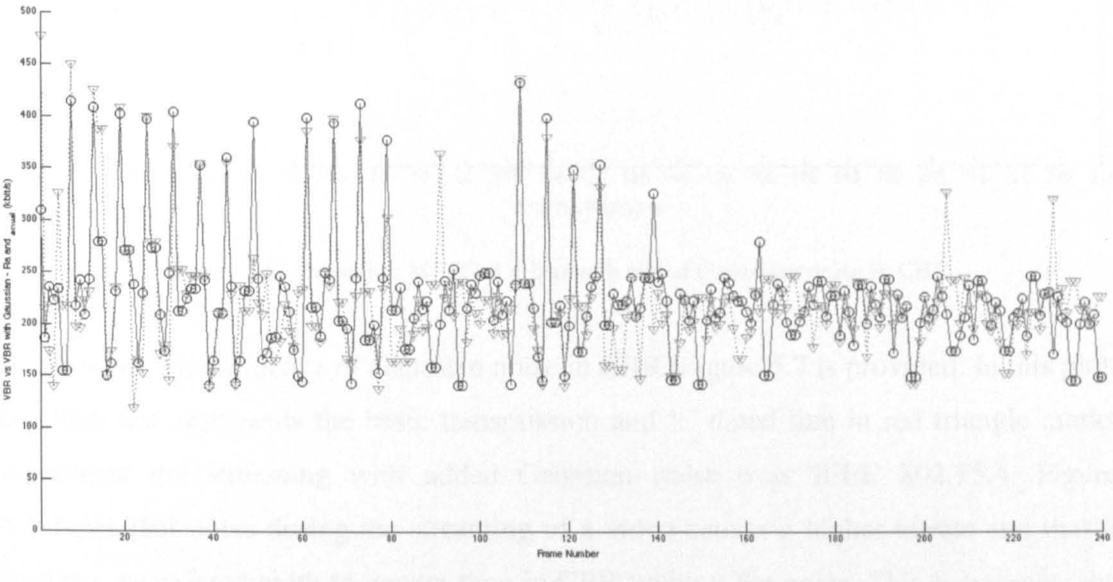


Figure 5.5: Basic streaming versus Added Gaussian noise in VBR

5.2.2 Video transmission with presence of noise using CBR

Use of CBR has proven that it has a substantial influence on controlling the data bitstream that needs to be transmitted within the ZigBee bitrate limits. This is shown in Figure 5.6.

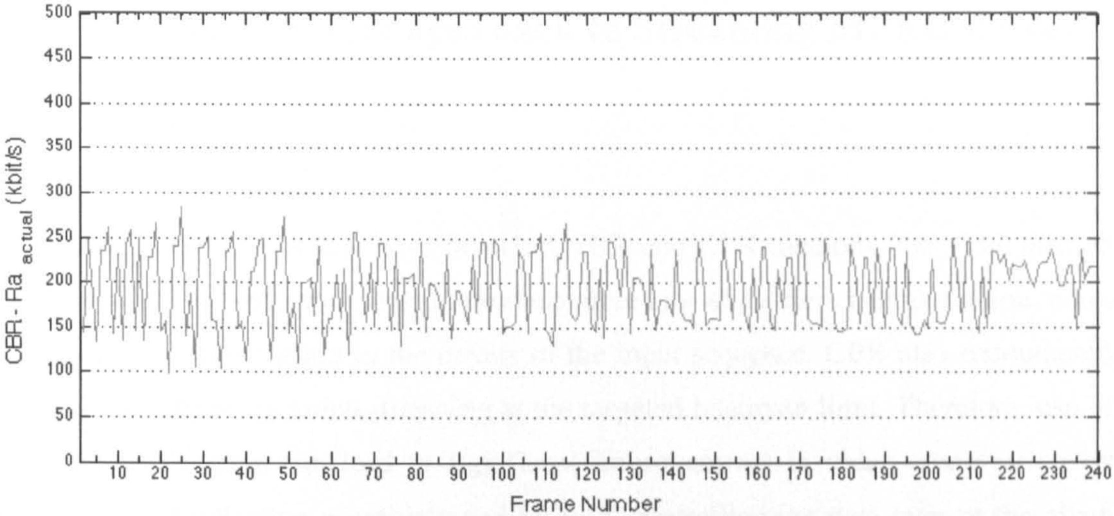


Figure 5.6: Streaming MPEG-4 video with added Gaussians noise in CBR

To compare the influence of Gaussian noise in CBR, Figure 5.7 is provided. In this plot, the blue line represents the basic transmission and ':' dated line in red triangle marks represents the streaming with added Gaussian noise over IEEE 802.15.4. Figure 5.7shows that noise during the streaming of a video causes a higher bitrate and that it requires more bandwidth to stream than in CBR without the noise. This is because, the encoder imprecisions the added noise as a high-frequency component in the motion compensated residual, and encodes this along with the desired residual data. This uses up more energy in the motion compensated residual, which leads to more bits being required to signal.

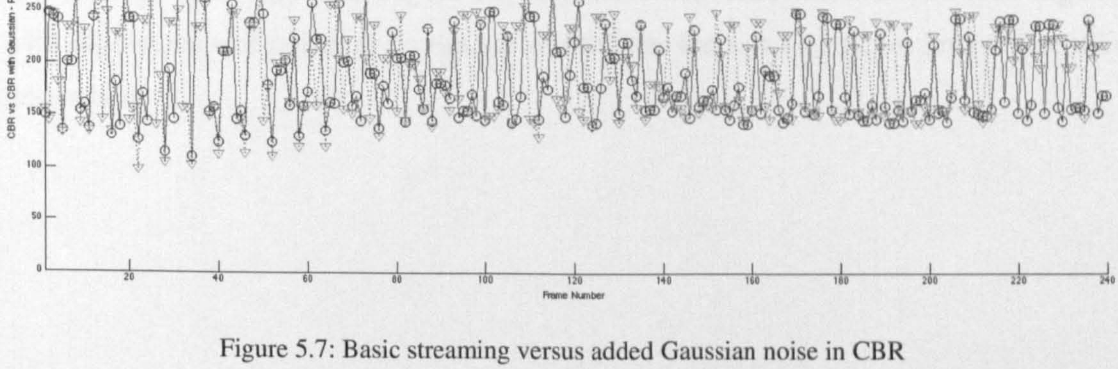


Figure 5.7: Basic streaming versus added Gaussian noise in CBR

5.3 Intelligent System approach to Streaming MPEG-4 over ZigBee

In previous experiments that have involved VBR and CBR methods, the VBR method demonstrated that an encoder would produce more bits when there is high motion, noise accrued, or a change made in the details of the input sequence. CBR also encountered difficulty in achieving video streaming at the targeted bitstream limit. Therefore, use of an intelligent system can lead to significant improvements in video compression by varying the quantization adaptively and by fully controlling the data rates at the given target. In order to do this, the particle swarm optimization applied in the CODEC and CBR model used as a framework for the proposed model (Figure 5.8).

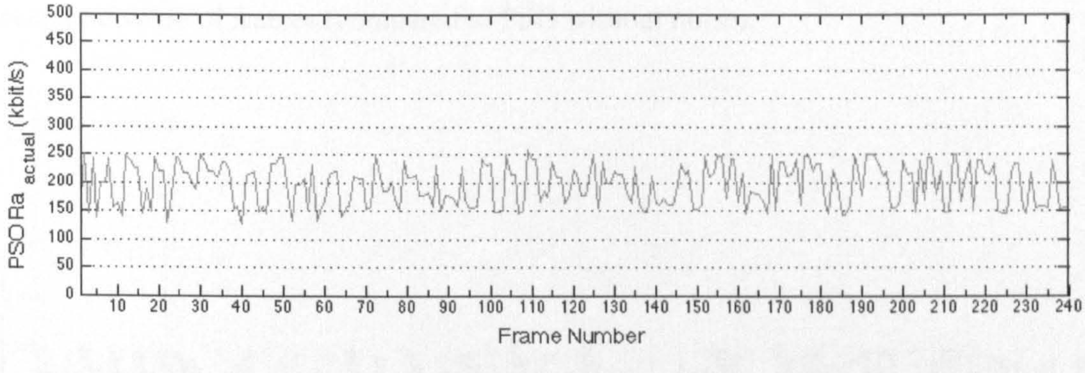


Figure 5.8: The adaptive quantization model with use of PSO

Figure 5.8 represents the usage of the bandwidth during the transmission when adaptive quantization is applied and scalar rate control achieved. This has the best results as it has the smallest bandwidth usage. The data rates have been controlled at a maximum of 250kbps. Using particle swarm optimization in this research-purposed algorithm has made this progress possible. Figure 5.9 represents the video transmission over ZigBee with the presence of noise, again the use of PSO has led to a significant improvement in rate control.

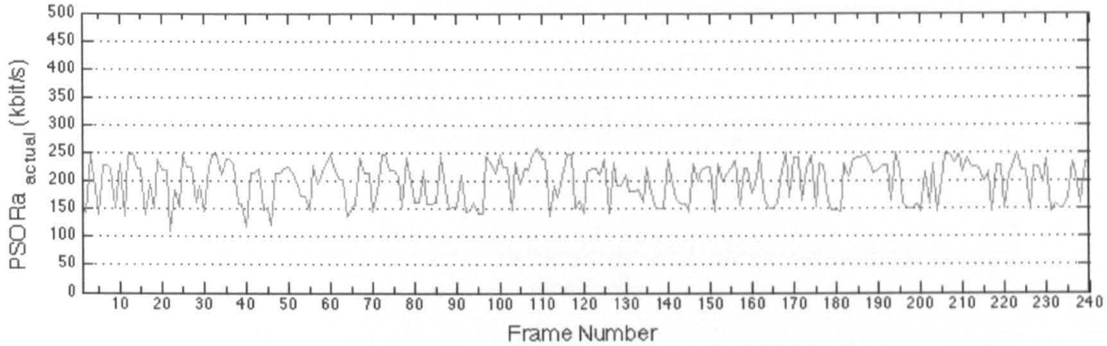


Figure 5.9: Video transmissions over ZigBee with the presence of noise using PSO

To review the PSO and PSO with noise trend, a plot provided in Figure 5.10. The plot shows little difference when streaming a video file with and without adding noise when using adaptive quantization. Figure 5.10 show that PSO and PSO with noise have a similar pattern. Both experience no data loss; however, PSO with noise has the higher mean rate value of frames (compared to PSO without noise).

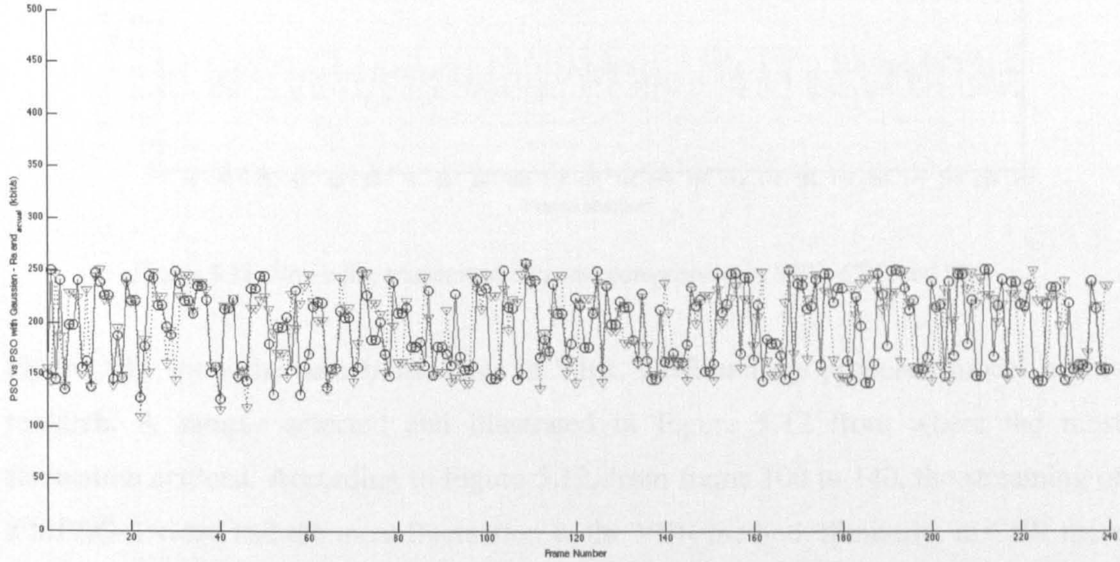


Figure 5.10: Video transmission over ZigBee with presence of noise and without noise using PSO

Figure 5.11 illustrates the VBR, CBR and PSO model plots together, and shows an overview of the simulation results for each model. These results will be discussed in detail. VBR and CBR models perform significantly worse (compared to PSO) at a lower bitrate of 250kbps and their rate of frame is even more distorted when noise is accumulated or when they are encoding the more complex video sequences.

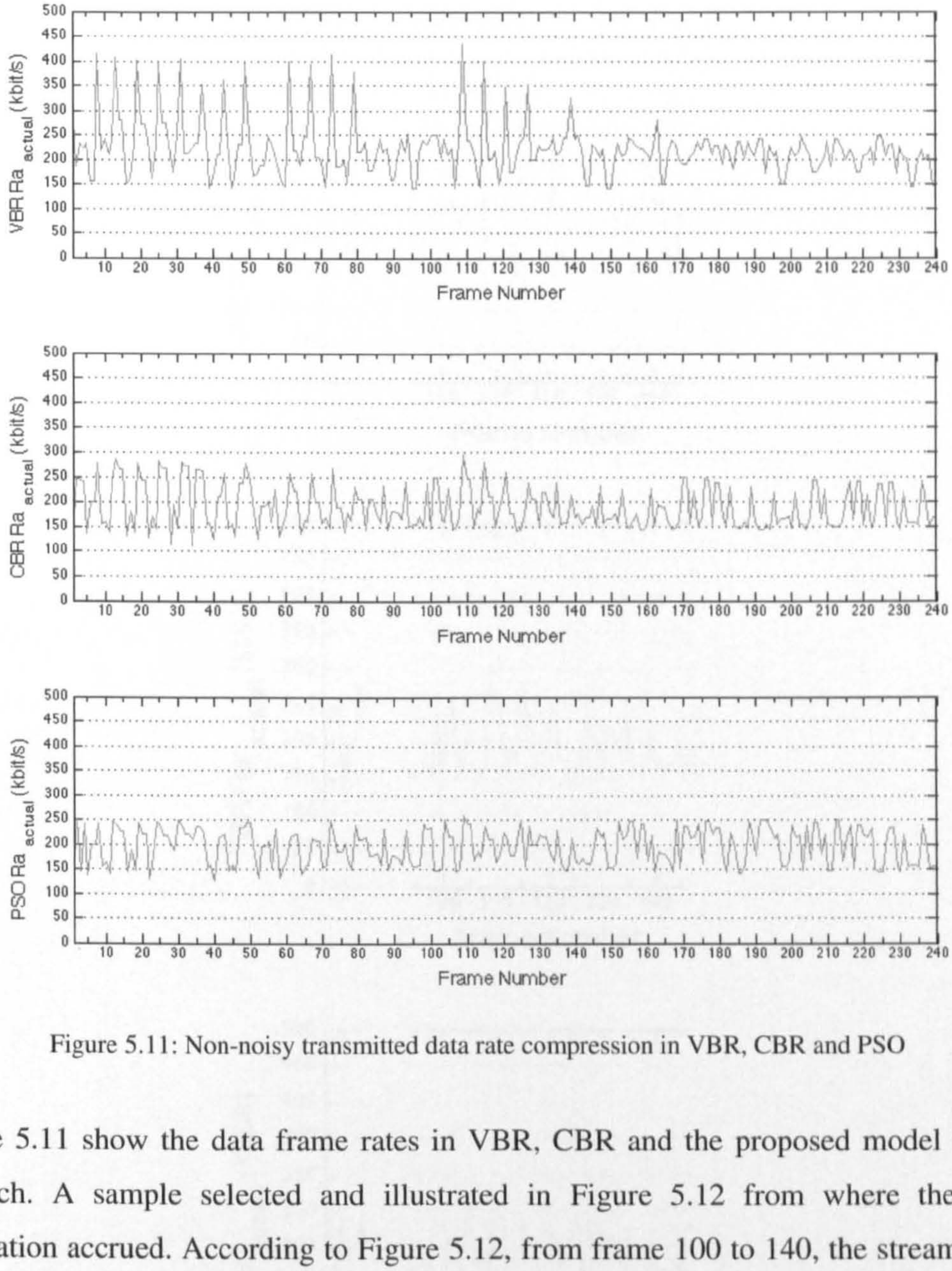


Figure 5.11: Non-noisy transmitted data rate compression in VBR, CBR and PSO

Figure 5.11 show the data frame rates in VBR, CBR and the proposed model in this research. A sample selected and illustrated in Figure 5.12 from where the most fluctuation accrued. According to Figure 5.12, from frame 100 to 140, the streaming of a MPEG-4 video had the most fluctuation in the VBR method. Similarly, in CBR there was a data loss, however, in the PSO method the data rate has been controlled to remain in the target bitrate. In comparison to the PSO adaptive model, neither VBR nor CBR are satisfying models for streaming over the IEEE 802.15.4 standard target rate.

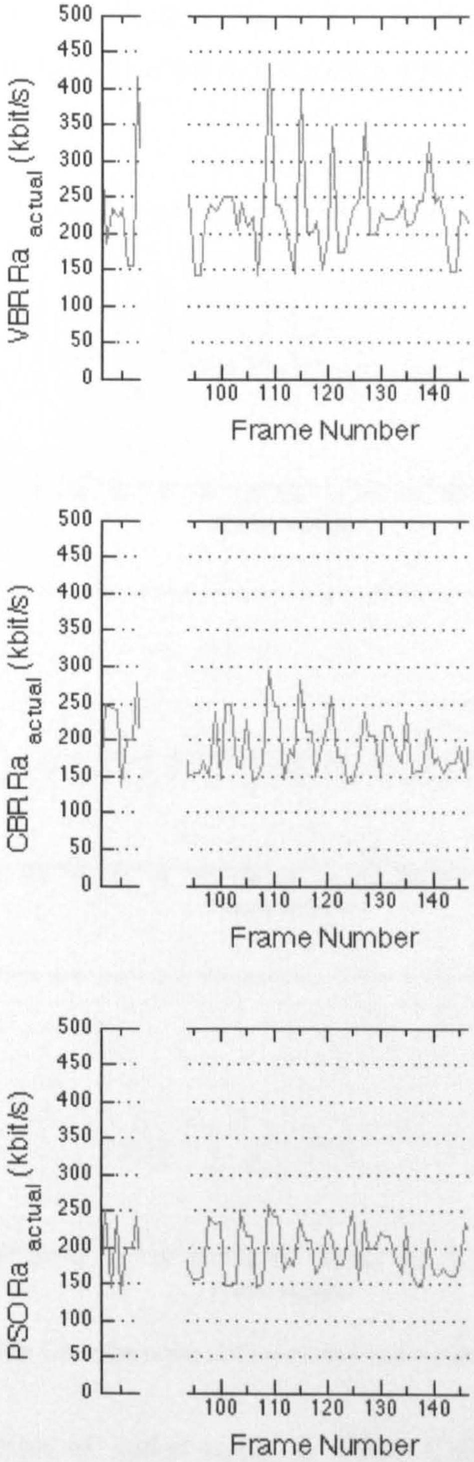


Figure 5.12: Selected range of fluctuated frames from 100 to 140.

Figure 5.13 shows a comparison between all the methods that have been used to stream MPEG-4 video with added Gaussian noise over the ZigBee wireless network in simulation. The VBR, CBR, and PSO with Gaussian noise are compared whilst streaming MPEG-4 video over ZigBee. Figure 5.13 illustrates that frame numbers in both VBR and CBR have been lost, but in PSO, even with Gaussian noise, there is still no data loss.

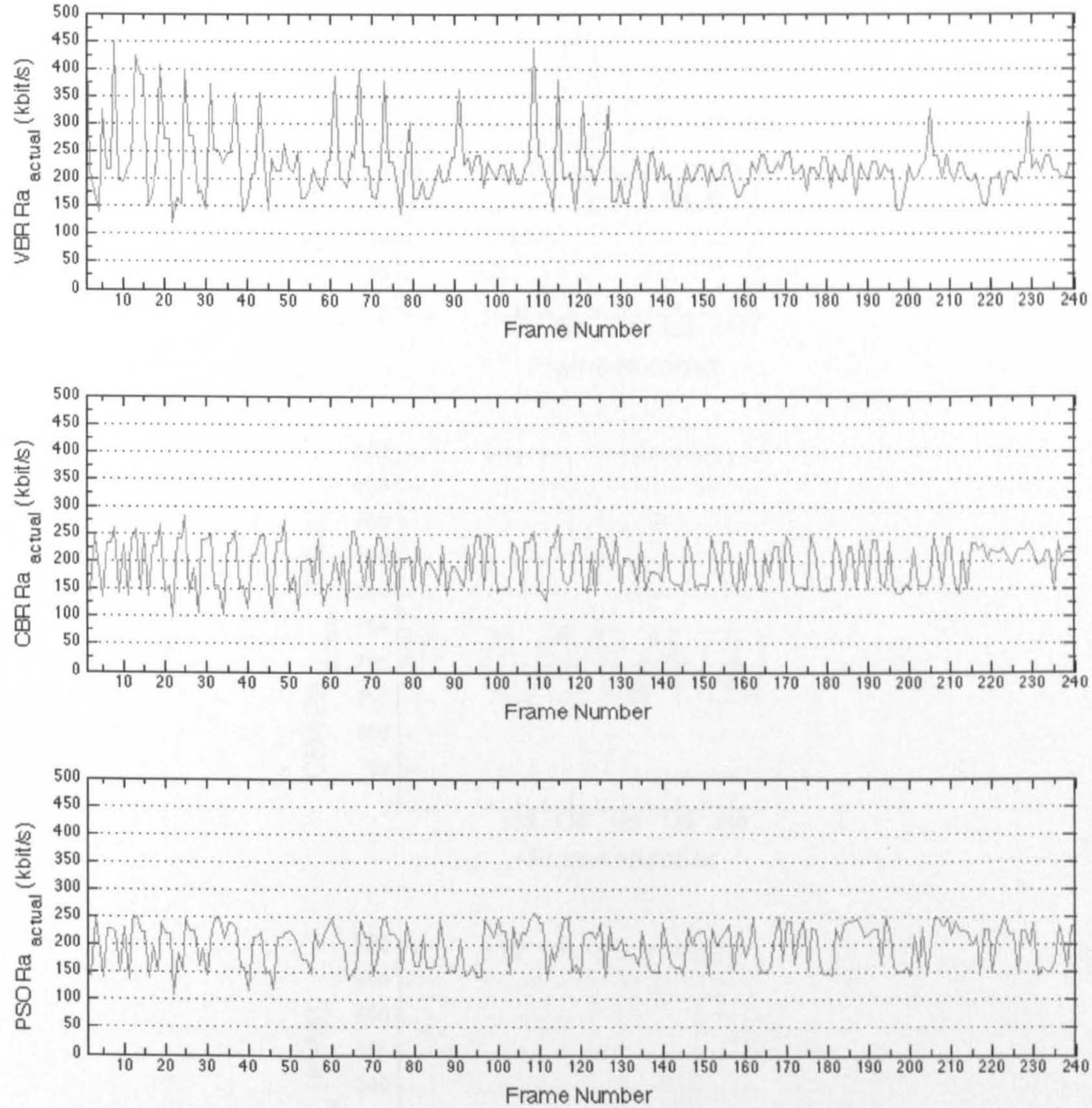


Figure 5.13: Transmitted with Gaussian noise and noisy data rate compression in VBR, CBR and PSO

Following the comparison of added noise in VBR, CBR and PSO methods, the fluctuation data from frame 100 to 140 are selected to compare. According to Figure 5.14, the streaming of the MPEG-4 video had the most fluctuation in the VBR. Similarly in CBR, there was a data loss. However, in the PSO method the data rate has been controlled to remain in our target bitrate. As expected, VBR and CBR in comparison to the PSO adaptive model are not satisfying models for streaming over the IEEE 802.15.4 standard target rate.

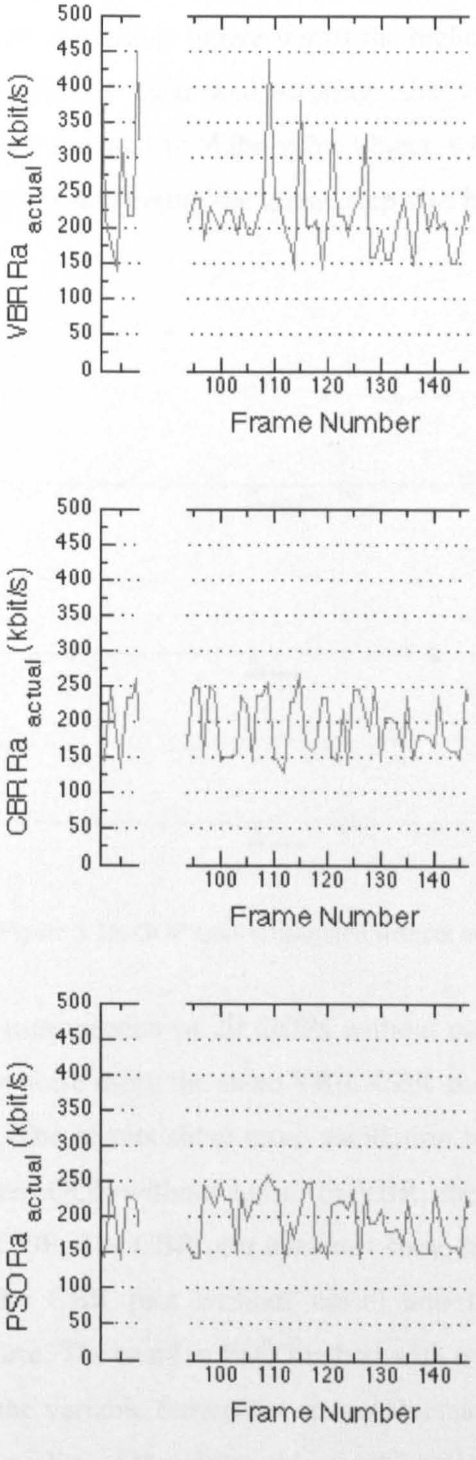


Figure 5.14: Selected range of fluctuated frames from 100 to 140 with Gaussian noise.

Figure 5.15 shows the comparison of 20 GOPs transmitted in simulation without the Gaussian noise using VBR, CBR and PSO models. The results show VBR's variable bitrate decreasing steadily from the eighth GOP. At the same time, the CBR plot indicates some fluctuation of variable bitrate whilst the highest rate is in the third GOP. The trend in the PSO plot is clear and striking; the variable bitrate has strong fluctuation. This is because the quality of the video object, which can cause more bitrate in a large complex bitrate, is decreasing the queue stepwise in favor of the target bitrate limit.

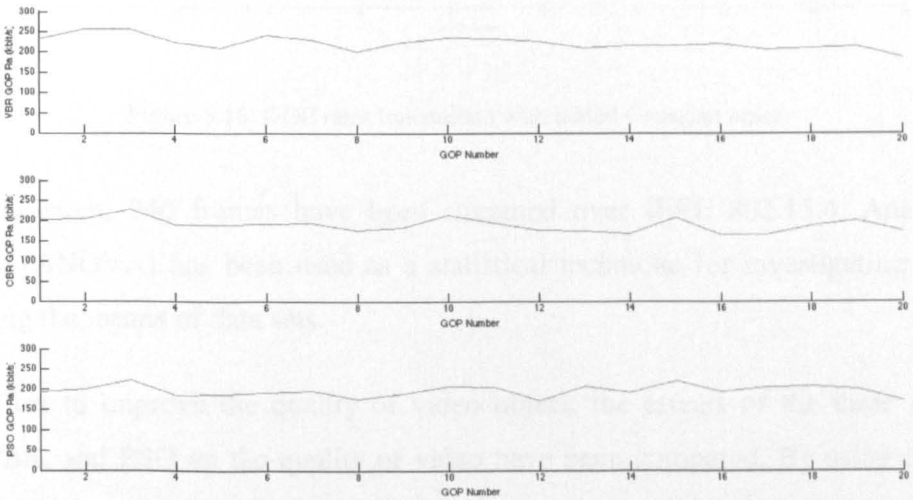


Figure 5.15: GOP rates transmitted without noise

Following on from the transmission of 20 GOPs without noise, the simulation carried out with added Gaussian noise using the same VBR, CBR and PSO models is compared together in Figure 5.16. The results show more vacillation in the variable bitrate in all three of the methods than GOP without noise. In VBR, the variable bitrate decreases steadily from the tenth GOP. The CBR plot indicates some more fluctuation of variable bitrate (compared to the CBR plot without noise) and the highest rate is in the nineteenth group of picture. The trend in PSO method with noise is the same as the PSO method without noise; the variable bitrate has strong fluctuation. The reason for this is the same as before, the quality of the video object, which causes more bitrate in a large complex bitrate, is decreasing the queue stepwise in favor of maintaining the target bitrate limit.

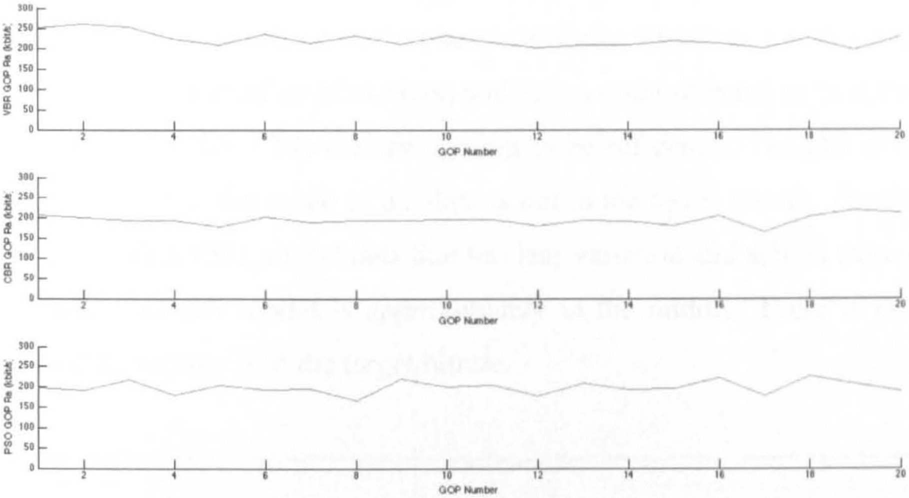


Figure 5.16: GOP rates transmitted with added Gaussian noise

In this research, 240 frames have been streamed over IEEE 802.15.4. Analysis of Variance (ANOVA) has been used as a statistical technique for investigating data by comparing the means of data sets.

In an effort to improve the quality of video object, the effects of the three methods VBR, CBR, and PSO on the quality of video have been compared. By using the One-Way ANOVA test, the significant difference can be find as if 'a' exists among the mean distortions obtained using these three methods.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	1.48469e+11	2	7.42343e+10	34.48	5.03859e-15
Error	1.54365e+12	717	2.15293e+09		
Total	1.69212e+12	719			

Figure 5.17: ANOVA data set in streaming VBR, CBR and PSO without noise

In Figure 5.17 an ANOVA test of the null hypothesis $H_0: \mu_1 = \mu_2 = \mu_3$ is performed by comparing the observed F-value 34.48. At a 0.05 level of significance, with the p-value of 5.38e-15, this tests the overall model to determine if there is a difference in means between methods. In this case, since the p-value is small, therefore it can be determined there is a significant difference between the three mean models.

Figure 5.18 shows the boxplots of the simulated models side-by-side. The median in the VBR boxplot is in the middle of the rectangle and the whiskers are about the same length. There are a large number of outliers, which is a point of concern as it shows that the data varies widely. The CBR median appears to be off-center. The plot contains no outside values; however, the range of the data is not in the target bitrate. The third plot, which is the plot using PSO, shows data that has less variation and spread than the other plots. The median of this model is approximately in the middle. There is no outside value and our data's range is in the target bitrate.

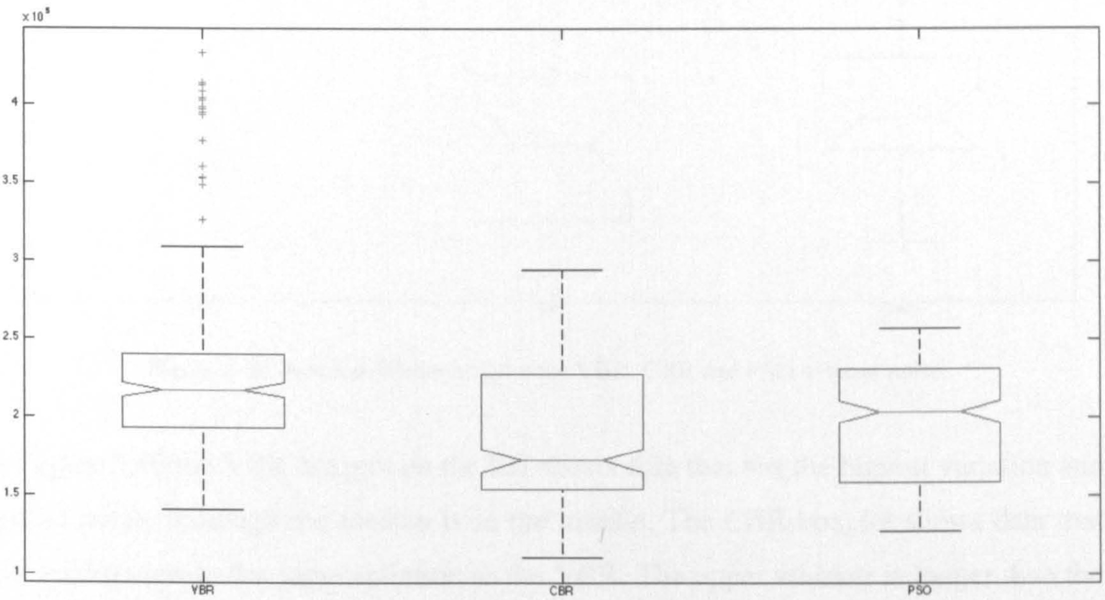


Figure 5.18: Boxplot of VBR, CBR and PSO models side-by-side

Kruskal-Wallis is a nonparametric method, which compares the medians of our three models frame rates of 20 GOPs streamed in the simulation. The result of this method is shown in Figure 5.19. Our factual null hypothesis is to test if the groups of pictures in the VBR, CBR, and PSO have the same median or not.

Kruskal-Wallis ANOVA Table

Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	7932.4	2	3966.2	26.01	2.25145e-06
Error	10062.6	57	176.54		
Total	17995	59			

Figure 5.19: The Kruskal-Wallis test that compares the medians rates of VBR, CBR and PSO without noise

The Kruskal-Wallis test gives a meaningful result; the p-value in this test is near zero (2.25145e-06), which casts doubt on the null H and suggests that at least one model is significantly different from the others.

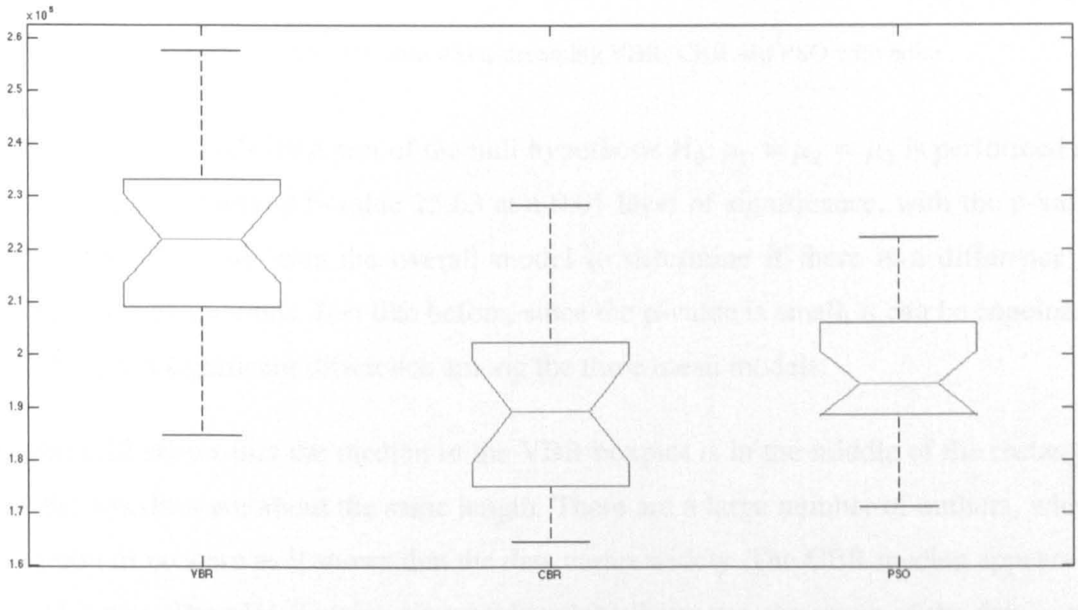


Figure 5.20: Kruskal-Wallis boxplot for VBR, CBR and PSO without noise

In Figure 5.20 the VBR boxplot on the left shows data that has the biggest variation and spread range, although the median is in the middle. The CBR boxplot shows data that has approximately the same variation as the VBR. The upper whisker is longer than the bottom one, which is a consequence of the single high data. The PSO boxplot data shows that the median is nearly in the middle, the whiskers are about the same length, and has the least variation and spread than the other plots.

5.3.1 Streaming in VBR, CBR and PSO methods in presents of noise

In a similar way to the previous simulations and ANOVA test for non-noise video streaming, in this section all three methods with Gaussian noise have been compared.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	1.14557e+11	2	5.72785e+10	25.63	1.77543e-11
Error	1.60243e+12	717	2.2349e+09		
Total	1.71698e+12	719			

Figure 5.21: ANOVA data when streaming VBR, CBR and PSO with noise

In Figure 5.21 an ANOVA test of the null hypothesis $H_0: \mu_1 = \mu_2 = \mu_3$ is performed by comparing the observed F-value 25.63 at a 0.05 level of significance, with the p-value of 1.77543e-11; this tests the overall model to determine if there is a difference in means between methods. Just like before, since the p-value is small, it can be concluded that there is a significant difference among the three mean models.

Figure 5.22 shows that the median in the VBR boxplot is in the middle of the rectangle and the whiskers are about the same length. There are a large number of outliers, which is a point of concern as it shows that the data varies widely. The CBR median appears to be off-center. The plot contains no outside values; however, the range of the data is not in our target bitrate. The third plot, which is the PSO plot, shows data that has less variation and spread than the other plots. The median of this model is approximately in the middle. There is no outside value and our data's range is in our target bitrate, just as before.

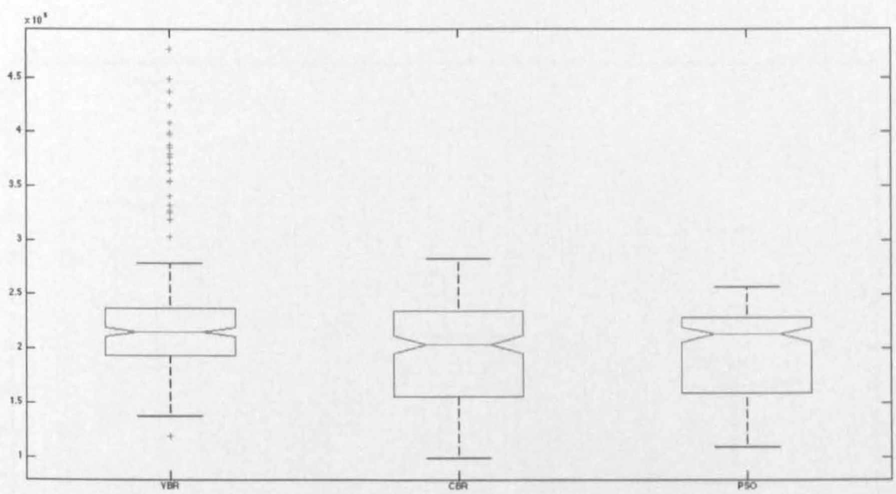


Figure 5.22: Boxplot of VBR, CBR and PSO models side-by-side with noise

Similarly, to streaming MPEG-4 video without the white Gaussian noise, the nonparametric Kruskal-Wallis test is run again comparing the medians of our three model rates of 20 GOPs with Gaussian noise. The small p-value in this test (6.08009e-06), as represented in Figure 5.23, shows that it can reject the null H. This shows that there is a significant difference between our medians, and at least one of the methods is different from the other methods.

Kruskal-Wallis ANOVA Table

Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	7326.4	2	3663.2	24.02	6.08009e-06
Error	10668.6	57	187.17		
Total	17995	59			

Figure 5.23: Kruskal-Wallis test, which compares the medians rates of VBR, CBR and PSO with noise

According to Figure 5.24 a different level of medians is illustrated. The VBR boxplot in Figure 5.24 shows data that has the biggest variation. The upper whisker is longer than the bottom one, which is illustrative of the single high value. The median of this plot appears to be off-center. The second boxplot shows data that is significantly downwardly skewed. The median of this plot is closer to the bottom of the rectangle than to the top. Although the whiskers are the same length, it shows that the CBR model has outlier data. The third boxplot presented is the PSO model where the whiskers are about the same length, the median is nearly in the middle and all the data is in our target bitrate.

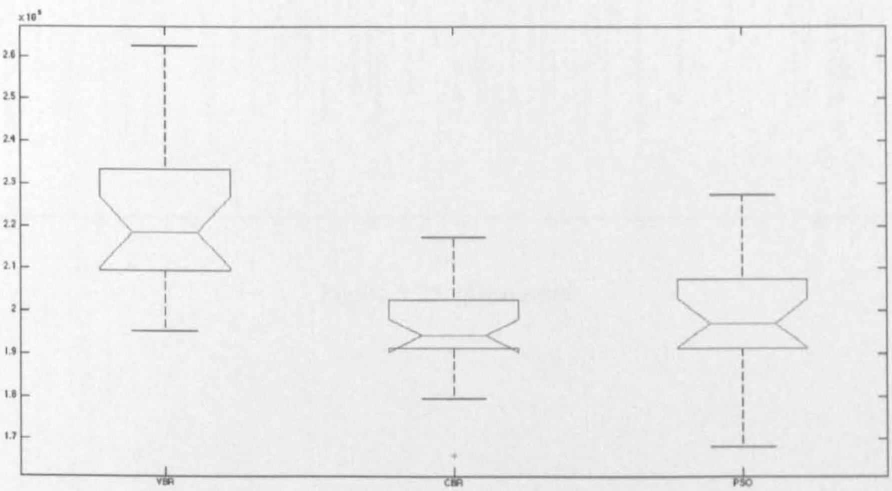


Figure 5.24: Kruskal-Wallis boxplot for VBR, CBR and PSO with noise

In the following sections, selected range of frames presented in Figure 5.12 from 100 to 140 is compared to find out more about the quality of the video sequences and the amount of data loss in VBR and CBR models with more sophisticated measurement methods. In addition, how the anticipated algorithm in this research can help to respond with a better quality of VOPs and with successful streaming at the targeted rate explained.

5.3.2 gBest

The PSO simulation results are studied from several perspectives of gBest, Best Fitness and calculation of desired Q-steps for each GOP. As explained in chapter 3, the gBest of a particle in this simulation is defined through the frames and their bitrates. In the simulation, every particle attracted to the best solution and each epoch was found, and the fitness of each particle evaluated according to the fitness function. Figure 5.25 shows the result of gBests for the transmitted frames from 1 to 240.

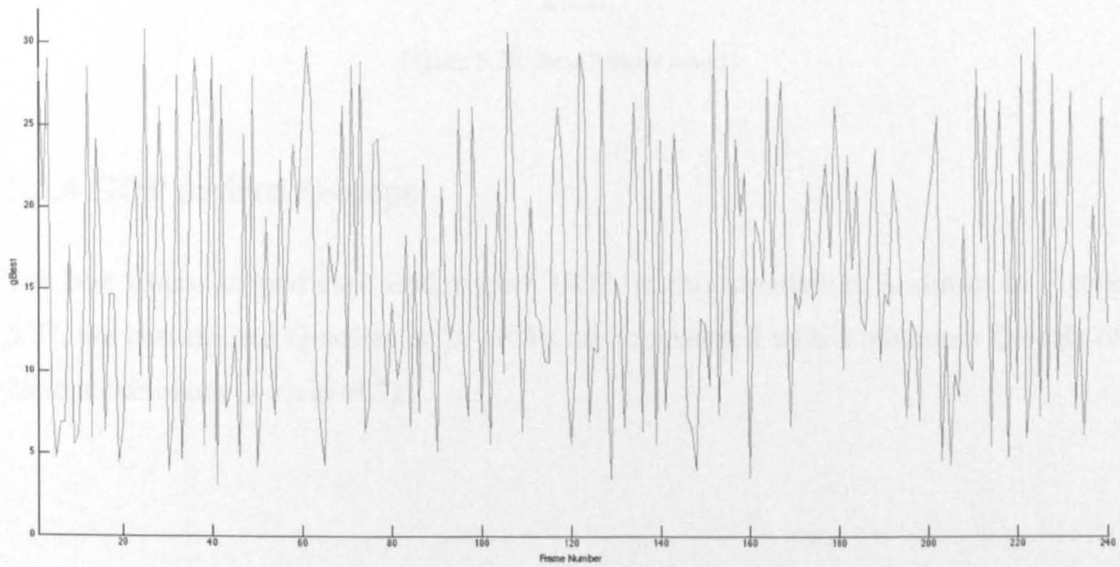


Figure 5.25: gBest result

5.3.3 Best Fitness

The best fitness of a particle used for each frame q-scale has a minimum of 9 to a maximum of 31. The best Fitness value for frames from 1 to 240 is represented in Figure 5.26.

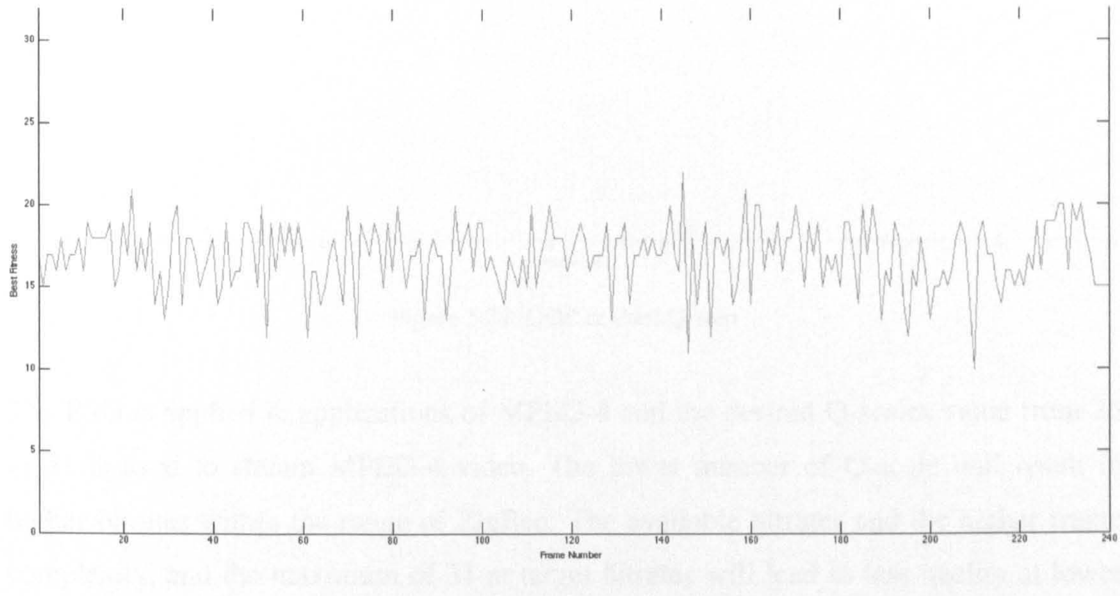


Figure 5.26: Best Fitness result

5.3.4 GOP desired Q-steps

The best fitness of particles used at each GOP, in this simulation, is shown in Figure 5.27, the best-desired Q-scales or 20 GOPs are represented with a minimum Q-scale of 26 to a maximum Q-scale of 31.

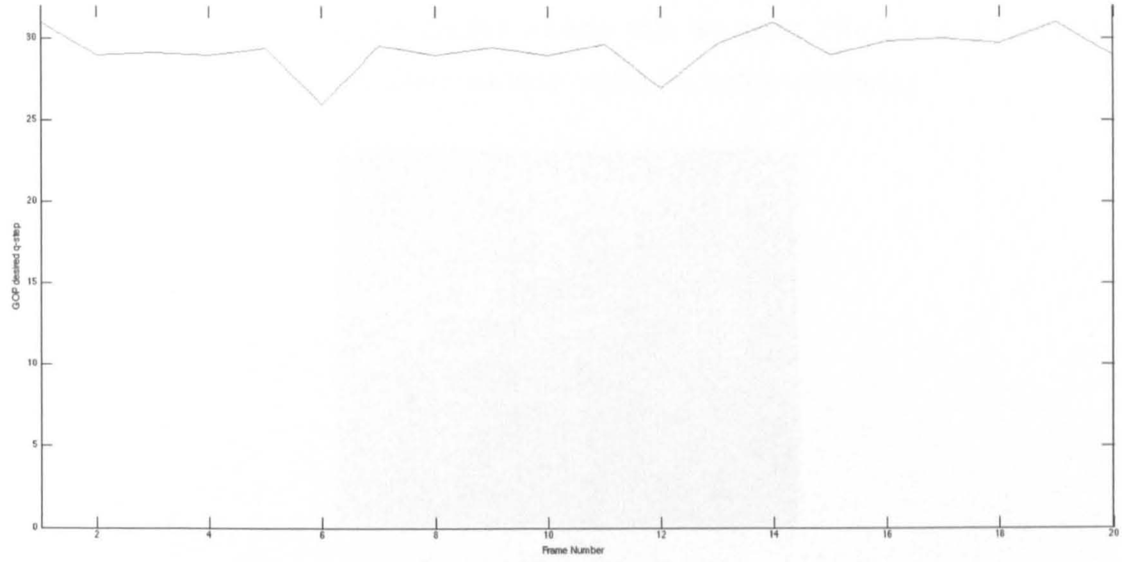


Figure 5.27: GOP desired Q-step

The PSO is applied in applications of MPEG-4 and the desired Q-scales value from 26 to 31 is used to stream MPEG-4 video. The lower number of Q-scale will result in higher bitrates within the range of ZigBee. The available bitrates and the higher frame complexity, and the maximum of 31 at target bitrates will lead to less quality at lower frame complexity.

5.4 Cost Functions

A number of different cost functions are developed and compared to address and measure the results of simulations in VBR, CBR and the developed algorithms in this research. The results of the methods that are used in these simulations (VBR, CBR and PSO) compared. Cost function examinations are done for a single random frame (#115) and the furthermost fluctuation experience from frame 100 to frame 140 is shown in Figure 5.12 .

5.4.1 Difference of Gaussians.

The Gaussian noise has been added into the video and has been saved as a separate video file. Figure 5.28 shows a sample frame taken before adding noise and streaming the video over IEEE 802.15.4 ZigBee sample that has been taken from an identical sequence to Figure 5.29 after noise has been added but before streaming.

Figure 5.28: Original frame

The basic video file and the noisy file have both been used as a source in VBR, CBR and the proposed method to stream MPEG-4 over ZigBee. In order to compare and measure the object quality and the differences between these samples the Difference of Gaussians (DoG) algorithm used, as defined from Figure 5.28 to Figure 5.29 [135], [134].

Figure 5.29: Frame with Gaussian white noise

The following equations from Equation 5.1 to 5.5, explains the example of an image which is first smoothed by convolution with Gaussian kernel of certain width σ_1 [135]

$$G_{\sigma 1}(x,y) = \frac{1}{\sqrt{2_{\pi\sigma_1^2}}} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \tag{5.1}$$

to get

$$g1\,(x,y) = G_{\sigma 1}(x,y) \times f(x,y) \tag{5.2}$$

with a different width , a second smoothed image can be obtained:

$$g2\,(x,y) = G_{\sigma 2}(x,y) \times f(x,y) \tag{5.3}$$

Than the difference of these two Gaussian smoothed images, DoG, can be used to detect edges in the image.

$$\begin{aligned} g1(x,y) - g2(x,y) &= G_{\sigma 1} \times f(x,y) - G_{\sigma 2} \times f(x,y) \\ &= (G_{\sigma 1} - G_{\sigma 2}) \times f(x,y) = DoG \times f(x,y) \end{aligned} \tag{5.4}$$

The DoG as an operator or convolution kernel is defined as

$$DoG \triangleq G_{\sigma 1} - G_{\sigma 2} = \frac{1}{\sqrt{2\pi}}\left[\frac{1}{\sigma_1}e^{-(x^2+y^2)/2\sigma_1^2} - \frac{1}{\sigma_2}e^{-(x^2+y^2)/2\sigma_2^2}\right] \tag{5.5}$$

The result of DOG is sigma 1 = 8 and sigma 2 = 0.5000. Furthermore, the Peak Signal to Noise Ratio (PSNR), an objective quality measure that is commonly used to find the errors and the quality of compressed and decompressed video images [61].

5.4.2 Getting the Universal Index Quality

Universal Index Quality (UQI) method used for the simple cross-distortion test [129], [130]. The proposed quality index is defined in Equations 5.6 and 5.7. The x is the value of the original image the y is the value of the test image and Q is quality index [131].

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]}, \tag{5.6}$$

where $\bar{x}$ and $\bar{y}$ are the Mean value of original x and test image y. The σ_x^2 represents the Variance of image x and σ_y^2 represents the variance of image$y$. Cross Variance between x and y is represented as σ_{xy}.

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i,$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2.$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}). \tag{5.7}$$

The Figure 5.30 is a sampling frame from the original source.

Figure 5.30: Original frame (source)

The source frame compares Figure 5.30 with the one that Gaussian white noise added shown in Figure 5.25.

Figure 5.31: Noisy image using Gaussian noise with sigma = 0.001 and image quality index = 0.73438.

The quality index of the noisy image/frame is calculated in the simulation in accordance to the formula and it is 0.73438 at the sigma level of 0.001dB as represented in Figure 5.31. The one to one evaluation of the original frame versus the noisy frame is represented below:

Universal frame Quality Index = 0.73 dB

Pearson Correlation Coefficient (original frame vs. Gaussian noisy frame) = 24237.16 dB

Pearson Correlation Coefficient (original frame vs. original frame) = 25343.00 dB

5.4.3 Peak Signal-to-noise-Ratio

PSNR was explained previously in chapters two and four, the signal used in the simulation is the original source of data, and the noise is the error introduced by compression after VBR, CBR and PSO transmissions with white Gaussian Noise. When comparing compression CODECs, PSNR is a way of telling approximately the human perception of the reconstructed quality. For this reason, a reconstruction may sometimes appear to be closer to the original than others do. PSNR is formulated in Equation 5.8 [61]:

$$PSNR = 10\log\frac{(2^n-1)^2}{MSE} = 10\log\frac{255^2}{MSE} \tag{5.8}$$

A random frame sample (#115) taken from the most fluctuated range of data, from frame 100 to frame 140 as is shown in Figure 5.12.

Figure 5.32 is the sample that has been taken for this evaluation. It is comparing the PSNR of VBR, CBR and PSO methods. The result shows that the PSNR result or the sample frame in VBR after transmission is 20.9351dB.

Figure 5.32: Sample frame transmitted in VBR

The PSNR result compared to the source in CBR using the same frame number is shown in Figure 5.33 and is 17.8369dB.

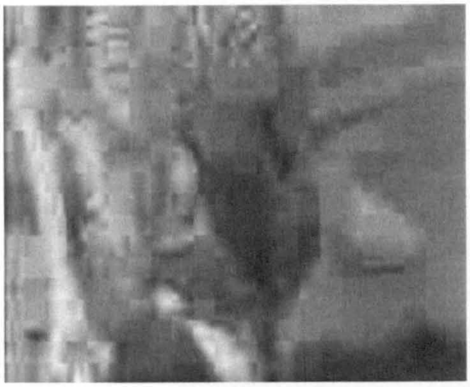


Figure 5.33: Sample frame transmitted in CBR

The PSNR result on the source of adaptive quantization using PSO on the same frame number is shown in Figure 5.34. In this test, PSO conducts the optimum result in between VBR and CBR with value of 20.8851dB; that is 4.37% less than VBR. Even though a small percentage of quality of the video object is lost, the PSO method compensates for this by remaining within the bandwidth target rate limit of IEEE 802.15. Like with the VBR, PSO has a greater value than CBR.

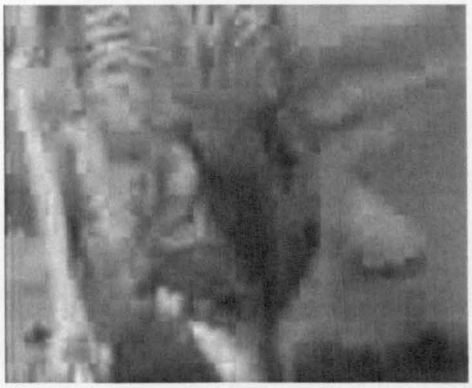


Figure 5.34: Sample frame transmitted in PSO

The frame rate of the sample frame (frame #115) is compared to evaluate the PSNR to transmission rate and the encoding quality. In VBR it is 398kbps, in CBR the frame rate is 279kbps and in PSO it is 237kbps.

The results in Table 5.1 show that VBR has a better quality. However, because of large frame rate size the frame is not suitable for the given rate of 250kbps in ZigBee. CBR, with the constant value of Q-step, also has a large frame rate but it has very low quality. Therefore, the results prove that with use of PSO and introducing an adaptive quantization it can achieve a good balance in higher PSNR than other commonly used methods, whilst improving the quality of the image during the encoding, and adaptively managing the best frame rate at the target bitrate. In our simulation, when the white Gaussian noise is added the high PSNR indicates that the reconstruction is of a higher quality.

Method	Frame rate	PSNR
VBR with Gaussian noise	379kbps	20.9351 dB
CBR with Gaussian noise	266kbps	17.8369 dB
PSO with Gaussian noise	209kbps	20.8851 dB

Table 5.1: PSNR and frame rates transmitted with white Gaussian noise

Table 5.2 shows the PSNR rates after transmission in the absence of noise in VBR, CBR and PSO methods. However, there is a resulting error, as CBR has the highest PSNR but the worst quality and the largest frame rate; in this case, the PSNR is undefined.

Method	Frame rate	PSNR
VBR	398kbps	20.4779 dB
CBR	237kbps	20.8043 dB
PSO	237kbps	20.7484 dB

Table 5.2 PSNR and frame rates without noise

Furthermore, for the purpose of comparison of VBR, CBR and the proposed PSO algorithm in this research a wide range of frames were fluctuated, as is described in Figure 5.12, and then examined for PSNR. Figure 5.35 shows the PSNR after streaming MPEG-4 over ZigBee in VBR:

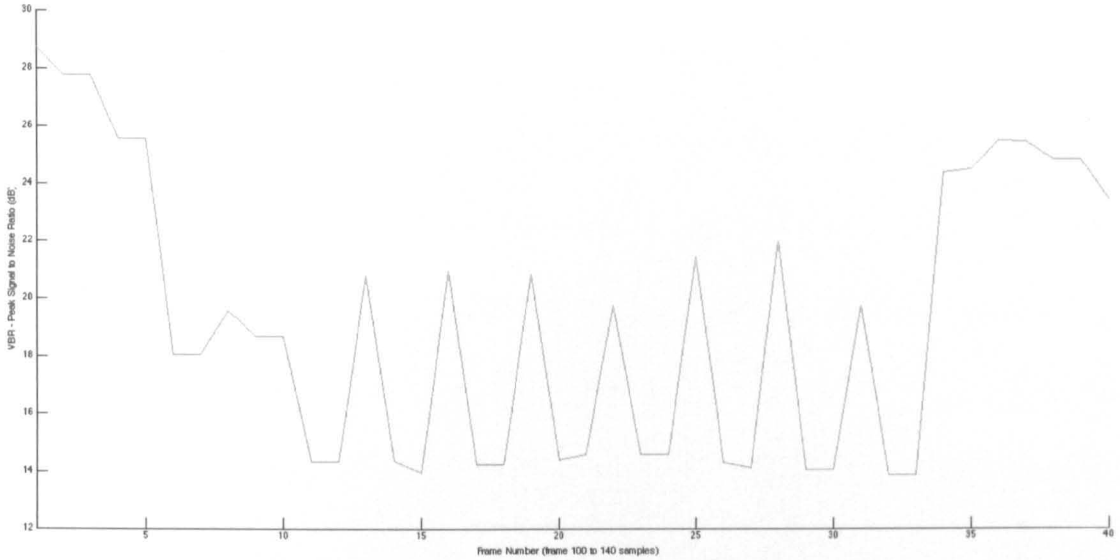


Figure 5.35: PSNR outcome after streaming over ZigBee using VBR model

The PSNR figures for all 40 frames are plotted frame-by-frame against the source of the MPEG-4 video before transmission. The recorded PSNR values are at a minimum of 13.84dB and a maximum of 28.73dB. A similar approach is used in CBR, which is shown in Figure 5.36

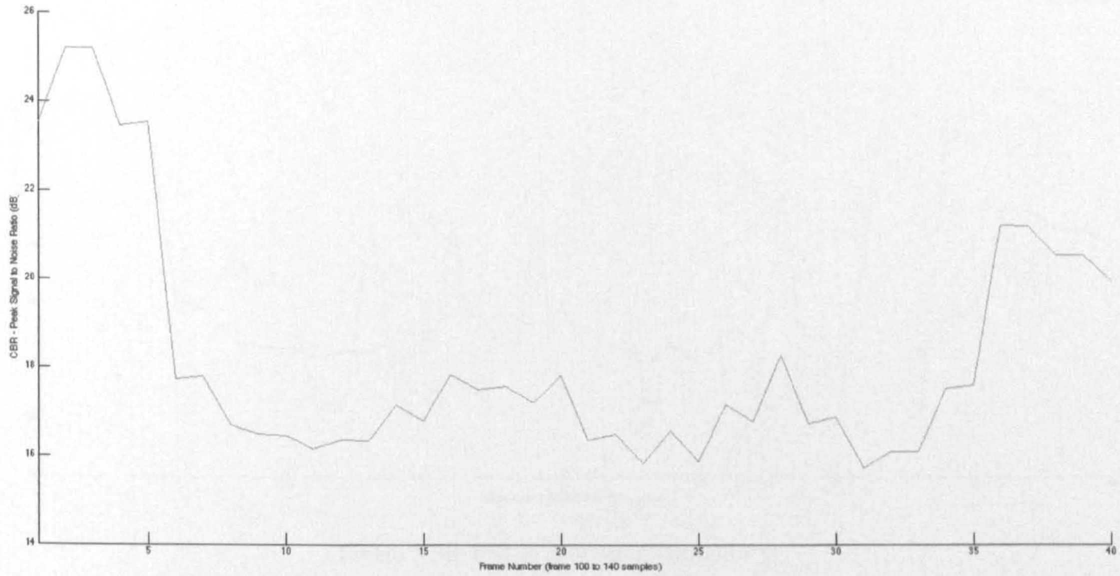


Figure 5.36: PSNR outcome after streaming over ZigBee using CBR model

The PSNR data after streaming in CBR presents a minimum PSNR of 15.72dB and a maximum PSNR of 25.18dB. Figure 5.37 shows the PSNR results when using PSO.

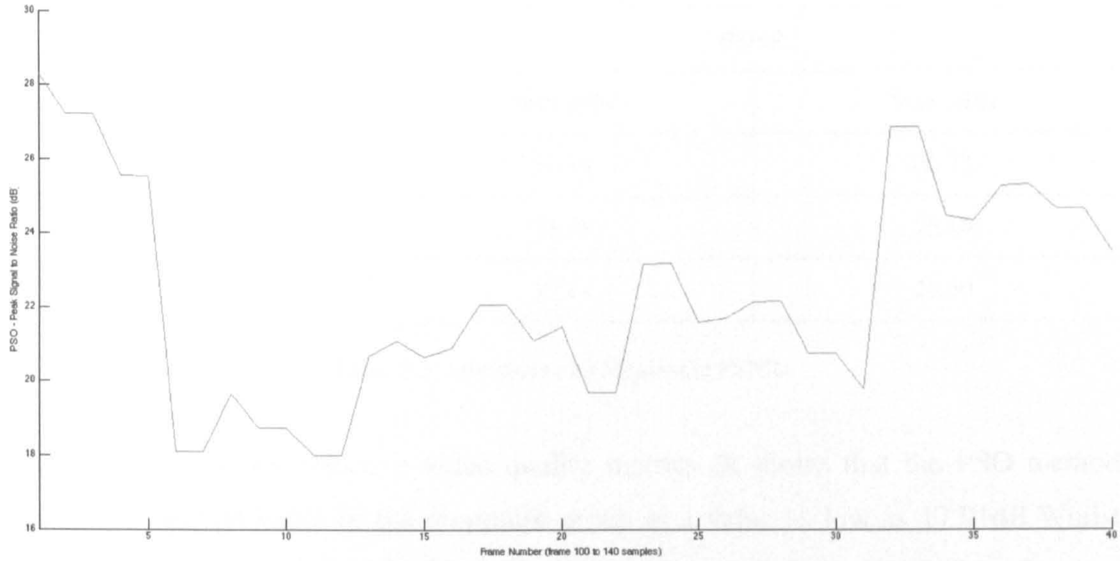


Figure 5.37: PSNR outcome after streaming over ZigBee using PSO

The PSNR results for the PSO algorithm present the minimum value of 17.91dB and a maximum value of 28.30dB. To compare the different PSNR results for each method, the PSNR results are grouped together and presented in Figure 5.38:

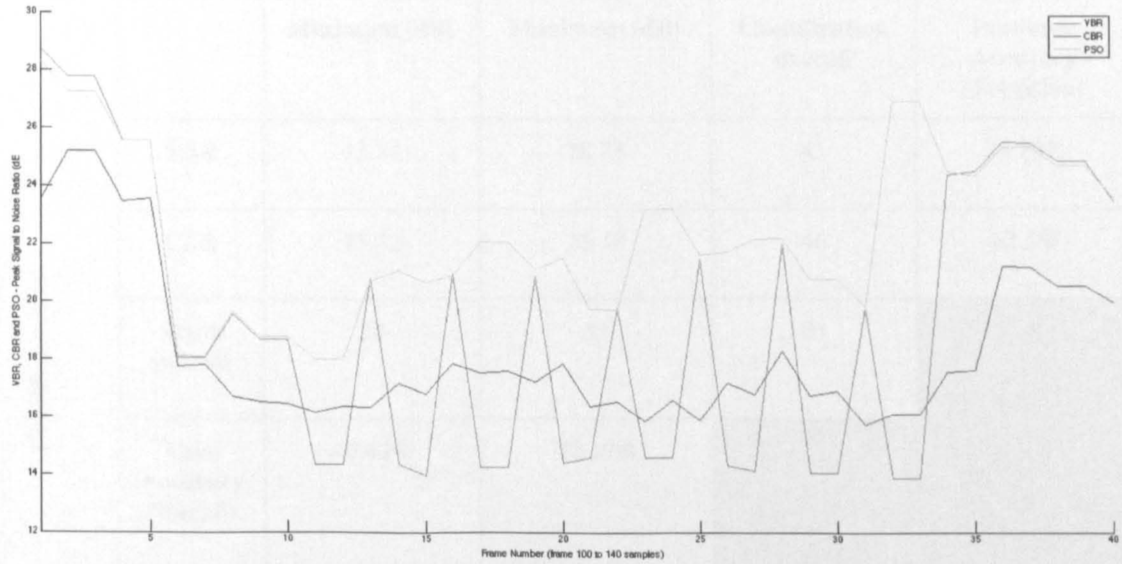


Figure 5.38: PSNR in VBR, CBR and PSO

The minimum and maximum numbers for each method is listed in Table 5.3.

Method	PSNR	
	Min (dB)	Max (dB)
VBR	13.84	28.73
CBR	15.72	25.18
PSO	17.91	28.30

Table 5.3: Minimum and Maximum PSNRs

The PSNR test is an objective video quality metrics. It shows that the PSO method carried the highest value in the minimum group at a value as low as 17.91dB.Whilst having an optimum value of 28.30 within the maximum group, which is less than the Maximum PSNR value of the VBR method and greater than the PSNR maximum value of the CBR. In order to be accurate about the differences a basic confusion matrix used shown in Table 5.4.

		Truth data (PSNR)			
Classifier results		Minimum (dB)	Maximum (dB)	Classification overall	Producer Accuracy (Precision)
	VBR	13.84	28.73	41	31.70%
	CBR	15.72	25.18	40	62.5%
	Truth overall	28	53	81	
	User Accuracy (Recall)	46.42%	47.17%		
	Overall accuracy	46.91%			

Table 5.4: Confusion matrix with VBR & CBR

		Truth data (PSNR)			
		Minimum (dB)	Maximum (dB)	Classification overall	Producer Accuracy (Precision)
Classifier results	VBR	13.84	28.73	41	31.70%
	PSO	17.91	28.30	45	62.22%
	Truth overall	30	56	86	
	User Accuracy (Recall)	43%	50%		
	Overall accuracy	47.67%			

Table 5.5: Confusion matrix with VBR & PSO

		Truth data (PSNR)			
		Minimum (dB)	Maximum (dB)	Classification overall	Producer Accuracy (Precision)
Classifier results	CBR	15.72	25.18	40	62.5%
	PSO	17.91	28.30	45	62.22%
	Truth overall	32	53	85	
	User Accuracy (Recall)	46.87%	52.83%		
	Overall accuracy	50.58%			

Table 5.6: Confusion matrix with CBR & PSO

According to the results shown in Figure 5.38 and in the Confusion matrix explaining the results of the tests carried out to find the PSNR; the PSO has a maximum PSNR of

52.83% and an overall accuracy of 50.58%.,therefore a higher PSNR value in this method indicates a higher quality image.

5.4.4 Root Mean Square Error

Root Mean Square Error (RMSE) is widely used to measure the differences between the predicted value and the actual values observed. The RMSE measures are 'dimensioned' in that it expresses average interpolator error in the units of the variable of interest. The RMSE is of special interest because it is the most often reported and misinterpreted of the three average-error statistics. Two procedures for interpolating average air temperature spatially are evaluated and compared to illustrate our approach to spatial cross-validation and the use of these dimensioned error measures [132]. In RMSE, formulated in Equation 5.9, the lower values are better as it is a negatively oriented measurement [132].

$$RMSE = \left[n^{-1} \sum_{i=1}^{n} |e_i|^2 \right]^{1/2} \tag{5.9}$$

Following the UIQ and PSNR tests, the RMSE for one to one evaluation of the original frame versus the noisy frame is 23.6497.

Table 5.7 shows the results of the VBR, CBR and PSO RMSE assessments.

Method	Frame rate (Frame #115)	RMSE
VBR with Gaussian noise	379kbps	51.0751
CBR with Gaussian noise	266kbps	36.8601
PSO with Gaussian noise	209kbps	23.6497

Table 5.7: RMSE single frame evaluations

To compare the differences in VBR and CBR with the algorithm developed in this research, the high-fluctuated frames from frame 100 to 140 are compared in Figure 5.39.

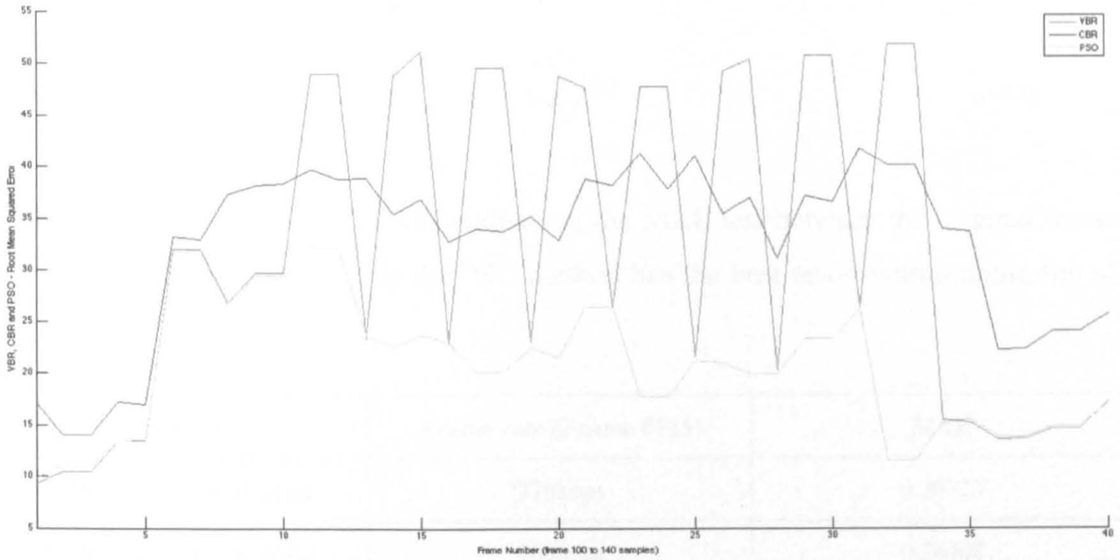


Figure 5.39: VBR, CBR and PSO RMSE

In addition to Figure 5.39, Table 5.8 presents the minimum and maximum value of the RMSE for all the methods.

Method	RMSE	
	Min	Max
VBR	9.33	51.77
CBR	14.03	41.72
PSO	9.80	32.13

Table 5.8: Minimum and Maximum RMSEs

According to Figure 5.39, Table 5.8, and when comparing the different forecasting errors, the adaptive quantization using PSO signifies the least residual variance with a RMSE=32.13.

5.4.5 Mean Absolute Error

The Mean Absolute Error is an average of all the absolute errors; $e_i = |f_i - y_i|$, where f_i is the prediction and y_i the true value [125]. Like RMSE in the previous test, the lower values are better as it is a negatively oriented measurement. Mean Absolute Error (MAE) is formulated in Equation 5.10 [132]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{1}^{n}|e_i| \tag{5.10}$$

Following on from the one to one evaluation, the MAE test between the original frame and the noisy frame shows that the PSO method has the best result with a minimum of 0.14497.

Method	Frame rate (Frame #115)	MAE
VBR with Gaussian noise	379kbps	0.36725
CBR with Gaussian noise	266kbps	0.24408
PSO with Gaussian noise	209kbps	0.14497

Table 5.9: MAE results for VRB, CBR and PSO

Figure 5.40 shows that PSO has taken the bottommost position in comparison to both the VBR and CBR methods where high-fluctuated frames samples are used (frame 100 to frame 140) as is shown in Figure 5.12.

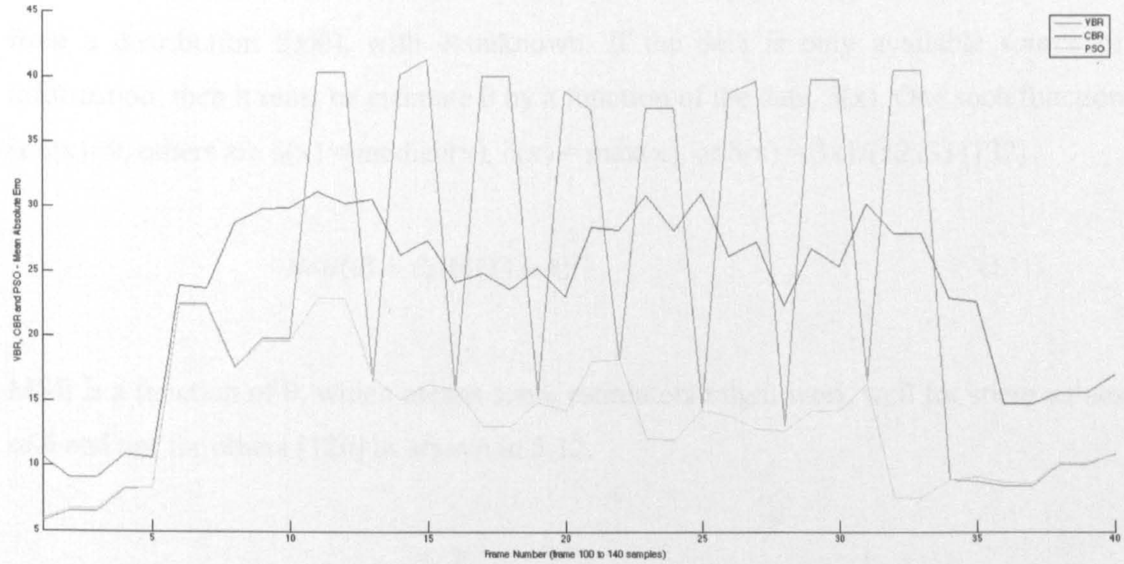


Figure 5.40: VBR. CBR and PSO MAEs

The precise MAE minimum and maximum values for each method are presented in Table 5.10.

Method	MAE	
	Min	Max
VBR	5.78	41.35
CBR	9.10	31.72
PSO	5.88	22.92

Table 5.10: MAE results

The evaluation of the MAE results shows that the PSO has maintained an optimum level of MAE compared to the VBR and CBR methods and has the lowest maximum level of 22.92. This data proves that the use of PSO, to set adaptively the optimum Q-scale, does maintain the best results and the lowest error.

5.4.6 Mean Square Error (MSE)

The Mean Square Error (MSE) is the variance of the estimator, which has the same units of measurement as the square of the quantity being estimated. The MSE as formulated in Equation 5.11 can be explained as, let x =$(x_1, \dots, x_n)$ be a random sample from a distribution f(x|θ), with θ unknown. If the data is only available source of information, then it must be estimate θ by a function of the data, δ(x). One such function is δ(x)=$\bar{x}$, others are δ(x) = median(x), δ(x) = max(x), or δ(x) = 3x1/(x2x3) [132].

$$MSE(\theta) = E_\theta[(\delta(X) - \theta)^2] \qquad (5.11)$$

MSE is a function of θ, which means some estimators might work well for some values of θ and not for others [126] as shown in 5.12.

$$MSE = \frac{1}{M\,N}\sum_{j=1}^{M}\sum_{k=1}^{N}(x_{j,k} - x'_{j,k})^2 \qquad (5.12)$$

In the one to one evaluation of the original frame versus the noisy frame, the MSE result is the lowest at 559.3130. Table 5.7: RMSE single frame evaluations shows the results of the VBR, CBR and PSO MSE evaluations.

Method	Frame rate (Frame #115)	MSE
VBR with Gaussian noise	379kbps	2608.6721
CBR with Gaussian noise	266kbps	1358.6715
PSO with Gaussian noise	209kbps	559.3130

Table 5.11: MSE single frame evaluations

Similar to the other tests, to compare the differences in VBR and CBR with the PSO, high-fluctuated frames from frame 100 to 140 are compared in Figure 5.41.

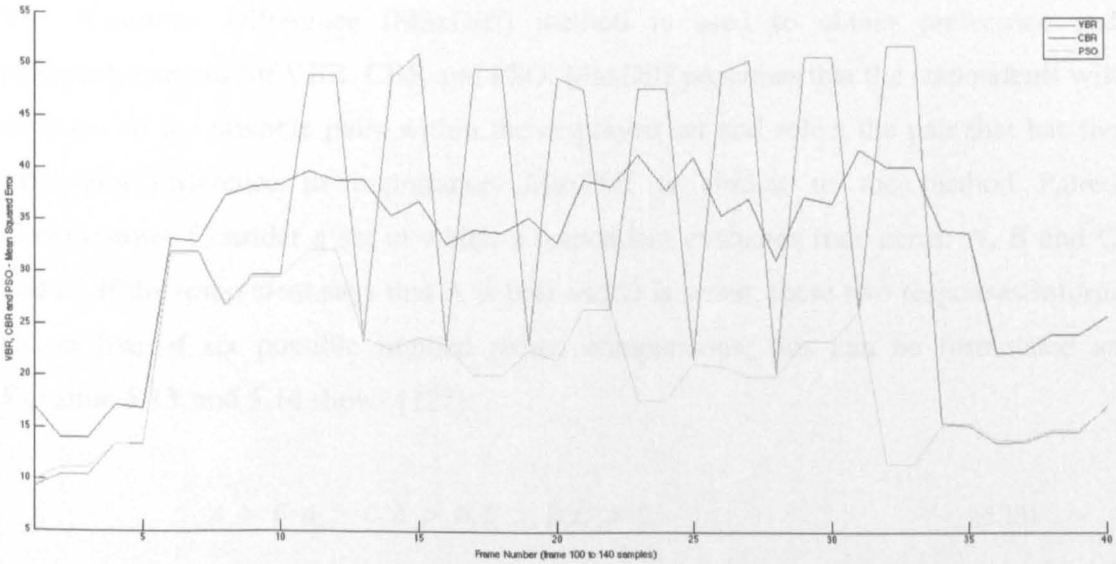


Figure 5.41: MSE in VBR, CBR and PSO

Figure 5.41 shows that the PSO has taken the bottom position in comparison to VBR and CBR. Table 5.12 shows the results of the VBR, CBR and PSO MSE evaluations for 240 frames.

Method	MSE	
	Min	Max
VBR	87.05	268.09
CBR	197.07	174.12
PSO	96.137	103.27

Table 5.12: MSE results

The evaluation of the MSE shows that the PSO has maintained an optimum level of MSE compared to the VBR and CBR methods and has the lowest maximum level.

5.4.7 Maximum Difference

The Maximum Difference (MaxDiff) method is used to obtain preference and importance scores for VBR, CBR and PSO. MaxDiff presumes that the respondents will evaluate all the possible pairs within the displayed set and select the pair that has the maximum difference in importance. MaxDiff is similar to the method Paired Comparisons. Consider a set in which a respondent evaluates four items: A, B and C and D. If the respondent says that A is best and D is worst, these two responses inform us on five of six possible implied paired comparisons; this can be formulated as Equation 5.13 and 5.14 shows [127]:

$$A > B, A > C, A > D, B > D, C > D \tag{5.13}$$

MaxDiff is formulated in 5.14:

$$\text{MaxDiff} = \text{Max}(|x_{j,k} - x'_{j,k}|) \tag{5.14}$$

The MaxDiff for one to one evaluation of the original frame versus the noisy frame is 167. Table 5.13 shows the assessments of the VBR, CBR and PSO MaxDiff.

Method	Frame rate (Frame #115)	MD
VBR with Gaussian noise	379kbps	163
CBR with Gaussian noise	266kbps	184
PSO with Gaussian noise	209kbps	167

Table 5.13: MaxDiff results in VBR, CBR and PSO

To compare the MaxDiff in VBR and CBR with the algorithm developed in this research, the high-fluctuated frames from frame 100 to 140 are compared in Figure 5.42.

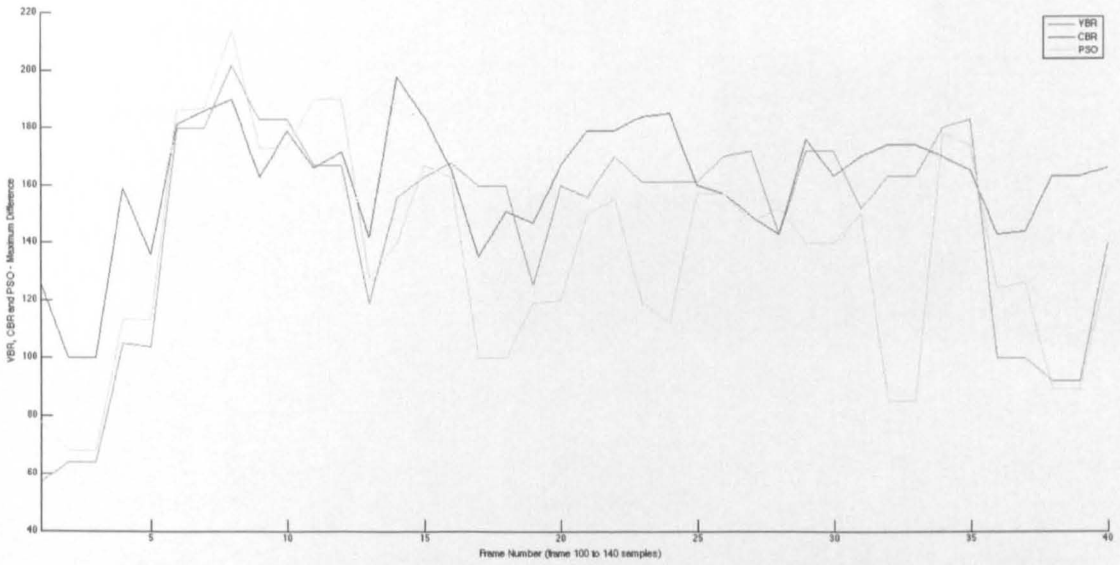


Figure 5.42: VBR, CBR and PSO MaxDiff

Table 5.14 presents the minimum and maximum value of the MaxDiff for all the methods.

Method	MaxDiff	
	Min	Max
VBR	57	202
CBR	100	198
PSO	68	214

Table 5.14: VBR, CBR and PSO MaxDiff results for 240 frames

The evaluation of the MaxDiff shows that the PSO has maintained an optimum level of MaxDiff compared to the VBR and CBR methods.

In order to be accurate about the differences a basic confusion matrix used below:

		Truth data (MaxDiff)			
		Minimum (dB)	**Maximum (dB)**	**Classification overall**	**Producer Accuracy (Precision)**
Classifier results	**VBR**	57	202	259	22%
	CBR	100	198	298	66.44%
	Truth overall	157	400	557	
	User Accuracy (Recall)	36.30%	49.5%		
	Overall accuracy	45.781%			

Table 5.15: MaxDiff confusion matrix for VBR and CBR

		Truth data (MaxDiff)			
		Minimum (dB)	**Maximum (dB)**	**Classification overall**	**Producer Accuracy (Precision)**
Classifier results	**VBR**	57	202	259	22%
	PSO	68	214	282	75.88%
	Truth overall	125	416	541	
	User Accuracy (Recall)	45.6%	51.44%		
	Overall accuracy	50.09%			

Table 5.16: MaxDiff confusion matrix for VBR and PSO

Truth data (MaxDiff)					
		Minimum (dB)	Maximum (dB)	Classification overall	Producer Accuracy (Precision)
Classifier results	CBR	100	198	298	33.55%
	PSO	68	214	282	75.88%
	Truth overall	168	412	580	
	User Accuracy (Recall)	59.524%	51.942%		
	Overall accuracy	54.138%			

Table 5.17: MaxDiff confusion matrix for CBR and PSO

According to the results shown in the confusion matrix tables and the results of the tests carried out to find the MaxDiff the PSO has a maximum MaxDiff of 51.942% and an overall accuracy of 54.138%.

5.5 Summary

In this chapter, experimental results from the simulation of ZigBee and the use of PSO in applications of MPEG-4, such as VBR and CBR models are discussed and examined. The simulation results show that the PSO method has achieved the best optimal performance in comparison to other widely used models. For this reason, the PSO model with the proposed scheme is a suitable approach for transmitting MPEG-4 over ZigBee.

Chapter 6: Conclusion

The recent growth over the past few years of wireless networks and portable devices has led to high demand for less expensive and low power consumption technologies. The IEEE 802.15.4 standard is a wireless sensor that is targeted at applications requiring low data rates, low power and less expense. IEEE 802.15.4 has potential as a cost effective, easy to use product, making it highly likely that it will soon be used to transfer large amounts of data. For this reason, this research presents a development of a design model to enable video to be transmitted over ZigBee. ZigBee is limited to a through-rate of 250kbps and is not designed to transfer a MPEG-4 compressed file; as a video file needs a substantially large amount of bandwidth to be transmitted successfully and ensure a good quality of picture.

In this research, a novel solution to transmit MPEG-4 over IEEE 802.15.4 developed. The computer simulation results from the experiments confirm that use of particle swarm optimization, as part of an optimization model and AI, to develop an adaptive scalar quantization video coding, improves the quality of picture whilst reducing data loss and communication delay, when compared to conventional MPEG video transmissions. The proposed model aims to achieve an optimum level of quality of pictures whilst maintaining the ZigBee target bitrate. The adaptive quantization increases the available bandwidth, which leads to improvement in the quality of picture by reducing the data loss. In this study, different rate control strategies compared with the proposed method and the results confirm that applying AI to the IEEE 802.15.4 standard can help the transmission of videos over the limited bandwidth of ZigBee. However, IEEE 802.15.4 bandwidth is very limited and streaming MPEG-4 video with high bitrate or HDTV signals are proven to be difficult.

The proposed model mechanism is to reside in the IEEE 802.15.4 standard transmitter device. The results of the simulations shows that streaming MPEG-4 in a VBR model leads to a substantial amount of data loss; this is because during streaming the instantaneous bitrate exceeds the data rate of ZigBee. Similarly, CBR has shown inferior quality in comparison to our model.

The simulation results when streaming both with and without the Gaussian noise, shows that the proposed model can use the available bandwidth at an optimum level. The proposed method, compared to other methods, demonstrates that it can improve MPEG-

4 streaming by using an optimum level of the available bandwidth with an improvement than the VBR model, and the CBR model.

In order to prove that these claims are correct, two statistical tests ANOVA and Kruskal-Wallis are used to examine the simulation results and PSNR as a way for objective measure for video quality.

The ANOVA test, for the rate of frame rates, determined that there is a significant difference in the means between the three models. The result of the ANOVA test shows that the proposed model in this research has less variation and spread, and no outside value. It can be concluded that the data range is in the target bitrate. Moreover, a nonparametric Kruskal-Wallis test determines that the CBR model has approximately the same variation as the VBR, and this indicates the high data rates in group of pictures for both models. However, the result for the idea in this research shows that it also has the least variation and spread range of data rates in group of pictures. The PSNR shows that the proposed idea carries an optimum value of PSNR, which is less than the maximum PSNR value of the VBR method and greater than the maximum PSNR value of the CBR method. These results determine that the proposed idea is superior to both the VBR and CBR methods.

This reduction is very significant as it allows more data to be transmitted, prevents data loss and by adaptively increasing the Q-scale to the available bandwidth limit, it allows a more complex video sequence with better video object quality. However, streaming at the target rate is a trade-off between quality and compressed bitrates in video CODEC compression, but the proposed model can control and accomplish an optimal quality of service in transmitting videos over ZigBee. Computer simulation results confirm the achievements that listed in bullets points:

- A novel adaptive scalar quantization developed to improve the quality of picture.
- A novel adaptive method using PSO introduced to the Scalar Rate Control to prevent excessive data loss.
- PSO used to improve and optimize the MPEG-4 applications.

These results confirms the use of PSO makes our proposed model superior to the VBR and CBR methods as it increases the efficiency of the bandwidth, prevents data loss and most importantly it can improve QoS and empower video within the MPEG-4 compression technique to be transmitted over IEEE 802.15.4 standard.

The proposed model in this research can be developed on commercial products, as bandwidth and low power consumption are now important factors. For this reason, it is desirable that the algorithm implemented on the IEEE 802.15.4 standard wireless device that can provide a better quality of picture than a classic device, which it can. This improvement is crucial for many devices, such as, surveillance and wireless CCTV devices, HDTV streamers, personal communication portable devices or web-cameras, as well as in the medical industry and hospitals where in intensive/invasive close monitoring care is needed.

6.1 Future recommendation

Today's wireless devices and their applications are mostly operated in personal private networks, in short-range wireless networks, which are in coexistence with long-rage networks. Hence, the demand for cost-effective and low data rate networks at a short distance wireless communication that uses low power consumption is rising. This makes the IEEE 802.15.4 standard a very attractive proposition, as it is very efficient, with low latency applications to use for MPEG-4 video streaming, and makes use of ZigBee as more than just a sensor.

This research project has accomplished a novel and an efficient technique, by applying PSO to applications of MPEG-4, ZigBee can now be used for real-time video transmission. This means, applied intelligent techniques achieved an efficient encoding, decreased the data overload and the data loss whilst maintaining a sufficient quality of picture at the transmitter end.

In the near future, more research could be carried out on this topic, to investigate the results of using a more specific intelligent technique on the different areas of video compression such as, video object segmentation, motion estimation process, entropy coding and model-based, mesh-based and wavelet-based video compression. Furthermore, the PSO model proposed in this research could be applied to a buffer management system, adding a buffer to the rate control system in the encoder prior to transmission could open a completely new area of research. Additionally, as quality of picture is an important factor in video transmission, our solution could be applied to the receiver for adaptive dequantization and to reduce the noise to transmit with a better quality of video objects and quality of service. The QoS requirements in the wireless a network includes, delay, packet losses and hop count [136]. Also at the hardware level, the idea of this research could be used to vary QoS for the delay-sensitive, bandwidth-intense and loss-tolerant applications.. However the adaptive systems can help solve the problem, but the adaptation may introduces the delay during the tuning period in which case QoS cannot be granted. The PSO which explores in search space for solving combinatorial optimization problems can be applied to solve delay-constrained problems [136]. Nonetheless, further work recommended solving delay constraint in adaptive systems using PSO or related AI techniques.

The proposed idea in this research could be developed in other research or be mixed with other types of AI techniques to develop hybrid models or schemes; as our design proves that this technique is useful in this domain.

References

[1] Y. Wang, S. Wenger, J. Wen, and A.K. Katsaggelos, Review of Error Resilient Coding Techniques for Real-Time Video Communications, IEEE Signal Proc. Magazine, Jul 2000, Vol. 17, Issue. 4, pp. 61-82, (2000)

[2] ATM Forum Inc, The User Network Interface (UNI), v. 3.1, ISBN 0-13-393828-X, Prentice Hall PTR, 1995, pp. 2-10, (1995)

[3] P. Haskell and D. Messerschmitt, Resynchronization of motion compensated video affected by ATM cell loss, Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on, 23-26 Mar 1992, Mar. 1992, Vol.3, pp. 545-548, (1992)

[4] A. Turing,. 1950, Computing Machinery and Intelligence. Mind 49, 433 – 460.

[5] C. Chang, R. Cheng, Traffic control in an ATM network using fuzzy set theory. In Proceedings of the IEEE INFOCOM'94 Conference, Toronto, Canada, June 1994. IEEE Communications Society. pp. 1200-1207 (1994)

[6] G. Ascia, V. Catania, G. Ficili, S. Palazzo, and D. PanNo. A VLSI, fuzzy expert system for real-time traffic control in ATM networks. IEEE Transactions on Fuzzy Systems, Vol. 5, Issue. 1, February 1997, pp. 20-31, (1997)

[7] R-G. Cheng and C-J. Chang, Design of a fuzzy traffic controller for ATM networks. IEEE/ACM Transactions on Networking, Vol. 4, Issue. 3, June 1996. pp. 460- 469, (1996)

[8] M-C. Huang, S. H. Hosseini, K. Vairavan & H. Lan, Fuzzy Congestion Control and Policing in ATM Networks. International Journal of Engineering (IJE), Vol. 3, Issue. 1, pp. 65-84, (2009)

[9] S-L. Wua, F-J. Sui, Z-T. Choub, An efficient quality-of-service MAC protocol for infrastructure WLANs, Journal of Network and Computer Applications 29, pp. 235-261, (2006)

[10] H. Kazemian and L. Meng, A fuzzy control scheme for video transmission in Bluetooth wireless. INS - Information Sciences, Vol. 176, Issue. 9, May 2006, pp. 1266-1289, (2006)

[11] H. Kazemian H. and L. Meng, An adaptive control for video transmission over Bluetooth. IEEE Transactions on Fuzzy Systems, Vol. 14, Issue. 2, April 2006, pp. 263-274, (2006)

[12] R. Hassan, B. Cohanim, O. De Weck, G. Venter, A Comparison Of Particle Swarm Optimization And The Genetic Algorithm. Structures, Structural Dynamics, and Materials Conference 2005, pp. 1—13, (2005)

[13] F. Yang, C. Zhang, T. Sun, Comparison of Particle Swarm Optimization and Genetic Algorithm for HMM training. Pattern Recognition, 2008. ICPR 2008. 19th International Conference. 8-11 Dec. 2008, pp. 1-4, (2008)

[14] H. Azarkish, S. Farahat,S. Masoud and H. Sarvari, Comparing the Performance of the Particle Swarm Optimization and the Genetic Algorithm on the Geometry Design of Longitudinal Fin. World Academy of Science, Engineering and Technology. vol. 1, issue 1, February 2012, pp. 1-5, (2012)

[15] S.N. Sivanandam, S.N. Deepa, A Comparative Study Using Genetic Algorithm and Particle Swarm Optimization for Lower Order System Modelling. International Journal of the Computer the Internet and Management. Date 2002, pp. 1-10, (2002)

[16] R. Burda, Multimedia over 802.15.4 and ZigBee Networks for Ambient Environment Control. Vehicular Technology Conference, 2007. Spring. IEEE 65th, 22-25 April 2007. pp. 179-183, (2007)

[17] A. Zainaldin, I. Lambadaris, B.Nandy, Video over Wireless ZigBee Networks: Multi-Channel Multi-Radio Approach. Wireless Communications and Mobile Computing Conference. IWCMC '08. International. 6-8 Aug. 2008, pp. 882- 887 (2008)

[18] K.S. Shilpa, D.G. Narayan, S. Kotabagi, M. Uma, Suitability Analysis of IEEE 802.15.4 Networks for Video Surveillance. Computational Intelligence and Communication Networks (CICN), International Conference on. 7-9 Oct. 2011 - pp. 702-706 (2011)

[19] H.B. Kazemian,. An intelligent video streaming technique in zigbee wireless. Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on 20-24 Aug. 2009, pp.121-126, (2009)

[20] H.K Arachchi, and W.A.C Fernando, PSO based bit rate optimization for MPEG-1/2 video coding. Image Processing, 2005. ICIP 2005. IEEE International Conference on 11-14 Sept. 2005, Vol. 2, pp.- 329-32, (2005)

[21] IEEE Computer Society, (August 31, 2007). IEEE Standard 802.15.4a-2007, (2007)

[22] C.N. Nyirenda and D.S. Dawoud, Self-Organization in a Particle Swarm Optimized Fuzzy Logic Congestion, Detection Mechanism for IP Networks. Scientia Iranica, Vol. 15, No. 6, December 2008, pp. 589-604, (2008)

[23] C.G. Emilio, Willea, M. Melliab, Selected Papers from the 3rd International Workshop on QoS in Multi service IP Networks (QoS-IP 2005) Algorithms for IP network design with end-to-end QoSnext term constraints star, open. Computer Networks, Vol. 50, Issue. 8, 6 June 2006, pp.1086-1103, (2006).

[24] R. V. Kulkarni and G.Kumar Venayagamoorthy, Particle Swarm Optimization in Wireless Sensor Networks: A Brief Survey October 21,2009, Vol.41, Issue. 2, pp. 262-267, (2011)

[25] IEEE Computer Society, (April 17, 2009). IEEE Standard 802.15.4c-2009, (2009)

[26] L. Peter,Fuhr, B. Walt, P. Chen, H. Kagan, D. Kaufman, andW. Manges - Wireless Technology Review: Radios, Frequencies, and Implications for Industry. Wireless Technology Review, Feb 2010, (2010)

[27] IEEE Computer Society, (April 17, 2009). IEEE Standard 802.15.4d-2009, (2009)

[28] IEEE 802 LAN/MAN Standards Committee (2012). [Online] Available from: http://www.ieee802.org/, [Accessed: 19 Oct 2009].

[29] J. Francomme, G. Mercier, and T. Val, A simple method for guaranteed deadline of periodic messages in 802.15.4 - cluster cells for automation control applications,11th IEEE International Conference on Emerging Technologies and Factory Automation September 2006, pp. 270-277, (2006)

[30] Cellnet, Why ZigBee (2012), [Online] Available from: http://www.ZigBee.org/ZigBee/en/events/documents/May2007_Open_House_Presentations/04A_ZigBee_OEM_Case_Studies_Cellnet.pdf , [Accessed: 1 Oct 2012]

[31] J.S. Lee, Y.C Huang Implementation of ZigBee, IEEE 802.15.4 Nodes for Wireless Sensor Networks. Institute (ITRI), Taiwan 18 Aug 2006. Information and Communications Research Laboratory, Industrial Technology Research (2006)

[32] ZigBee Specification, The ZigBee Specification describes the infrastructure and services available to applications operating on the ZigBee platform - January 17 2008, (2008)

[33] S. Farahani, ZigBee Wireless Networks and Transceivers - ISBN: 978-0-7506-8393, (2008)

[34] M. S. Hansen, S. Støa, Practical Evaluation of IEEE 802.15.4/ ZigBee Medical Sensor Networks - June 2006 - Thesis, Norwegian University of Science and Technology Department of Electronics and Telecommunications, (2006)

[35] D. Yang, Y. Xu and M. Gidlund, Wireless Coexistence between IEEE 802.11- and IEEE 802.15.4-Based Networks : A Surve, International Journal of Distributed Sensor Networks, Vol. 2011, Apr 2011, Article ID 912152, pp. 17, (2011)

[36] ZigBee/IEEE 802.15.4 Summary - Sinem Coleri Ergen, (September 10, 2004) [Online] Available from: http://pages.cs.wisc.edu/~suman/courses/838/papers/ZigBee.pdf [Accessed 18 Oct 2009].

[37] C.J.M. Liang, N.B. Priyantha, J.Liu, A. Terzis, Surviving Wi-Fi Interference in Low Power ZigBee Networks, November 3-5, 2010, pp. 309-322, (2010)

[38] Daintree Networks, Getting Started with ZigBee and IEEE 802.15.4, (2008) [Online] Available from: http://www.daintree.net/downloads/whitepapers/ZigBee_primer.pdf[Accessed 20Jun 2011].

[39] K. Pahlavan, and P. Krishnamurthy, Networking Fundamentals: Wide, Local and Personal Area Communications e-book. ISBN10: 0470992891, (2009)

[40] P. Cordierl, and M. Terre, Impact of Clear Channel Assessment Mode on the Performance of ZigBee Operating in a WiFi Environment - Madani Zeghdoudl, France Telecom R&D 2006 IEEE, (2006)

[41] S.S.R. Ahamed, The Role Of ZigBee Technology In Future Data Communication System - Journal of Theoretical and Applied Information Technology 2005 - 2009, (2009)

[42] T.S. Hanzral, G. Singh, Performance of Free Space Optical Communication System with BPSK and QPSK Modulation, IOSR Journal of Electronics and Communication Engineering (IOSRJECE), ISSN : 2278-2834 Vol. 1, Issue 3 (May-June 2012), pp.38-43, (2012)

[43] R. Ahmad, O. Side, W. M. H. Wan, S. Hassin, K.K. Moh, Development of Verilog-Based OQPSK Demodulator for Zigbee Applications on FPG, European Journal of Scientific Research, Vol. 64 Issue 4, November, 2011, pp. 555-562, (2011)

[44] P. Wararkar, N. Chavan, Wireless Integrated System using IEEE 802.15.4, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 2, Issue. 3, May-Jun 2012, pp. 395-399, (2012)

[45] L. Greve, ECEE 4242 Fall 2010, (2010), [Online] Available from: http://ecee.colorado.edu/~liue/teaching/comm_standards/2010F_802.15/home.html, [Accessed 01 Oct 2012].

[46] LAN/MAN Standards Committee of the IEEE Computer Society, IEEE Std 802.11b-1999 (R2003) - the IEEE 802.11b specification,e-book, 16 September 1999, ISBN 0-7381-1812-5, (1999)

[47] J. Geie, 802.11 Beacons Revealed,(2002), [Online], Available from: http://www.wi-fiplanet.com/tutorials/article.php/1492071, [Accessed 3 Des 2011]

[48] K. Schader, ZigBee Preface, ZigBee Standards Organization, e-book, ZigBee Alliance, Available from: ZigBee.org http://ZigBee.org/imwp/idms/popups/pop_download.asp?contentID=15865 [Online] [Accessed 19 Oct 2009], pp. 17, (2009)

[49] N-C. Liang, P-C. Chen, T. Sun, G. Yang, L-J. Chen, M. Gerla, Zigbee mesh network - Impact of Node Heterogeneity in ZigBee Mesh Network Routing, Man and Cybernetics, 2006. SMC apos;06, IEEE International Conference on Vol. 1, Issue. 8, 11 Oct. 2006 pp.187 – 19, (2006)

[50] Software Technologies Group, Inc, Types of ZigBee Networks, stg.com, (2009). [Online],Available from: http://www.stg.com/wireless/ZigBee_netw.html, [Accessed 26 Aug 2012]

[51] Y. Yu, L. Chen, Enhanced Protection Mechanism for Improving Co-Existence of IEEE 802.11b and IEEE 802.11g Wireless LANs. Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th. pp. 1- 5, (2009)

[52] ARC Electronics 2009. RS232 Data Interface, (2009), [Online] Available from: http://www.arcelect.com/rs232.htm, [Accessed 20 Oct 2009]

[53] Computer Fraud & Security, How can Bluetooth services and devices be effectively secured?, Vol. 2006, Issue. 1, January 2006, Pages 4-7 Elsevier - ScienceDirect, Oct 2009, pp. 4-7, (2009)

[54] Wireless Developer Network, a MindSites Group, LLC Trade Community 2009. Wireless DevNet, An Introduction to Bluetooth, (2009), [Online],Available from: http://www.wirelessdevnet.com/channels/bluetooth/features/bluetooth.html [Accessed 20 Oct 2009]

[55] O. Tickoo, B. Sikdar, On the impact of IEEE 802.11 MAC on traffic characteristics. Selected Areas in Communications, IEEE Journal, Feb 2003, Vol. 21, No. 2, pp. 229-239, (2003)

[56] Y. Liu, M. Lee, and T. Saadawi, A Bluetooth Scatternet Route structure for Multi-hop Ad Hoc Networks IEEE JSAC, Selected Areas in Communications, IEEE Journal, Feb 2003. Vol.: 21, Issue. 2, pp. 229-239 (2003)

[57] M. Walters, C. Walker, D. Powell, What kind of security is on the Z-Wave system? How do I prevent an external source from turning on or off lights or systems?, (Updated 26 February 2007), [Online] Available from: http://www.zwaveworld.com/ask/ask8.php [Accessed 20 Oct 2009]

[58] Ofcom 2009 IEEE, ULTRA WIDE BAND (UWB) COLLOQUIUM. (Updated 23 July 2002), [Online], Available from: http://www.ofcom.org.uk/static/archive/ra/topics/uwb/uwb-index.htm [Accessed 20 Oct 2009].

[59] M. S. Imtiaz, DSSS Digital Transceiver Design For Ultra Wideband. International Journal of Advances in Engineering & Technology, Nov 2011. IJAET, Vol. 1, issue, 5, pp.144, (2011)

[60] THE MOVING PICTURE EXPERTS GROUP (MPEG), (2008), [Online] http://mpeg.chiariglione.org/, [Accessed 20 Oct 2009].S. S. Madeiro, J. A. Carmelo, B. Filho, B. Fernando, Lima. Neto, Adaptive clustering Particle Swarm Optimization. IPDPS '09 Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing, pp. 1-8 (2009)

[61] I. E. G. Richardson, H.264 and MPEG-4 Video Compression Video Coding for Next-Generation Multimedia: Video Coding for Next-generation Multimedia - ISBN-10: 0470848375 (2003)5

[62] J. Golston, Comparing Media CODECs for Video Content. # 250 at Embedded Systems Conference, San Francisco 2004. (2004), e-Book, 26 Nov 2011, pp. 18, (2011)

[63] S. Aramvith, S. Aramvith , M-T. Sun, MPEG-1 AND MPEG-2 Video Standards, 1999 Academic Press, Image and Video Processing, pp. 597-610, (1999)

[64] R. Koenen, MPEG-4 Overview - (V.21 – Jeju Version) , International organisation for standardisation organisation internationale de normalisation iso/iec jtc1/sc29/wg11 coding of moving pictures and audio. ISO/IEC JTC1/SC29/WG11 N4668 March 2002, (2002)

[65] Peiya Liu, MPEG-4: A Multimedia Standard for the Third Millennium, Part 1, IEEE MultiMedia, Vol. 1, Number 1, 1994, pp.89-90, (1994)

[66] T.K. Capin and D. Thalmann, Controlling and Efficient Coding of MPEG-4 Compliant Avatars. Proceeding of IWSNHC3DI, Santorini, Greece, 1999. (1999)

[67] B.G. Haskell and A. Puri, MPEG Video Compression Basics - L. Chiariglione (ed.), The MPEG Representation of Digital Media, Springer, ISBN: 978-1-4419-6183-9, pp.9-38, (2012)

[68] H. Wu, M. Claypool, R. Kinicki, Guidelines for Selecting Practical MPEG Group of Pictures, IMSA'06 Proceedings of the 24th IASTED international conference on Internet and multimedia systems and applications, ACM, 2006, pp. 61-66, (2006)

[69] D. Marshall, MPEG Video, B-Frames, (2001), [Online], Available from: http://www.cs.cf.ac.uk/Dave/Multimedia/node258.html [Accessed 08 Aug 2012]

[70] A. Barjatya, Block Matching Algorithms For Motion Estimation, Spring 2004, 30 Oct 2004, pp.485-492, (2007)

[71] M. Yang, H. Cui and K. Tang, Efficient tree structured motion estimation using successive elimination. Vision, Image and Signal Processing, IEE Proceedings,Issue. 5, Vol 151, October 2004, pp.369-377, (2004)

[72] M. Alwakeel and Z. Shaaban, Face Recognition Based on Haar Wavelet Transform and Principal Component Analysis via Levenberg-Marquardt Backpropagation Neural Network. European Journal of Scientific Research, ISSN 1450-216X Vol.42, No.1, pp.25-31, (2010)

[73] Syed Ali Khayam, The Discrete Cosine Transform, Theory and Application, eBook, 23 March 2003, Michigan State University, pp. 32, (20036)

[74] Computer Vision System Toolbox, Compute 2-D discrete cosine transform (DCT), MATLAB, [Online], Available from: http://www.mathworks.co.uk/help/toolbox/vision/ref/2ddct.html, [Accessed 08 Aug 2012]

[75] P.N. Tudor, Res. & Dev. Dept., BBC, Tadworth, UK, MPEG-2 VIDEO COMPRESSION, IEE Journal, December 1995, Vol. 7, Issue. 6, pp. 257-264

[76] R.M. Gray & D.L. Neuhoff, 'Quantization', IEEE Transactions on Information Theory Journals, Vol. 44, Issue. 6, October 1998, pp.2325-2383, (1998)

[77] S. So and K. K. Paliwal, Comparison of Quantization Techniques for DCT-based Image Coding. Proc. Microelectronic Engineering Research Conference, Nov 2001, (2001)

[78] M. Sharma, Compression Using Huffman Coding. IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010, pp. 133-141, (2010)

[79] T. Edwards, Discrete Wavelet Transforms: Theory and Implementation, Stanford University, September 1991, [Online] Available from: http://qss.stanford.edu/~godfrey/wavelets/wave_paper.pdf, [Accessed 29 Aug 2012]

[80] N. Topiwala. Wavelet image and video compression, ISBN:0792381823, 1998, (1998)

[81] M. Ghanbari. Standard CODEC – Image compression to advanced video, coding, ISBN: 0-85296-710-1, 2003, (2003)

[82] N. Hirose, Coding of audio-visual objects. Part 3: Audio, eBook, Third edition 1 Dec 2005, pp. 90-128, (2005)

[83] J. Signe,Y. Fisher, A. Eleftheriadis, MPEG-4's binary format for scene description, Signal Processing: Image Communication 15 (2000), Elsevier, pp. 321 - 34, (2000)

[84] C. Herpel and A. Eleftheriadis, MPEG-4 Systems: Elementary stream management, Signal Processing: Image Communication 15, Special Issue on MPEG-4, Elsevier, pp. 321-345, (2000)

[85] J.R. Ohm, Stereo/Multiview Video Encoding Using the MPEG Family of Standards, May 1999, Proc. SPIE, Vol. 3639, pp. 242-253, (1999)

[86] A. Tripathi and M. Claypool, Improving Multimedia Streaming with Content-Aware Video Scaling. In Proceedings of the Second International Workshop on Intelligent Multimedia Computing and Networking (IMMCN), Mar. 2002, (2002)

[87] D. Singer and W. Belknap, Intermedia Format (MP4) VM text - MPEG-4 Systems - international organisation for standardisation organisation internationale de normalisation, iso/iec jtc1/sc29/wg11- coding of moving pictures and audio, ISO/IEC JTC1/SC29/WG11, MPEG98/N2612p4, Dec 1998, (1998)

[88] Crevier, Russell and Norvig, The starting and ending dates of the sections in this article are adopted, 2003, pp.16–27, (2003)

[89] G. F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving. ISBN-13: 9780321545893, pp. 1-2, (2008)

[90] A. Bennet and D. Bennet, The Intelligent Complex Adaptive System Model for Organizations, (2012), [Online] Available from: http://www.mountainquestinstitute.com/ICAS.htm [Accessed 07Aug 2012].

[91] S. A. Frank, The Design of Natural and Artificial, Adaptive Systems, in Adaptation, M.R. Rose and G.V. Lauder, eds. Academic Press, Chapter 14, pp. 451-505, (1996)

[92] H. H. Lund, Adaptive Approaches Towards Better GA, Performance in Dynamic Fitness Landscapes, ISSN 0105-8517, No 487, (1994)

[93] C. Blum and D. Merkle, Swarm Intelligence, Springer; 2008 edition, ISBN-10: 3540740880, ISBN-13: 978-3540740889, P43, (2008)

[94] S. Luke, Essentials of Metaheuristics. Lulu, eBook, 2009, ISBN: 978-0-557-14859-2, (2009)

[95] E. D. Acebo and J. Lluis de la Rosa, Introducing Bar Systems: A Class of Swarm Intelligence Optimization Algorithms, CiteSeerX, (2013)

[96] J. Brownlee, Clever Algorithms - Nature-Inspired Programming Recipes - ISBN: 978-1-4467-8506-5, (2011)

[97] G. Beni, and J. Wang, Swarm Intelligence in Cellular Robotic, Systems , Proceedings of NATO Advanced Workshop on Robots and Biological Systems, Proceeding SAB'04 Proceedings of the 2004 international conference on Swarm Robotics, Vol 102, ISBN:3-540-24296-1 978-3-540-24296-3, pp. 1-9, (1989)

[98] G. Narzisi, Classic Methods for Multi-Objective Optimization, (2008), [Online] , Courant Institute of Mathematical Sciences, New York University Available from: http://cims.nyu.edu/~gn387/glp/lec2.pdf [Accessed 08 Aug 2012]

[99] E. R. Morey, Minimization & Maximization, (2002), [Online] Available from: http://www.colorado.edu/economics/morey/6808/min%26max.pdf [Accessed 08 Aug 2012]

[100] Pedersen, M.E.H.: SwarmOps for Matlab - Numeric & Heuristic Optimization, November 2010, http://www.hvass-labs.org/projects/swarmops/matlab/, (2010)

[101] J. Kennedy, R. C. Eberhart, and Y. Shi., Swarm Intelligencel, Morgan Kaufmann Publishers, San Francisco, CA, (2004)

[102] Kennedy, J. and R.C. Eberhart, Particle swarm optimizationl, Proc. IEEE Int'l Conf. on Neural Networks, IV, 1942–1948. Piscataway, NJ: IEEE Service Center, (1995)

[103] A. Owen and I. Harvey, Adapting Particle Swarm Optimization for Fitness Landscapes with Neutrality, 1-5 April 2007, IEEE Swarm Intelligence Symposium, pp. 258-265 (2007)

[104] J. Chunlin, Z. Yangyang, T. Mengmeng, and Y. Shengxiang,Particle Filter with Swarm Move for Optimization - Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X - pp. 909-918. (2008)

[105] V. Selvi and R.Umarani, Comparative Analysis of Ant Colony and Particle Swarm Optimization Technique, International Journal of Computer Applications, Vol. 5- No.4, August 2010, pp. 22-27, (2012)

[106] Y. Penga, B. L. Lua, A hierarchical particle swarm optimizer with latin sampling based memetic algorithm for numerical optimization, Applied Soft Computing Journal, Elsevier, Jun 20912, pp 1-14, (2012)

[107] A. Abraham, S. Das, S. Roy, Swarm Intelligence Algorithms for Data Clustering, Soft Computing for Knowledge Discovery and Data Mining 2008, Springer, ISBN 978-0-387-69934-9, pp. 279-313, (2008)

[108] M. Dorigo, L. Maria Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, Evolutionary Computation, IEEE Transactions on Apr 1997 - Vol. 1, Issue. 1 pp. 53-66, (1997)

[109] J. E. Onwunalu, L. J. Durlofsky, Application of a particle swarm optimization algorithm for determining optimum well location and type, Computational Geosciences, Vol 14, Issue 1, ISSN 1420-0597, Jan 2010, Springer Science, pp. 183-198, (2010).

[110] X. Li and P. Andries Particle Swarm Optimization: An introduction and its recent developments, The 2007 GECCO conference companion on Genetic and evolutionary computation , ISBN: 978-1-59593-698-1, pp. 3391-3414, (2007)

[111] Clerc, M.: Particle Swarm Optimization. iSTE, ISBN 10: 1-905209-04-5, (2006)

[112] A. J. R. Medinal, G. T. Pulido and J. G. Ram'ırez-Torres, A Comparative Study of Neighborhood Topologies for Particle Swarm Optimizers, Oct 2009, Proceedings of the International Joint Conference on Computational Intelligence, pp. 152-156, (2009)

[113] S. Abraham, S. Sanyal and M. Sanglikar, Particle Swarm Optimization Based Diophantine Equation Solver, Int. Journal of Bio-Inspired Computation, March 2010, Vol.2, No. 2, pp. 100 - 114, (2010)

[114] S. Ghosh, S. Das, D. Kundu, K. Suresh, A. A. Genev, Inter-particle communication and search-dynamics of lbest particle swarm optimizers: An analysis. Nature-Inspired Collective Intelligence in theory and Practice, Vol. 182, Issue 1, 1 January 2012, pp. 156--168, (2012)

[115] J. J. Moré, S. B. Garbow, E. Kenneth, Hillstrom, Algorithm 566: FORTRAN Subroutines for Testing Unconstrained Optimization Software [C5], [E4]. Journal ACM Transactions on Mathematical Software, Vol 7, Issue 1, March 1981, pp. 136-140, (1981)

[116] P.N, Sugatan,et al., Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, KanGAL, Tech. Rep 2005005, Nanyang Technological University, Singapore., May 2005. (2005)

[117] I. I. C. Liera, M. A. C. Liera, M.C. J. A. Castro, Parallel particle swarm optimization using GPGPU. Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on 9-11 Feb 2011, pp. 560-564, (2011)

[118] R. Germundsson, Mathematica Version 4, 2000, Mathematica Journal, 7, pp. 497-524, (2000)

[119] H. H. Rosenbrock, An Automatic Method for Finding the Greatest or Least Value of a Function, Computer Journal. 3, 1960, pp. 175-184, (1960)

[120] T. Schoene, S. A. Ludwig, J. Raymond , Spiteri- Step-Optimized Particle Swarm Optimization. Evolutionary Computation (CEC), 2012 IEEE Congress on 10-15 June 2012, pp. 1- 9 (2012)

[121] J.F. Schutte, J.A. Reinbolt, B.J. Fregly, R.T. Haftka1, A.D. George, Parallel Global Optimization with the Particle Swarm Algorithm, Journal Of Numerical Methods In Engineering 2003, (2003)

[122] Jitesh, Modulation & Demodulation in Zigbee, Matlab file exchange[Online] Available from: http://www.mathworks.co.uk/matlabcentral/fileexchange/36258-modulation-demodulation-in-ZigBee, [Accessed 18 Apr 2012]

[123] M. Hast, M. Ohlin, D. Henriksson, A. Cervin, J. Eke, TrueTime, Simulation of Networked and Embedded Control Systems. http://www.control.lth.se/truetime/ Department of Automatic Control, Lund University June 2010, (2012)

[124] H. W. Tranter, K. S. Shanmugan T. S. Rappaport, K. L. Kosbar, Principles of Communication Systems Simulation with Wireless Applications, ISBN-13: 978-0-13-494790-7, (2003)

[125] K. Taji, T. Miyake & H. Tamura, On Error Backpropagation Algorithm Using Absolute Error Function, Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference, 1999, Vol. 5, pp. 401-406, (1999)

[126] M. Zhou, Estimators, Mean Square Error, and Consistency, eBook, January 20, 2006, University of Kentucky, Available from: http://www.ms.uky.edu/~mai/sta321/mse.pdf, [Accessed 18 Apr 2012]

[127] S. H. Cohen and SHC & Associates, Maximum Difference Scaling: Improved Measures of Importance and Preference for Segmentation, Sawtooth Software Conference Proceedings, 2003, Sequim, WA. (2003)

[128] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler , Network Working Group, (2007), [Online], Available from: http://tools.ietf.org/html/rfc4944 Online, [Accessed 03 Oct 2012]

[129] Zhou Wang, Member, IEEE, Alan C. Bovik, Fellow, IEEE - Hamid R. Sheikh, and Eero P. Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions On Image Processing, Vol. 13, No. 4, April 2004. Journal, pp. 1-14, (2004)

[130] Z. Wang and A. C. Bovik, A universal image quality index, IEEE Signal Processing Letters, Mar. 2002, Vol. 9, pp. 81–84, (2002)

[131] Chaitanya Chukka, Image Quality Assessment, eBook, Jul 2010, (2012)

[132] C. J. Willmott and K. Matsuura, On the use of dimensioned measures of error to evaluate the performance of spatial interpolators. International Journal of Geographical Information Science. Vol. 20, No. 1, January 2006, pp. 89–102, (2006)

[133] H. Rady,Entropy and Mean Square Error for Improving the Convergence of Multilayer Backprobagation Neural Networks : A Comparative Study, International Journal of Electrical & Computer Sciences IJECS-IJENS Vol. 11, No. 05, pp. 68-79, (2012)

[134] V. Struc, The INface toolbox v2.0 for illumination invariant face recognition. Photometric normalization techniques used for achieving illumination invariant face recognition, 29 Jan 2010 (Updated 26 Jan 2012), [Online] http://www.mathworks.co.uk/matlabcentral/fileexchange/26523-the-inface-toolbox-v2-0-for-illumination-invariant-face-recognition, [Accessed 05 Oct 2012]

[135] R. Wang, Difference of Gaussian (DoG), eBook, Version 2008 (1.71), [Online], Available from: http://fourier.eng.hmc.edu/e161/lectures/gradient/node10.html, [Accessed 05 Oct 2012]

[136] R. Qu, Y. Xu, J. P. Castro, D. Landa-Silva, Particle Swarm Optimization for the Steiner Tree in Graph and Delay-Constrained Multicast Routing Problems, Journal of Heuristics, March 2012, Springer, pp. 1-20, (201

Abbreviations

4

4SS: Four Step Search 53

A

ACO: Ant Colony Optimization 70

ACS: The Ant Colony Systems 70

AI: Artificial Intelligence 5

AIEE: American Institute of Electrical Engineers 15

ANOVA: Analysis of Variance 126

ANSI: American National Standards Institute 4

APS: Application Support 18

ARPS: Adaptive Rood Pattern Search 54

ASK: Amplitude Shift Keying 25

ATM: Asynchronous Transfer Mode 3

AWGN: Additive White Gaussian Noise 20

B

BER: Bit Error Rate 96

BLER: Block Error Rate 99

BPSK: Binary Phase Shift Keying 25

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z

Bibliography

[1] D. Soud, H.B. Kazemian, A fuzzy approach to connection admission control in 802.15.4b wireless systems. The 29 th Annual Conference of the IEEE Industrial Electronic Society (IECON'03), 6th Nov. 2003, pp. 1914-1919, (2003)

[2] D. Soud, H.B. Kazemian, A fuzzy approach to active usage parameter control in IEEE 802.15.4b wireless networks. Expert Systems: The Int. Journal of Knowledge Engineering and Neural Networks. Blackwell publishers Ltd., Nov. 2004, vol. 21, No. 5, pp. 269-278, (2004)

[3] IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) - 2002, pp. 1- 473, (2002)

[4] N. Sarris, M. G. Strintzis, 3d Modeling & Animation: Synthesis and Analysis Techniques for the Human Body. ISBN 1-59140-299-9 h/c

[5] J. Zheng, and M. J. Lee, Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?: A Discussion on a Potential Low Power Low Bit Rate Standard IEEE Communications magazine, June 2004, pp. 140-146, (2004)

[6] Paul Ferguson, Geoff Huston, Quality of service: delivering QoS on the Internet and in corporate networks, Wiley, 1998, Original from the University of Michigan, Digitized 6 Dec 2007, ISBN 0471243582, 9780471243588

[7] M. Guizani, A. Rayes, B. Khan, A. Al-Fuqaha, Network Modeling and Simulation: A Practical Perspective. Wiley-Interscience, 2010, ISBN 0470035870, 9780470035870, (2010)

[8] D.H.K. Tsang, B. Bensaou, S.T.C. Lam, Fuzzy-based rate control for real-time MPEG video, IEEE Transaction on Fuzzy Systems 6 (4) (1998), pp. 504–516, (1998)

[9] J. Bates, Optimizing voice in ATM/IP mobile networks, McGraw-Hill telecom: Engineering, McGraw-Hill Professional, 2002, ISBN 0071395946, (2002)

[10] D.L. Gall, MPEG: A video compression standard for multimedia application, Communications of the ACM 34 (4) (1991), pp. 46-58, (1991)

[11] L., W.S. Linb, S. Rahardjab, X. Linb, X.K. Yangc, Z.G. Lib, Geometrically determining the leaky bucket parameters for video streaming over constant bit-rate channels Ping, Signal Processing: Image Communication 20, pp. 193-204, (2005)

[12] Y. Xiao, C.L.P. Chen, Y. Wang, Optimal admission control for multi-class of wireless adaptive multimedia services, IEICE Transaction on Communications, special issue on Mobile Multimedia Communications E84-B (4) (2001) 795-804.

[13] Mobile Computing, IEEE Transactions on, Issue Date: Nov. 2006, Vol.: 5 Issue.11, ISSN: 1536-1233, pp 1561-1574, (2006)

[14] G. F luger, Artificial intelligence Structures and Strategies for Complex Problem Solving, Fifth Edition. ISBN 0 321263189

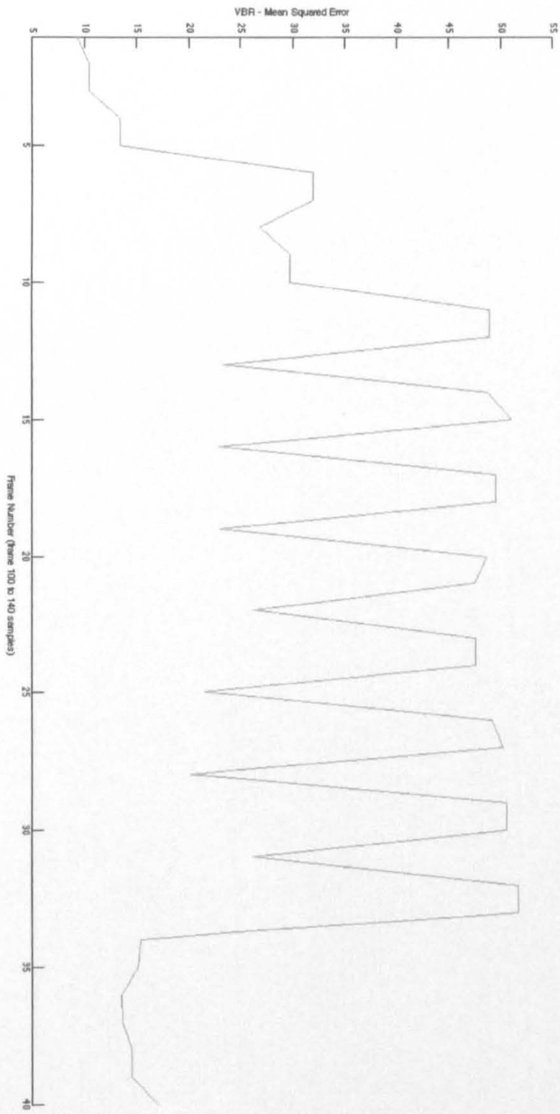
[15] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, IEEE - Overview of the H.264/AVC Video Coding Standard - IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, , JULY 2003, VOL. 13, No. 7pp. 560-576.

[16] J. Mitchell and W. P, nnebaker. MPEG Video: Compression Standard. Chapman and Hall, 1996. ISBN 0412087715.

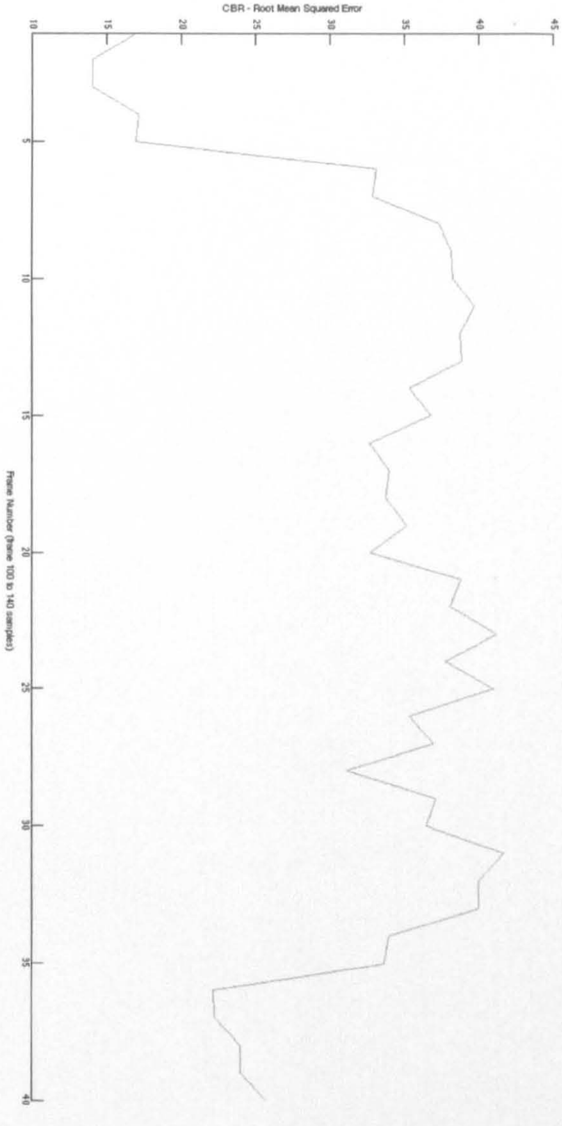
[17] K. Park and W. Wang. QoS-Sensitive Transport of Real-Time MPEG Video Using Adaptive Forward Error Correction. In Proceedings of IEEE Multimedia Systems, Jun 1999, pp. pages 426 - 432, (1999)

Appendix I

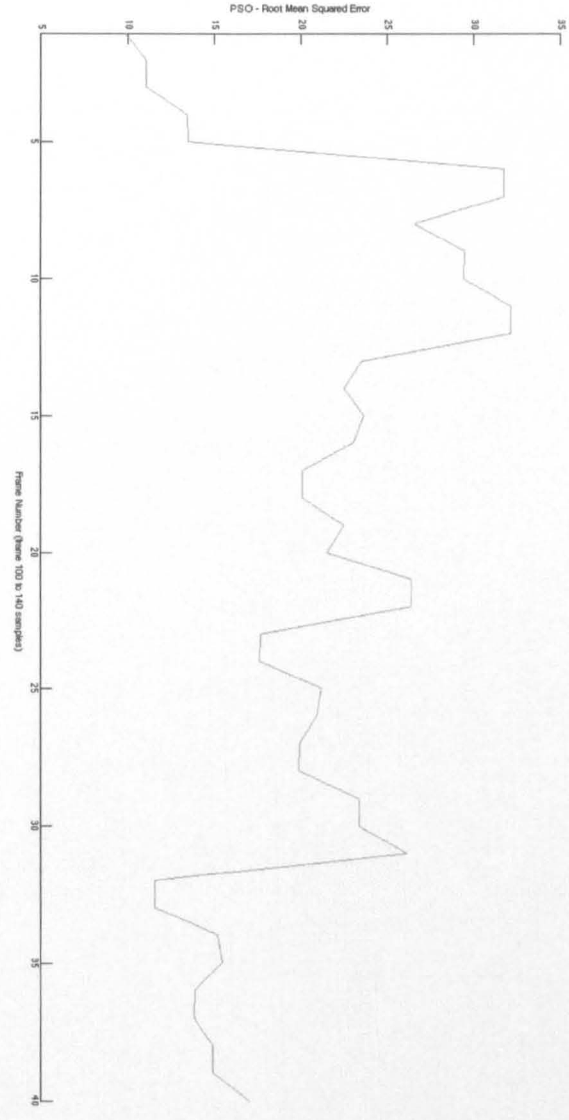
VBR, Root Mean Square Error



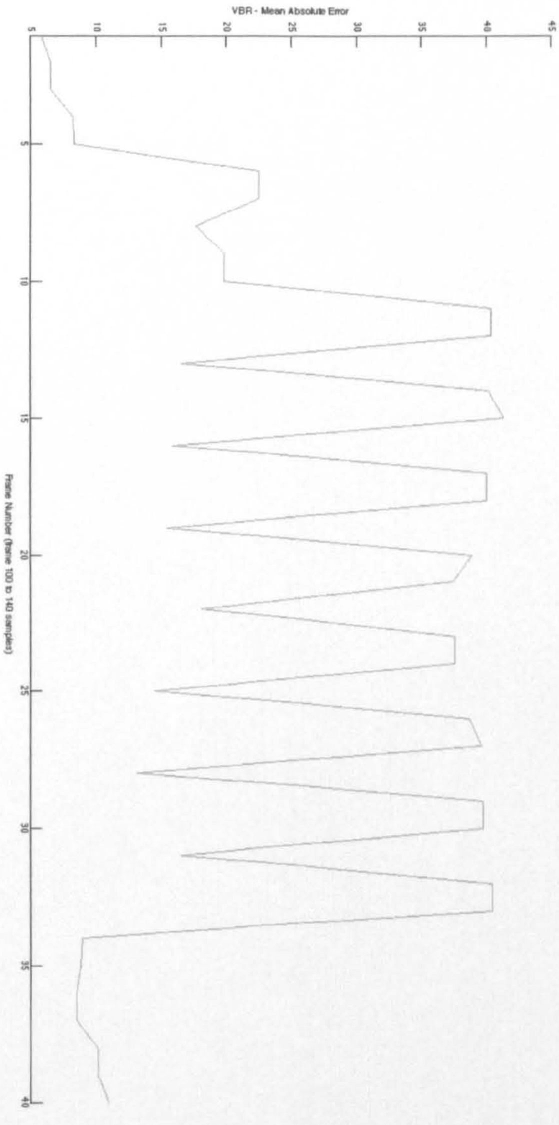
CBR, Root Mean Square Error



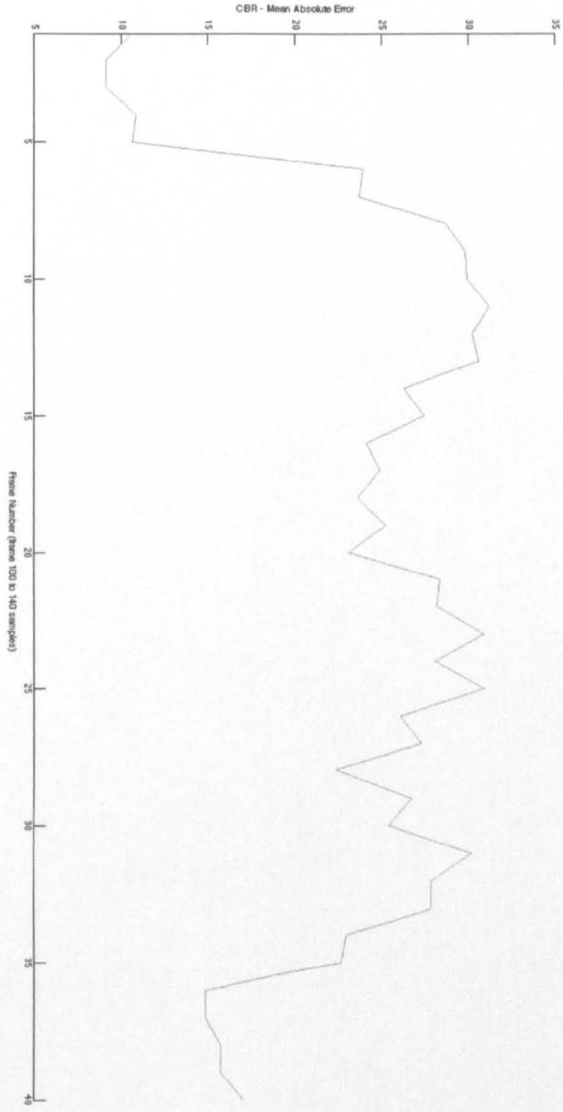
PSO, Root Mean Square Error



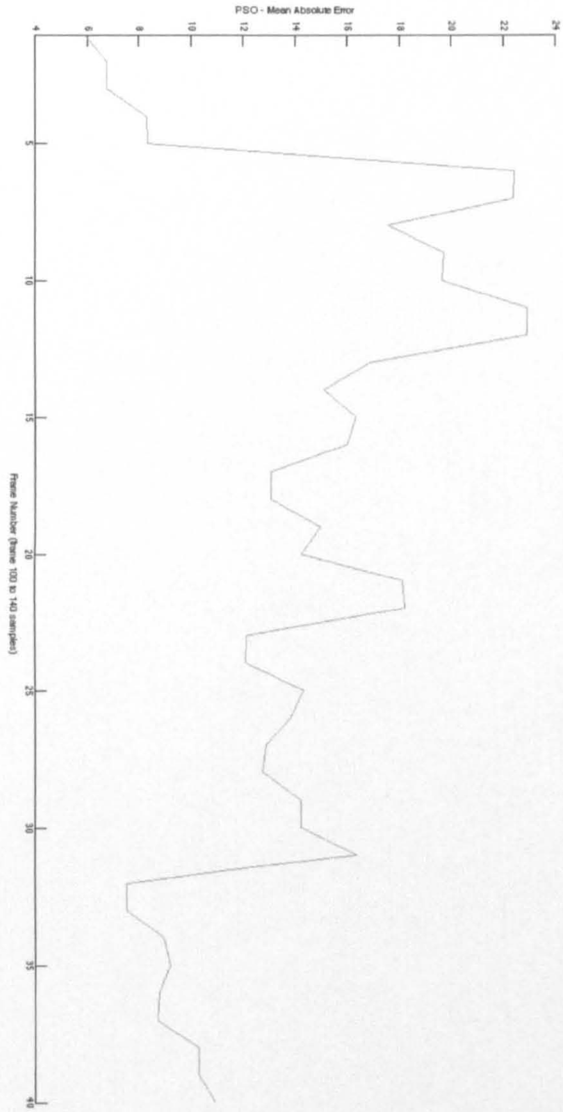
VBR, Mean Absolute Error



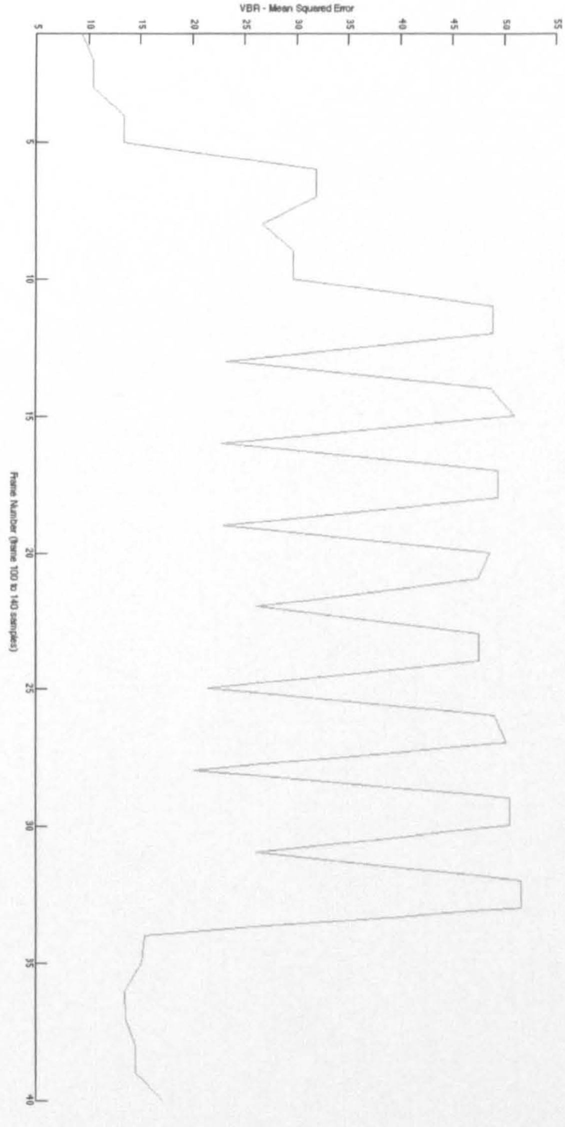
CBR, Mean Absolute Error



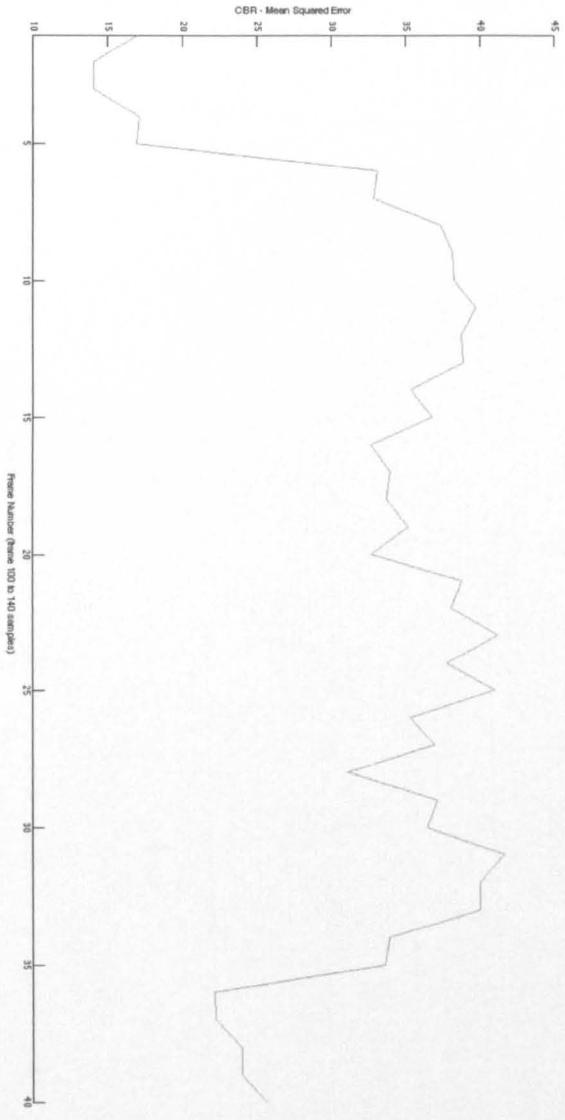
PSO, Mean Absolute Error



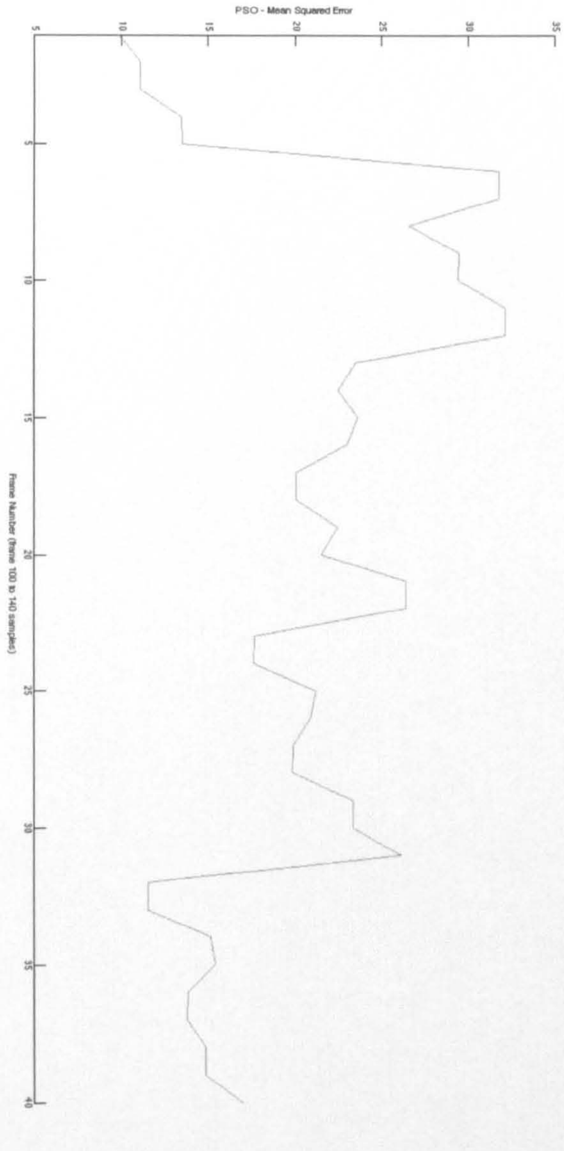
VBR, Mean Squared Error



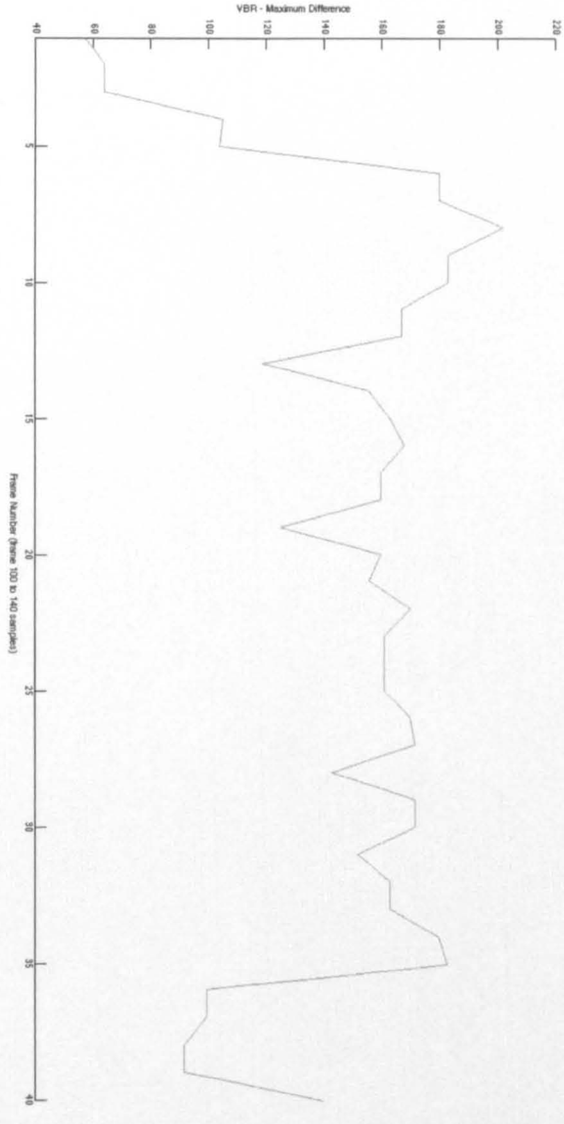
CBR, Mean Squared Error



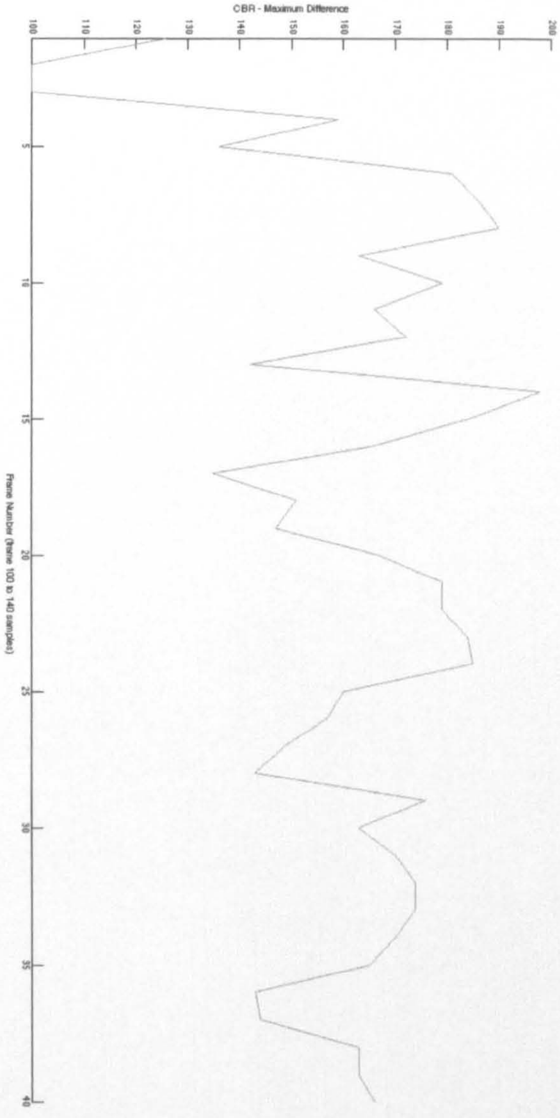
PSO, Mean Squared Error



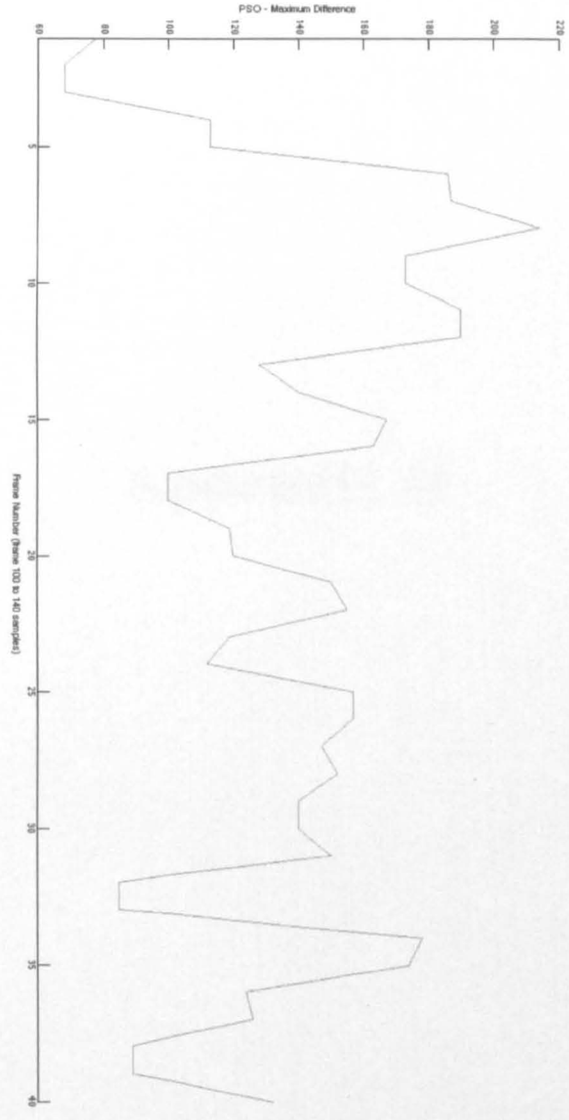
VBR, Maximum Difference



CBR, Maximum Difference



PSO, Maximum Difference



Appendix II

Adaptive Scalable Rate Control over IEEE 802.15.4 using Particle Swarm Optimization

(Submitted at International Conference on Swarm Intelligence (ICSI 2013), June 12-15, 2013)

Anonymous authors

Abstract. The IEEE 802.15.4 standard, known as ZigBee, is limited to a through-rate of 250kbps providing support for small packet file transitions and it is designed to provide highly efficient connectivity with low power-usage. ZigBee is commonly used in wireless architecture and in controlling and monitoring applications. ZigBee's cost effective potential makes it highly likely that it will soon be used to transfer large amounts of data or stream video. However, ZigBee's bandwidth is very low for video transmissions over IEEE 802.15.4 networks therefore this will be difficult to achieve. Additionally, the ZigBee limitation could become a real problem if the user wishes to transmit a large amount of data in a very short time. Hence, in this paper a solution has been accomplished by applying Particle Swarm Optimization to Scalable Rate Control in order to increase the available bandwidth, which leads to both an improvement in the quality of picture and a reduction in the data loss when transmitting MPEG-4 video over the ZigBee wireless sensor networks.

Keywords: Adaptive Scalable Rate Control, MPEG-4, Particle Swarm Optimization, ZigBee, Video streaming, IEEE 802.15.4

1 Introduction

The IEEE 802.15.4 standard known as ZigBee is a new frequency standard in wireless technology. It is designed to be cost-effective and is targeted at Low-Rate Wireless Personal Area Networks (LR-WPANs) as a short distance wireless communication, with low power consumption, radio frequency applications. The IEEE 802.15.4 standard is aimed primarily at remote control and sensor applications. It has developed extremely fast in smart homes and smart office networks with flexibility and seamless mobility [1]. There are already many existing wireless technologies such as WI-FI, UWB, Bluetooth and WIMAX, but these technologies have been designed purely based on the need for high data rates, which affects the power consumption as well as the cost. None of these technologies fulfilled their potential of producing a low power and low data technology, something that ZigBee aims to do. ZigBee technology is intended to be simpler and less expensive than other Wireless Personal Area Networks (WPANs) such as Bluetooth, UWB and other IEEE wireless standards. This has led to the invention of the wireless low data rate personal area networking technology IEEE 802.15.4, which has received tremendous attention from industry leaders and critics. However, unlike Bluetooth and Wi-Fi, which have a pipeline of 1Mbps, ZigBee devices are limited to a through-rate of 250kbps. The specified maximum range of operation for ZigBee devices is about 80m, substantially further than that used by IEEE 802.11.x or Bluetooth competent devices. For example, to stream a video file, the required bandwidth for a 320×200 colour video at 25 frames per second is 320×200×24×25=38.4 Mbps. Whilst with the IEEE 802.15.4 standard bandwidth limitation, approximately one uncompressed greyscale (8-bit) frame 256×100 pixels per second, excluding the protocol overhead, and reliable communication with no interference at speed above 115200 bps can be transmitted. This means even the compressed stream is far too much for the ZigBee data rate limit. Video transmission over IEEE 802.15.4 networks would be, therefore, difficult to achieve and this limitation could become a real problem if the user wishes to transmit a large amount of data in a very short period of time. On a positive note, due to ZigBee's low power output, ZigBee devices can sustain themselves on a small battery for many months, or even years, and their self-organising capability makes them ideal devices to introduce multimedia applications to transfer or live streaming. In this paper an adaptive scalable rate control technique is proposed in order to stream MPEG-4 video-codec over ZigBee. Rate control is widely used in video streaming, and has been widely studied, it refers to the scalability of quality, image size and frame rate, respectively in both the spatial and temporal model.

Below is a brief history of the many different pertinent pieces of research that have been conducted in rate control. Research has been carried out by Xie and Chia in layer rate control for high bitrate signal-to-noise-ratio (SNR) scalable video coding using MPEG-2 video-codec standard. They have proposed uniform scale quantization and rate distortion theories as well as a conventional linear rate model, for non-scalable coding [2]. More research was conducted by Zainaldin, Lambadaris and Nandy, which introduced an adaptive rate control, low bit-rate video transmission over wireless ZigBee for surveillance video application. Zainaldin and colleagues developed a rate control algorithm (RC-VBR) adapted to MPEG4 variable bit rate (VBR) video coders over Zigbee networks, which avoids the unpredictable rate variations of the VBR coding and removes the coding delay in constant bit-rate (CBR) [3]. Research has been carried out by Shafei, Rezaei, Tavakoli and Mohanna into variable bit rate video coding using fuzzy logic and they have proposed a new adaptive neuro-fuzzy inference system (ANFIS) for the video rate control algorithm [4]. According to these studies, adaptive scalable rate control can significantly improve the performance. Hence, an adaptive scalable rate control using Particle Swarm Optimization (PSO) technique is proposed in this paper. PSO is a relatively new AI technique, developed by Kennedy and Eberhart, and is a population-based stochastic optimization technique inspired by the social behaviour of birds. The algorithm is very simple but powerful [5][6]. The idea of PSO is that populations of potential solutions are intended to move collectively through a problem search space, under their respective algorithmic strategies, towards 'fitter' regions (represented by better solutions) and ideally to a solution representing the global optimum [7]. PSO adapts behaviour and looks for the best solution vector in the search space. A solution is called a particle; each particle has a velocity that directs the "flying" of a particle as well as a cost value and fitness that is evaluated and minimised by the function [7][8].

2 Methodology

The basic idea of rate control involves modifying the encoding parameters in order to maintain a target output bitrate. The most obvious parameter to vary is the Quantizer Parameter (QP) or step size, since increasing QP reduces coded bitrate, at the expense of lower decoded quality. Quantization has a significant impact on rate control, for example, by modifying the encoding parameters in order to maintain a target output bitrate. Commonly, this modification is made by setting the QP during the encoding to maintain the bitrate at the target bitrate. Optimising the tradeoff between bitrate and quality is a challenging task and many different approaches and algorithms have been proposed and implemented. The choice of rate control algorithm depends on the nature of the video application [9].

Scalar quantisation works with both the forward and inverse transform by dividing or multiplying the constant parameters. Different scalar quantisation (Q-scale) values influence the amount of compression. Q-scale value can be set for Prediction (P), Bidirectional (B) and Intra (I) frames separately on a scale of 1 to 31. The larger the number, the more the video will be compressed and therefore, the easier that video is to transmit, but the quality of the video will have been greatly affected. Therefore, the use of particle swarm optimization has been proposed and implemented in order to balance the bitrate and maintain good video quality. With PSO, the output rate of the encoder can be closely controlled during the encoding process as it determines the optimum Q-scale size in an ad-hoc way. This approach should eliminate any data loss and packet drops. Fig. 1 shows the process of the implementation of this idea and the main core functionality of this process. The adaptive quantization in this simulation uses the initial value of 15 for each Q-set to start with as is shown in Fig. 1. However, soon after running the simulation, it will override the value of its initial settings to decide adaptively on the required Q-sets to find the best transitions rates. The proposed intelligent system, which is based on CBR and PSO, should minimise data loss and distortion whilst ensuring that the decoder does not suffer from underflow or overflow.

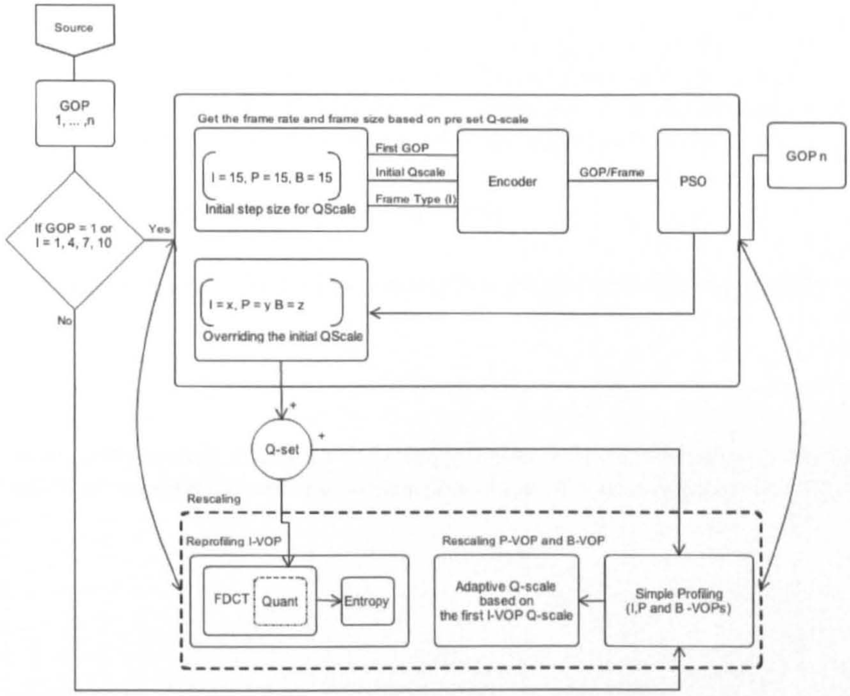


Fig. 1. Adaptive quantization process.

As shown in Fig. 1 during the processing of the first Group of Pictures (GOP), the I-frame bitrate and frame size is collected and given to the PSO system, in addition for each P-frame and B-frame the total bitrate and frame sizes are calculated to predict and to set the first QP steps to initialize the rest of the GOPs. The target bit rate is calculated based on the number of frames in the GOP and the minimum and maximum level of bits that are available by calculating the prediction P-frame rate. If the previous frame is an I-frame, it is used as a reference to predict the next frame's complexity and is allocated a suitable number of bits and subsequently the quantize step size Q for the following P and B frames is calculated. The desired bit rate or target rate is expressed in Equation (1):

$$Target\ rate = \left(\frac{Number\ of\ frames}{frame\ rate\ (24)}\right) ZigBee \tag{1}$$

Finding the bitrate of an uncompressed video using resolution and frame rate, and lossless video through approximations of quality is expressed in Equation (4.4):

$$Bitrate = (x \times y) \times \frac{\mathrm{MF}}{\mathrm{B}} \times Rate \tag{2}$$

where x is the frame width, with the value of the video file as 176 pixels, and y is the height with the value of 144 pixels. The number 4 is the value of MF (motion factor), which is divided by 8 bits, and therefore, 1000 is the value of rate which resulting bitrate of frame. The value of frame bitrate than is passed to the PSO for optimization. The optimization process is formulated as given in Equation (4.5) [10]:

$$\vec{v} \leftarrow w\vec{v} + \varphi_p r_p(\vec{p} - \vec{x}) + \varphi_g r_g(\vec{g} - \vec{x}) \tag{3}$$

where $\vec{x}$ denotes the current position of a particle travelling at velocity $\vec{v}$ and

$$\vec{x} \leftarrow \vec{x} + \vec{v} \tag{4}$$

The lower and upper boundaries of the Q-scales are 1 and 31 respectively. The lower and upper boundaries constitute the search-space and force the optimization method to move the candidate frame back to the boundary value if it has exceeded the boundaries that are denoted as $\vec{b}_{lo}$ and $\vec{b}_{up}$ as formulated in Equation (5):

$$f\colon [\vec{b}_{lo}, \vec{b}_{up}] \rightarrow \mathbb{R} \tag{5}$$

The evaluation of frame rates is then passed into the Rosenbrock function. The first input argument is the frame rates to be evaluated. Instead of iteratively recalculating the number of particles from the dimensionality of the position matrix, the information is passed to the function through the second input argument, which correspond to each frame row or rate that has been evaluated. Personal and global bests, including the best fitness, are updated based on how well they minimize Equation (4.9):

$$\sum_{d=1}^{D}(100(x_{d+1}-x_d^2)^2+(x_d-1)^2) \qquad \textbf{(6)}$$

After training the data the result of the PSO then determines the Q-Step size for each GOP.

3 Results

The best fitness of particles used at each GOP, in this simulation, is shown in Fig. 2. The desired Q-step values for 20 GOPs are represented with a minimum Q-scale of 26 and a maximum Q-scale of 31.

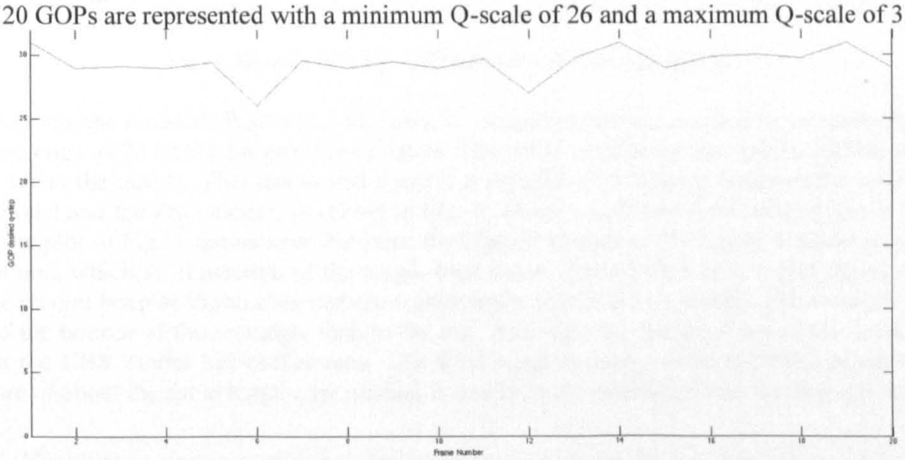


Fig. 2. Desired Q-step values for 20 GOPs

The PSO is applied in applications of MPEG-4 and the desired Q-scales value from 26 to 31 to stream MPEG-4 video. The lower number of Q-scale will result in higher bitrates within the range of ZigBee. The available bitrates and the higher frame complexity, and the maximum of 31 at target bitrates will lead to less quality at lower frame complexity. The ANOVA test, for the rate of frame rates, determined that there is a significant difference in the means between the three models. The ANOVA test of the null hypothesis $H_0: \mu_1 = \mu_2 = \mu_3$ is performed by comparing the observed F-value 25.63 at a 0.05 level of significance, with the p-value of 1.77543e-11; this tests the overall model to determine if there is a difference in means between methods. Just like before, since the p-value is small, it can be concluded that there is a significant difference between the 3 mean values. Fig. 3 shows that the median in the VBR boxplot is in the middle of the rectangle and the whiskers are about the same length. There are a large number of outliers, which is a point of concern as it shows that the data varies widely. The CBR median appears to be off-centre. The plot contains no outside values; however, the range of the data is not in our target bitrate. The third plot, which is the PSO plot, shows data that has less variation and spread than the other plots. The median of this model is approximately in the middle. There is no outside value and our data's range is in our target bitrate, just as before.

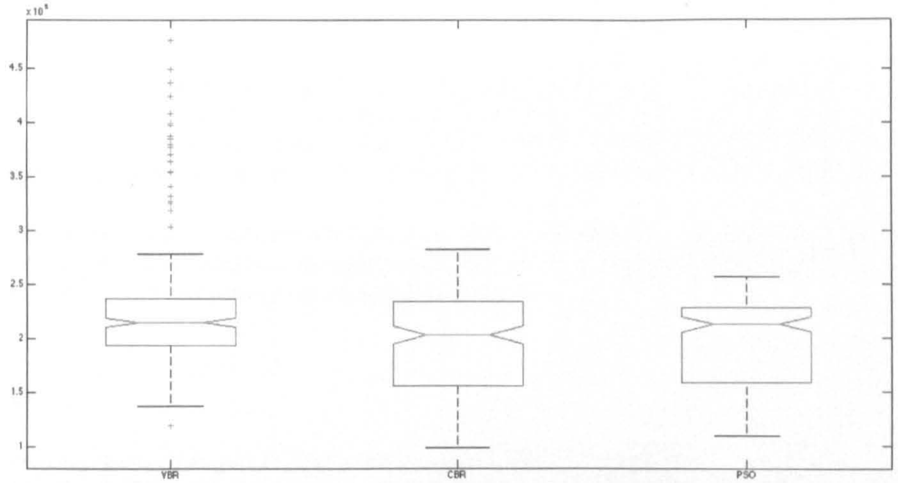


Fig. 3. Boxplot of VBR, CBR and PSO models side-by-side

In the simulation, the Kruskal–Wallis test was used as the nonparametric method to compare the medians of the frame rates of 20 GOPs for our three models. The small p-value in this test (6.08009e-06), shows that it can reject the null H. This shows that there is a significant difference between the medians using the VBR model and the PSO model, as shown in Fig. 4, where a different level of medians is illustrated. The VBR boxplot in Fig. 4 shows data that have the biggest variation. The upper whisker is longer than the bottom one, which is illustrative of the single high value. The median of this plot appears to be off centre. The second boxplot shows data that are significantly downwardly skewed. The median of this plot is closer to the bottom of the rectangle than to the top. Although the whiskers are of the same length, it shows that the CBR model has outlier data. The third boxplot presented is the PSO model where the whiskers are of about the same length, the median is nearly in the middle and all the data are in our target bitrate.

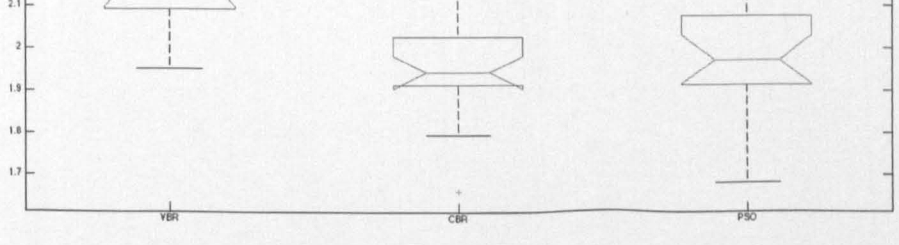


Fig. 4. Kruskal-Wallis boxplot for VBR, CBR and PSO

The result of the ANOVA test shows that the proposed model in this paper has less variation and spread, and no outside value. It can be concluded that the data range is in the target bitrate. Moreover, a nonparametric Kruskal-Wallis test determines that the CBR model has approximately the same variation as the VBR, and this indicates the high data rates in group of pictures for both models. However, the result for the idea in this paper shows that it also has the least variation and spread range of data rates in group of pictures. These results confirm that the use of particle swarm optimization makes our proposed model superior to the VBR and CBR methods as it increases the efficiency of the bandwidth, prevents data loss and most importantly it can improve quality of service and empower video within the MPEG-4 compression technique to be transmitted over IEEE 802.15.4 standard.

4 Conclusion

In this paper, a novel solution to Adaptive Scalable Rate Control to MPEG-4 applications over IEEE 802.15.4 is developed. The computer simulation results confirm that use of particle swarm optimization to develop an adaptive scalar quantization video coding, improves the quality of picture whilst reducing data loss and communication delay, when compared to conventional MPEG video transmissions.

The proposed model aims to achieve an optimum level of quality of picture whilst maintaining the ZigBee target bitrate. The adaptive quantization increases the available bandwidth, which leads to improvement in the quality of picture by reducing the data loss.

References

1. IEEE Computer Society, (August 31, 2007). IEEE Standard 802.15.4a (2007)
2. Xie, J., Chia, L.-T.: Enhancement layer rate control for high bitrate SNR scalable video coding. J. Vis. Commun. Image R. 16, pp. 159--179 (2005)
3. Zainaldin, A., Lambadaris, I., Nandy, B.: Adaptive Rate Control Low bit-rate Video Transmission over Wireless Zigbee Networks. Communications, 2008. ICC '08. IEEE International Conference on 19-23 May 2008, pp. 52--58 (2008)
4. Shafei, M., Rezaei, M., Tavakoli, S., Mohanna, F.: A Fuzzy Bit Allocation Algorithm for Variable Bit Rate Video Coding. Machine Vision and Image Processing (MVIP), 2011 7th Iranian, pp. 1--4 (2011)
5. Kennedy, J., Eberhart, R.C., Shi., Y.: Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco, CA (2004)
6. Kennedy, J., Eberhart, R.C.: Particle swarm optimization, Proc. IEEE Int'l Conf. on Neural Networks, IV, pp. 1942--1948. Piscataway, NJ: IEEE Service Center (1995)
7. Owen, A., Harvey, I.: Adapting Particle Swarm Optimization for Fitness Landscapes with Neutrality, 2007 IEEE Swarm Intelligence Symposium (2007)
8. Ji, C., Zhang, Y., Tong, M., Yang, S.: Particle Filter with Swarm Move for Optimization, Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X, pp. 909--918 (2008)
9. Richardson, I.E.G.: H.264 and MPEG-4 Video Compression, John Wiley & Sons Ltd, Chichester, England (2003)
10. Pedersen, M.E.H.: SwarmOps for Matlab - Numeric & Heuristic Optimization, November 2010, http://www.hvass-labs.org/projects/swarmops/matlab/ (2010)

Appendix III

SwarmOps for Matlab

Numeric & Heuristic Optimization
Source-Code Library for Matlab
The Manual

Revision 1.0

By

Magnus Erik Hvass Pedersen

November 2010

Contents

1. Introduction

SwarmOps is a source-code library for doing numerical optimization in Matlab and GNU Octave. It features popular optimizers which do not use the gradient of the problem being optimized. The Matlab version of SwarmOps differs from the C# and C versions in that it does not support meta-optimization, which is the use of one optimizer to tune the behavioural parameters of another optimizer.

1.1 Installation

To install SwarmOps unpack the archive to a directory. No further action is needed.

1.2 Updates

To obtain updates to the SwarmOps source-code library or to get newer revisions of this manual, go to the library's webpage at: www.hvass-labs.org

1.3 Manual License

This manual may be downloaded, printed and used for any personal purpose, be it commercial or non-commercial, provided the author(s) are not held responsible for your actions or any damage caused by your use of the manual. If you want to distribute the manual commercially, for example in a printed book or on a web-page that requires payment then you must obtain a license from the author(s).

2. What Is Optimization?

Solutions to some problems are not merely deemed correct or incorrect but are rated in terms of quality. Such problems are known as optimization problems because the goal is to find the candidate solution with the best, that is, *optimal* quality.

Fitness Function

SwarmOps works for real-valued and single-objective optimization problems, that is, optimization problems that map candidate solutions from n-dimensional real-valued spaces to one-dimensional real-valued spaces. Mathematically speaking we consider optimization problems to be functions f of the following form:

$$f: \mathbb{R}^n \rightarrow \mathbb{R}$$

In SwarmOps it is assumed that f is a minimization problem, meaning that we are searching for the candidate solution $\vec{x} \in \mathbb{R}^n$ with the smallest value $f(\vec{x})$. Mathematically this may be written as:

$$\text{Find } \vec{x} \text{ such that } \forall \vec{y} \in \mathbb{R}^n\text{: } f(\vec{x}) \leq f(\vec{y})$$

Typically, however, it is not possible to locate the exact optimum and we must be satisfied with a candidate solution of sufficiently good quality but perhaps not quite optimal. In this manual we refer to the optimization problem f as the fitness function but it is also known in the literature as the cost function, objective function, error function, quality measure, etc. We may refer to candidate solutions as positions, agents or particles, and to the entire set of candidate solutions as the search-space.

Maximization

SwarmOps can also be used with maximization problems. If $h: \mathbb{R}^n \rightarrow \mathbb{R}$ is a maximization problem then the equivalent minimization problem is: $f(\vec{x}) = -h(\vec{x})$

Boundaries

SwarmOps allows for a simple type of constraints, namely search-space boundaries. Instead of letting f map from the entire n-dimensional real-valued space, it is often practical to use only a part of this vast search-space. The lower and upper boundaries that constitute the search-space are denoted as $\vec{b}_{lo}$ and $\vec{b}_{up}$ so the fitness function is of the form:

$$f : [\vec{b}_{lo}, \vec{b}_{up}] \to \mathbb{R}$$

Such boundaries are typically enforced in the optimization methods by moving candidate solutions back to the boundary value if they have exceeded the boundaries.

3. Optimization Methods

This chapter gives brief descriptions of the optimization methods that are supplied with SwarmOps and recommendations for their use.

3.1 Choosing an Optimizer

SwarmOps for Matlab implements the following optimization methods:

Method	Filename	Parallel Version
Pattern Search (PS)	`ps.m`	-
Local Unimodal Sampling (LUS)	`lus.m`	-
Differential Evolution (DE)	`de.m`	`deparallel.m`
Particle Swarm Optimization (PSO)	`pso.m`	`psoparallel.m`
Many Optimizing Liaisons (MOL)	`mol.m`	`molparallel.m`

The first optimizer you may try when faced with a new optimization problem is the PS method which is often sufficient and has the advantage of converging (or stagnating) very quickly. PS also does not have any behavioural parameters that need tuning so it either works or doesn't. If the PS method fails at optimizing your problem you may want to try the LUS method from section 3.3 which sometimes works a little better than PS (and sometimes a little worse). You may need to run PS and LUS several times as they may converge to sub-optimal solutions. If PS and LUS both fail you will want to try the DE, MOL or PSO methods and experiment with their behavioural parameters.

As a rule of thumb PS and LUS stagnate rather quickly, say, after $40 \cdot n$ iterations, where n is the dimensionality of the search-space, while DE, MOL and PSO require substantially more iterations, say, $500 \cdot n$ or $2000 \cdot n$ and sometimes even more.

If these optimizers fail, you either need to tune their behavioural parameters using SwarmOps for C# or C, or use another optimizer altogether, e.g. CMA-ES.

3.2 Pattern Search (PS)

The optimization method known here as Pattern Search (PS) is originally due to Fermi and Metropolis as described in (1) and a similar method is due to Hooke and Jeeves (2). The implementation presented here is the variant from (3).

How it Works

PS uses one agent / position in the search-space which is being moved around. Let the position be denoted $\vec{x} \in \mathbb{R}^n$ which is initially picked at random from the entire search-space. The initial sampling range is the entire search-space: $\vec{d} = \vec{b}_{up} - \vec{b}_{lo}$. The potential new position is denoted $\vec{y}$ and is sampled as follows. First pick an index $R \in \{1, \dots, n\}$ at random and let $y_R = x_R + d_R$ and $y_i = x_i$ for all $i \neq R$. If $\vec{y}$ improves on the fitness of $\vec{x}$ then move to $\vec{y}$. Otherwise halve and reverse the sampling range for the R'th dimension: $d_R \leftarrow -d_R/2$. Repeat this a number of times.

3.3 Local Unimodal Sampling (LUS)

The LUS optimization method performs local sampling by moving a single agent around in the search-space with a simple way of decreasing the sampling range during optimization. The LUS method was presented in (3) (4).

How it Works

The agent's current position is denoted $\vec{x} \in \mathbb{R}^n$ and is initially picked at random from the entire search-space. The potential new position is denoted $\vec{y}$ and is sampled from the neighbourhood of $\vec{x}$ by letting $\vec{y} = \vec{x} + \vec{a}$, where $\vec{a} \sim U(-\vec{d}, \vec{d})$ is a random vector picked uniformly from the range $(-\vec{d}, \vec{d})$, which is initially $\vec{d} = \vec{b}_{up} - \vec{b}_{lo}$, that is, the full range of the entire search-space defined by its upper boundaries $\vec{b}_{up}$ and its lower boundaries $\vec{b}_{lo}$. LUS moves from position $\vec{x}$ to position $\vec{y}$ in case of improvement to the fitness. Upon each failure for $\vec{y}$ to improve on the fitness of $\vec{x}$, the sampling range is decreased by multiplication with a factor q:

$$\vec{d} \leftarrow q \cdot \vec{d}$$

where the decrease factor q is defined as:

$$q = \sqrt[\gamma n]{1/2} = \left(\frac{1}{2}\right)^{1/\gamma n}$$

where n is the dimensionality of the search-space and γ is a user-defined parameter used to adjust the rate of sampling-range decrease. A value of $\gamma = 3$ has been found to work well for many optimization problems.

3.4 Differential Evolution (DE)

The multi-agent optimization method known as Differential Evolution (DE) is originally due to Storn and Price (5). Many DE variants exist and the one implemented here is a basic variant known as DE/rand/1/bin.

How it Works

DE uses a population of agents. Let $\vec{x}$ denote the position of an agent being updated and which has been picked at random from the entire population. Let $\vec{y} = [y_1, \dots, y_n]$ be its new potential position computed as follows:

$$y_i = \begin{cases} a_i + F(b_i - c_i), & r_i < CR \vee i = R \\ x_i, & \text{else} \end{cases}$$

where the vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$ are the positions of distinct and randomly picked agents from the population. The index $R \in \{1, \ldots, n\}$ is randomly picked and $r_i \sim U(0,1)$ is also picked randomly for each dimension i. A move is made to the new position $\vec{y}$ if it improves on the fitness of $\vec{x}$. The user-defined parameters consist of the differential weight F, the crossover probability CR, and the population-size NP.

3.5 Particle Swarm Optimization (PSO)

The optimization method known as Particle Swarm Optimization (PSO) is originally due to Kennedy, Eberhart, and Shi (6) (7). It works by having a swarm of candidate solutions called particles, each having a velocity that is updated recurrently and added to the particle's current position to move it to a new position.

How it Works

Let $\vec{x}$ denote the current position of a particle from the swarm. Then the particle's velocity $\vec{v}$ is updated as follows:

$$\vec{v} \leftarrow \omega\vec{v} + \varphi_p r_p(\vec{p} - \vec{x}) + \varphi_g r_g(\vec{g} - \vec{x})$$

where the user-defined parameter ω is called the inertia weight and the user-defined parameters φ_p and φ_g are weights on the attraction towards the particle's own best known position $\vec{p}$ and the swarm's best known position $\vec{g}$. These are also weighted by the random numbers $r_1, r_2 \sim U(0,1)$. In addition to this, the user also determines the swarm-size S. In the SwarmOps implementation the velocity is bounded to the full range of the search-space so an agent cannot move farther than from one search-space boundary to the other in a single move.

Once the agent's velocity has been computed it is added to the agent's position:

$$\vec{x} \leftarrow \vec{x} + \vec{v}$$

3.6 Many Optimizing Liaisons

A simplification of PSO is called Many Optimizing Liaisons (MOL) and was originally suggested by Kennedy (8) who called it the "Social Only" PSO. The name MOL is due to Pedersen et al. who made more thorough studies (9). MOL differs from PSO in that it eliminates the particle's best known position $\vec{p}$. This has been found to improve performance somewhat and also makes it easier to tune the behavioural parameters.

4. Tutorial

4.1 Basics

After having unpacked the SwarmOps archive to a directory you can try executing the following commands in Matlab or Octave:

```
chdir ~/SwarmOps/ % or wherever you unpacked to.
molparameters; % load parameters for mol optimizer.
data = myproblemdata(2000); % create problem's data-struct.
[bestX, bestFitness, evaluations] = ...
   mol(@myproblem, data, MOL_DEFAULT) % perform optimization.
```

This example uses MOL to optimize the problem defined in the file myproblem.m. First the script molparameters.m is executed which defines behavioural parameters for MOL to be used in various optimization scenarios; we will just use a default choice of such parameters here. Next, a struct is created by calling the function myproblemdata(2000) which holds information about the search-space boundaries and dimensionality, maximum number of evaluations to perform, etc. The struct holds data that the optimizer needs and it may also hold data that the specific optimization problem needs. Finally, optimization is performed by calling the mol()-function with a handle to the myproblem()-function defined in the file myproblem.m, the data-struct just created, and the behavioural parameters for the MOL optimizer which will be used, here MOL_DEFAULT. This performs one optimization run and the output is the best-found position in the search-space, its fitness, and the number of evaluations actually used.

4.2 Custom Optimization Problem

To implement your own optimization problem modify one of the functions already implemented, e.g. myproblem.m or sphere.m, and remember to implement the data-struct creator as well, e.g. myproblemdata.m or spheredata.m.

4.3 Parallel Optimizers

There are basically two ways of parallelizing optimization; parallelizing the optimization problem or the optimization method. It is not always possible to parallelize an optimization problem and since multi-agent optimizers lend themselves well to parallelization, SwarmOps provides parallel versions of PSO, MOL and DE. These are invoked much the same way as their non-parallelized versions, e.g.:

```
matlabpool open 8; % Create 8 workers in matlab.
[bestX, bestFitness, evaluations] = ...
   molparallel(@myproblem, data, MOL_PAR_DEFAULT)
matlabpool close; % Close the worker pool.
```

The number of workers in the matlabpool should be related to the population size used by the optimizer and available on your computer, e.g. the behavioural parameters MOL_PAR_DEFAULT will allocate 32 particles for the MOL optimizer, so 2, 4, 8, 16, or 32 would be a good number of workers for matlabpool, but more workers will not be utilized.

Note that parallelized execution will only save time if the fitness function is time-consuming to compute, otherwise the overhead of distributing the computation will eliminate the time saving. Also note that GNU Octave does not support parallelism.

Bibliography

1. *Variable metric method for minimization.* **Davidon, W.C.** 1, s.l. : SIAM Journal on Optimization, 1991, Vol. 1, pp. 1-17.

2. *"Direct Search" solution for numerical and statistical problems.* **Hooke, R. and Jeeves, T.A.** 2, s.l. : Journal of the Association for Computing Machinery (ACM), 1961, Vol. 8, pp. 212-229.

3. **Pedersen, M.E.H.** *Tuning & Simplifying Heuristical Optimization (PhD Thesis).* s.l. : School of Engineering Sciences, University of Southampton, United Kingdom, 2010.

4. **Pedersen, M.E.H. and Chipperfield, A.J.** *Local Unimodal Sampling.* s.l. : Hvass Laboratories, 2008. HL0801.

5. *Differential evolution - a simple and efficient heuristic for global optimization over continuous space.* **Storn, R. and Price, K.** s.l. : Journal of Global Optimization, 1997, Vol. 11, pp. 341-359.

6. *Particle Swarm Optimization.* **Kennedy, J. and Eberhart, R.** Perth, Australia : IEEE Internation Conference on Neural Networks, 1995.

7. *A Modified Particle Swarm Optimizer.* **Shi, Y. and Eberhart, R.** Anchorage, AK, USA : IEEE International Conference on Evolutionary Computation, 1998.

8. *The particle swarm: social adaptation of knowledge.* **Kennedy, J.** Indianapolis, USA : Proceedings of the IEEE International Conference on Evolutionary Computation, 1997.

9. *Simplifying Particle Swarm Optimization.* **Pedersen, M.E.H. and Chipperfield, A.J.** s.l. : Applied Soft Computing, 2010, Vol. 10, pp. 618-628.

- Lund University
- Faculty of Engineering
- Automatic Control

TrueTime: Simulation of Networked and Embedded Control Systems

TrueTime is a Matlab/Simulink-based simulator for real-time control systems. **TrueTime** facilitates co-simulation of controller task execution real-time kernels, network transmissions, and continuous plant dynamics. Features of the simulator include

- Written in C++ MEX, event-based simulation
- External interrupts
- Possibility to write tasks as M-files or C++ functions. It is also possible to call Simulink block diagrams from within the code functions
- Network block (Ethernet, CAN, TDMA, FDMA, Round Robin, Switched Ethernet, FlexRay and PROFINET)
- Wireless network block (802.11b WLAN and 802.15.4 ZigBee)
- Battery-powered devices, Dynamic Voltage Scaling, and local clocks
- Stand-alone network interface blocks

From June 2010 the network parts of TrueTime are also available for Modelica using the Dymola 7.4 simulation tool from Dassault Systeme:

News

- NEW 2012-04-27
  TrueTime 2.0 beta 7 has been released. Minor bugs removed and compilation issues resolved.

Reference Manual

Anton Cervin, Dan Henriksson, Martin Ohlin: TrueTime 2.0 beta 5 - Reference Manual". Department of Automatic Control, Lund Univers Sweden, June 2010.

Software

TrueTime is Matlab-based and requires Matlab 7.0 (R14) with Simulink 6.0 (R14) or later. Control System Toolbox is required to run some of the examples.

TrueTime has been tested under Linux, Windows, and Mac, but will probably run on other platforms as well. Please note:

- The blocks and kernel functions may be compiled using gcc under Linux and Mac OS X or using Visual Studio C++ Express Edition under Windows XP or Windows Vista. Other C++ compilers should also work, but might require some small tweaks.
- If you have trouble compiling the MEX files using Visual Studio C++, see the following guide

TrueTime together with some examples can be downloaded as a compressed archive.

- Download TrueTime 2.0 beta 7 (zip archive including precompiled files for Windows and Mac). NOTE: sometimes crashes on 64-bit Matlab for unknown reasons.
- Download TrueTime 1.5 (compilation issues with Matlab R2011 and later)
- Release history
- License

- TrueTime Network for Modelica based on External C (direct download link) Available under the GPL license.
- TrueTime Network for Modelica (direct download link) Available under the Modelica 2 license.

Developers

The following people have contributed to TrueTime (in reverse chronological order):

- Martin Hast
- Martin Ohlin
- Dan Henriksson
- Anton Cervin
- Johan Eker

Main TrueTime Publication

If you write a paper and want to make a reference to TrueTime, please cite the following publication:

Anton Cervin, Dan Henriksson, Bo Lincoln, Johan Eker, Karl-Erik Årzén: "How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and TrueTime." *IEEE Control Systems Magazine,* **23:**3, pp. 16–30, June 2003.

```
import mpeg.*
%% end

qIPB=[];
minScale = 2;
maxScale = 31;

if (nargin<6)
    %Setting the initial Qscale
    q_init = [15 15 15];
else
    q_init = varargin{:};
end

for iGOP = iGOPs

    if (nargin>=5)

        ofname = [ofname_common, 'tmp4Ra2QGOP'];

        sumRaBits = 12/24 * Ra_desired; % Ra_desired alraedy in bit/s
        [FMrate fmsize] = MPEG_encoder(iGOP,avifname,q_init,ofname,I1);

        % Get the sum of P-frame rates
        tryFramBits = sum(fmsize);
        averageRate = tryFramBits + sum(FMrate)/2;

        % AI_q = pso.psoAi(sum(FMrate));
        AI_q = pso.psoAi(averageRate);

        if (AI_q > q_init(3))
            desireQScale = AI_q;
        else
            desireQScale = q_init(3);
        end

        q_init = [desireQScale desireQScale desireQScale];

        if ( sumRaBits>=tryFramBits & (abs(sumRaBits-tryFramBits) < 2e-
3*sumRaBits) | ((sumRaBits>tryFramBits) & (q_init==[minScale minScale
minScale])) | ((sumRaBits<tryFramBits) & (q_init==[maxScale maxScale
maxScale])) )
            data = tryFramBits;
            newAI_q = pso.psoAi(data);
            adoptedMaxData = (max(AI_q, (31 - newAI_q) + ceil(q_init(3)
/2)));
            if (adoptedMaxData <=31)
                desireQScale = adoptedMaxData;
                q_init = [desireQScale desireQScale desireQScale];
            elseif( adoptedMaxData > 31 & newAI_q > q_init(3))
                desireQScale = newAI_q;
                q_init = [desireQScale desireQScale desireQScale];
            end
```

```
            qIPB = q_init, return;
        end
    end

    % The first I-frame in next GOP is needed for the last two B-frames
in this GOP
    [qI, flag] = Ra2Q('i',Ra_desired,avifname,iGOP,[q_init(2:3)],
max(minScale, q_init(3)-3), min(maxScale, q_init(3)+3),
ofname_common,I1,q_init(3));
    if (flag<0 & qI<maxScale) % In case the preset q_max is too low
        %% desireQScale ~ 15
        [qI, flag] = Ra2Q('i',Ra_desired,avifname,iGOP,[15 ,
desireQScale], qI, maxScale, ofname_common,I1);
        %% desireQScale ~ 15
    elseif (flag>0 & qI>minScale) % In case the preset q_min is too
high
        [qI, flag] = Ra2Q('i',Ra_desired,avifname,iGOP,[15,
desireQScale], minScale, qI, ofname_common,I1);
    end

    [qP, flag] = Ra2Q('p',Ra_desired,avifname,iGOP,[qI, q_init(3)],
max(minScale, qI-3), min(maxScale, qI+3), ofname_common,I1, qI);
    if (flag<0 & qP<maxScale) % In case the preset q_max is too low
        [qP, flag] = Ra2Q('p',Ra_desired,avifname,iGOP,[qI,
desireQScale], qP, maxScale, ofname_common,I1);
    elseif (flag>0 & qP>minScale) % In case the preset q_min is too
high
        [qP, flag] = Ra2Q('p',Ra_desired,avifname,iGOP,[qI,
desireQScale], minScale, qP, ofname_common,I1);
    end

    [qB, flag] = Ra2Q('b',Ra_desired,avifname,iGOP,[qI, qP],
max(minScale, qP-3), min(maxScale, qP+3), ofname_common,I1, qP);
    if (flag<0 & qB<maxScale) % In case the preset q_max is too low
        [qB, flag] = Ra2Q('b',Ra_desired,avifname,iGOP,[qI, qP], qB,
maxScale, ofname_common,I1);
    elseif (flag>0 & qB>minScale) %In case the preset q_min is too high
        [qB, flag] = Ra2Q('b',Ra_desired,avifname,iGOP,[qI, qP],
minScale, qB, ofname_common,I1);
    end

    qIPB = [qIPB; qI, qP, qB];


end
```

```
function data = Q2RosenbrockData(dim, maxEvaluations, frameRate)
data = pso.benchmarkdata(dim, 0.001, maxEvaluations, 1, 31, -frameRate,
frameRate);
end
```

```
function bestQFitness = psoAi( frameRate )

pso.psoparameters

data =  pso.Q2RosenbrockData(4,100, frameRate); % create problemís
data-struct.

[bestX, bestFitness, evaluations] = ...
pso.psoparallel(@pso.rosenbrock, data, PSO_DEFAULT); % perform
optimization.

%abs(bestFitness)
%eturns arrays F and E. Argument F is an array of real values,
%usually in the range 0.5 <= abs(F) < 1. For real bestFitness,
%F satisfies the equation: bestFitness = F.*2.^E.
%Argument E is an array of integers that, for real bestFitness,
%satisfy the equation: bestFitness = F.*2.^E.

[F,E] = log2(bestFitness);

bestQFitness = E;
end
```

```
clear all; clc;

%% report format characters
newlineInAscii1 = [13 10];
spaceInInAscii = 32;
% for printing, newline causes much confusion in matlab and is provided
here as an alternative
newline = char(newlineInAscii1);
spaceChar = char(spaceInInAscii);

%% plot parameters
plotIndex = 1;
plotRowSize = 2;
plotColSize = 1;

%% read the image
targetFolder = './images';

IMG = 'original.jpg';   % IMG : originalImage
IMG = strcat(targetFolder, '/', IMG);
IMG = imread(IMG);
IMG = rgb2gray(IMG);
IMG = double(IMG);

%% noise parameters
sigma = 0.001;
offset = 0.01;

erosionFilterSize = 1;
dilationFilterSize = 1;
mean = 0;

noiseTypeModes = {
    'gaussian',          % [1]
    'salt & pepper',     % [2]
    'localvar',          % [3]
    'speckle',           % [4] (multiplicative noise)
    'poisson',           % [5]
    'motion blur',       % [6]
    'erosion',           % [7]
    'dilation',          % [8]
    % 'jpg compression blocking effect'   % [9]
    % [10] Interpolation/ resizing noise <to do>
    };

noiseChosen = 1;
noiseTypeChosen = char(noiseTypeModes(noiseChosen));

originalImage = uint8(IMG);

%% plot original
titleStr = 'Original';

imagePlot( originalImage, plotRowSize, plotColSize, ...
    plotIndex, titleStr );
```

```
plotIndex = plotIndex + 1;

%%
for i = 1:(plotRowSize*plotColSize)-1

    IMG_aforeUpdated = double(IMG);    % backup the previous state just
in case it gets updated.

    % returns the noise param updates for further corruption
    % IMG may be updated as the noisy image for the next round
    [IMG, noisyImage, titleStr, sigma, dilationFilterSize,
erosionFilterSize] = ...
        noisyImageGeneration(IMG, mean, sigma, offset,
dilationFilterSize, erosionFilterSize, noiseTypeChosen);

    imageQualityIndex_Value = imageQualityIndex(double(originalImage),
double(noisyImage));

    titleStr = [titleStr ',' newline 'Image Quaity Index: '
num2str(imageQualityIndex_Value)];

    imagePlot( noisyImage, plotRowSize, plotColSize, ...
        plotIndex, titleStr );
    plotIndex = plotIndex + 1;

end

if (~strcmp(char(class(noisyImage)), 'uint8'))
    disp('noisyImage is NOT type: uint8');
end

%% PSNR
psnr_Value = PSNR(originalImage, noisyImage);
fprintf('PSNR = +%5.5f dB \n', psnr_Value);
%% RMSE
[mse, rmse] = RMSE2(double(originalImage), double(noisyImage));
fprintf('MSE = %5.5f \n', mse);
fprintf('RMSE = %5.5f \n', rmse);
%% Universal Quality Index
imageQualityIndex_Value = imageQualityIndex(double(originalImage),
double(noisyImage));
fprintf('Universal Image Quality Index = %5.5f \n',
imageQualityIndex_Value);
%% PearsonCorrelationCoefficient
pcc = compute_PearsonCorrelationCoefficient (double(originalImage),
double(noisyImage));
fprintf('PearsonCorrelationCoefficient (originalImage vs noisyImage) =
%5.5f \n', pcc);
pcc = compute_PearsonCorrelationCoefficient (double(originalImage),
double(originalImage));
fprintf('PearsonCorrelationCoefficient (originalImage vs originalImage)
= %5.5f \n', pcc);

%% Signal to signal noise ratio, SNR
noise = double(noisyImage) - double(originalImage); % assume additive
```

```
noise
% check noise
noisyImageReconstructed = double(originalImage) + noise;
residue = noisyImageReconstructed - double(noisyImage);
if (sum(residue(:) ~= 0))
    disp('The noise is NOT relevant.');
end
snr_power = SNR(originalImage, noise);
fprintf('SNR = %5.5f dB \n', snr_power);
%% Mean absolute error, MAE
mae = meanAbsoluteError(double(originalImage), double(noisyImage));
fprintf('MAE = %5.5f \n', mae);
```

```
function PlotResult_PSO_GUI(GOP,Ra_frate,r_actual,Yf)

import mpeg.*

clear all;
clc;


%% end
% ==============================
% Data for bit rate

load ('./tx250kbps_VBR.mat')
%load ('./tx250kbps_VBR.mat')

f = (GOP-1)*12;
fNo = [1:1:f]'; % Generate frame number for ploting

% Bit rate plot
figure(1)

r(1:f) = r_actual/1000;

%%FR 240
subplot(3,1,1,'YGrid','on',...
    'XTick',[0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
170 180 190 200 210 220 230 240 250],...
    'XMinorTick','on');

hold on

plot(fNo,Ra_frate(1:f)./1000,'r','linewidth',1.5)
xlabel('Frame Number','fontsize',14)
ylabel('VBR Ra _a_c_t_u_a_l (kbit/s)','fontsize',14)
xlim([1 240])
ylim([0 500])

box on
hold all


load ('./tx250kbps_CBR.mat')
%load ('./tx250kbps_CBR.mat')

subplot(3,1,2,'YGrid','on',...
    'XTick',[0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
170 180 190 200 210 220 230 240 250],...
    'XMinorTick','on');
hold on

plot(fNo,Ra_frate(1:f)./1000,'r','linewidth',1.5)
xlabel('Frame Number','fontsize',14)
ylabel('CBR Ra _a_c_t_u_a_l (kbit/s)','fontsize',14)
xlim([1 240])
```

```
ylim([0 500])

box on
hold off


load ('./tx250kbps_PSO.mat')
%load ('./tx250kbps_PSO.mat')

subplot(3,1,3,'YGrid','on',...
    'XTick',[0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
170 180 190 200 210 220 230 240 250],...
    'XMinorTick','on');

hold on

plot(fNo,Ra_frate(1:f)./1000,'r','linewidth',1.5)
xlabel('Frame Number','fontsize',14)
ylabel('PSO Ra _a_c_t_u_a_l (kbit/s)','fontsize',14)
xlim([1 240])
ylim([0 500])

box on
hold off


%%

%%GOP

load ('./tx250kbps_VBR.mat')
%load ('./tx250kbps_VBR.mat')


% Data for bit rate
f = (GOP-1);
fNo = [1:1:f]'; % Generate frame number for ploting

figure(2)

subplot(3,1,1,'YGrid','on',...
    'XTick',[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]);

hold on
plot(fNo,Ragop(1:f)./1000,'r','linewidth',1.5)
xlabel('GOP Number','fontsize',14)
ylabel('VBR GOP Ra (kbit/s)','fontsize',14)
xlim([1 20])
ylim([0 300])

box on
hold off
```

```
load ('./tx250kbps_CBR.mat')
%load ('./tx250kbps_CBR.mat')


subplot(3,1,2,'YGrid','on',...
    'XTick',[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]);

hold on
plot(fNo,Ragop(1:f)./1000,'r','linewidth',1.5)
xlabel('GOP Number','fontsize',14)
ylabel('CBR GOP Ra (kbit/s)','fontsize',14)
xlim([1 20])
ylim([0 300])

box on
hold off


load ('./tx250kbps_PSO.mat')
%load ('./tx250kbps_PSO.mat')


subplot(3,1,3,'YGrid','on',...
    'XTick',[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]);

hold on
plot(fNo,Ragop(1:f)./1000,'r','linewidth',1.5)
xlabel('GOP Number','fontsize',14)
ylabel('PSO GOP Ra (kbit/s)','fontsize',14)
xlim([1 20])
ylim([0 300])

box on
hold off
%%
```

```
function PlotResult_PSO_GUI(GOP,Ra_frate,r_actual,Yf)
%% Import MPEG package
import mpeg.*

load ('./tx250kbps_PSO.mat')
%load ('./tx250kbps_PSO.mat')

f = (GOP-1)*12;
fNo = [1:1:f]'; % Generate frame number for ploting

% Bit rate plot
figure(1)

r(1:f) = r_actual/1000;

subplot(1,1,1,'YGrid','on',...
    'XTick',[0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
170 180 190 200 210 220 230 240 250],...
    'XMinorTick','on');

hold on
plot(fNo,Ra_frate(1:f)./1000,'r','linewidth',1.5)
xlabel('Frame Number','fontsize',11)
ylabel('Ra _a_c_t_u_a_l (kbit/s)','fontsize',11)
xlim([1 240])
ylim([0 500])
hold off
```

```
load('./../tryMPEG/WolfWithGaussian/Matrix/gBestPSO.mat')

% Data for bit rate
f = (GOP-1)*12;
fNo = [1:1:f]'; % Generate frame number for ploting
% Bit rate plot
figure(1)
%subplot(3,1,1)
hold on
plot(fNo,PbestGFitness(1:f),'r','linewidth',1.5)
xlabel('Frame Number','fontsize',11)
ylabel('gBest','fontsize',11)
xlim([1 240])
ylim([0 32])
hold off
figure(2)
%subplot(3,1,2)
plot(fNo,PbestQFitness(1:f),'r','linewidth',1.5)
xlabel('Frame Number','fontsize',11)
ylabel('Best Fitness','fontsize',11)
xlim([1 240])
ylim([0 32])
x = (GOP-1);
xNo = [1:1:x]'; % Generate frame number for ploting
figure(3)
%subplot(3,1,3)
plot(xNo,Qavg_desired(1:x),'r','linewidth',1.5)
xlabel('Frame Number','fontsize',11)
ylabel('GOP desired q-step','fontsize',11)
xlim([1 20])
ylim([0 32])
```

```
% ---------------------------------------------------------
% SwarmOps - Heuristic optimization for Matlab
% Copyright (C) 2003-2010 Magnus Erik Hvass Pedersen.
% Please see the file license.txt for license details.
% SwarmOps on the internet: http://www.Hvass-Labs.org/
% ---------------------------------------------------------

% Create data-struct for a benchmark problem.
% Parameters:
%     dim; the dimensionality of the search-space, e.g. 10.
%     acceptableFitness; stop optimization if this fitness is
%                        achieved.
%     maxEvaluations; the maximum number of fitness evaluations
%                     to perform in optimization.
%     lowerInit; initialization lower-boundary.
%     upperInit; initialization upper-boundary.
%     lowerBound; search-space lower-boundary.
%     upperBound; search-space upper-boundary.
% Returns:
%     data; the data-struct.
function data = benchmarkdata(dim, ...
                              acceptableFitness, ...
                              maxEvaluations, ...
                              lowerInit, upperInit, ...
                              lowerBound, upperBound)
    data = struct( ...
            'Dim', dim, ...
            'AcceptableFitness', acceptableFitness, ...
            'MaxEvaluations', maxEvaluations, ...
            'LowerInit', lowerInit*ones(1, dim), ...
            'UpperInit', upperInit*ones(1, dim), ...
            'LowerBound', lowerBound*ones(1, dim), ...
            'UpperBound', upperBound*ones(1, dim));
end

% ---------------------------------------------------------
```

```
% ------------------------------------------------------
% SwarmOps - Heuristic optimization for Matlab
% Copyright (C) 2003-2010 Magnus Erik Hvass Pedersen.
% Please see the file license.txt for license details.
% SwarmOps on the internet: http://www.Hvass-Labs.org/
% ------------------------------------------------------

% Enforce boundaries:
%     if x<lower then y=lower
%     pso.if x>upper then y=upper
%     pso. y=x
% The implementation below works for arrays as well.
% Parameters:
%     x; position to be bounded.
%     lower; lower boundary.
%     upper; upper boundary;
% Returns:
%     y; bounded position.
function y = bound(x, lower, upper)
    y = min(upper, max(lower, x));
end

% ------------------------------------------------------
```

```
% ------------------------------------------------------
% SwarmOps - Heuristic optimization for Matlab
% Copyright (C) 2003-2010 Magnus Erik Hvass Pedersen.
% Please see the file license.txt for license details.
% SwarmOps on the internet: http://www.Hvass-Labs.org/
% ------------------------------------------------------

% Initialize an agent with a uniformly random position.
% Parameters:
%     dim; dimensionality of search-space.
%     lower; lower boundary of search-space.
%     upper; upper boundary of search-space.
% Returns:
%     x; random position.
function x = initagent(dim, lower, upper)
    x = rand(1, dim).*(upper-lower) + lower;
end

% ------------------------------------------------------
```

```
% ------------------------------------------------------
% SwarmOps - Heuristic optimization for Matlab
% Copyright (C) 2003-2010 Magnus Erik Hvass Pedersen.
% Please see the file license.txt for license details.
% SwarmOps on the internet: http://www.Hvass-Labs.org/
% ------------------------------------------------------

% Initialize a population of agents with uniformly random
% positions.
% Parameters:
%     numAgents; number of agents in population.
%     dim; dimensionality of search-space.
%     lower; lower boundary of search-space.
%     upper; upper boundary of search-space.
% Returns:
%     x; random position.
function x = initpopulation(numagents, dim, lower, upper)
    % Preallocate array for efficiency.
    x = zeros(numagents,dim);

    for i=1:numagents
        x(i,:) = pso.initagent(dim, lower, upper);
    end
end

% ------------------------------------------------------
```

```matlab
% ------------------------------------------------------
% SwarmOps - Heuristic optimization for Matlab
% Copyright (C) 2003-2010 Magnus Erik Hvass Pedersen.
% Please see the file license.txt for license details.
% SwarmOps on the internet: http://www.Hvass-Labs.org/
% ------------------------------------------------------

% Behavioural parameters for Particle Swarm Optimization (PSO)
% tuned by Pedersen (1). The parameter-array consists of
% the following parameters:
% - Swarm-size (denoted s)
% - Inertia weight (denoted omega)
% - Particle's best weight (denoted phiP)
% - Swarm's best weight (denoted phiG)
%
% Select the parameters that most closely match the
% characteristics of your optimization problem.
% For example, if you want to optimize a problem where
% the search-space has 25 dimensions and you can perform
% 100000 evaluations, then you could first try using the
% parameters PSO_20DIM_40000EVALS. If that does not yield
% satisfactory results then you could try
% PSO_30DIM_60000EVALS or perhaps PSO_20DIM_400000EVALS_A.
% If that does not work then you will either need to tune
% the parameters for the problem at hand, or you should
% try using another optimizer.
%
% Literature references:
% (1) M.E.H. Pedersen.
%     Good parameters for Particle Swarm Optimization.
%     Technical Report HL1001, Hvass Laboratories, 2010.

% Parameters for non-parallel version:

PSO_2DIM_400EVALS_A       = [ 25,  0.3925,  2.5586, 1.3358];
PSO_2DIM_400EVALS_B       = [ 29, -0.4349, -0.6504, 2.2073];
PSO_2DIM_4000EVALS_A      = [156,  0.4091,  2.1304, 1.0575];
PSO_2DIM_4000EVALS_B      = [237, -0.2887,  0.4862, 2.5067];
PSO_5DIM_1000EVALS_A      = [ 63, -0.3593, -0.7238, 2.0289];
PSO_5DIM_1000EVALS_B      = [ 47, -0.1832,  0.5287, 3.1913];
PSO_5DIM_10000EVALS_A     = [223, -0.3699, -0.1207, 3.3657];
PSO_5DIM_10000EVALS_B     = [203,  0.5069,  2.5524, 1.0056];
PSO_10DIM_2000EVALS_A     = [ 63,  0.6571,  1.6319, 0.6239];
PSO_10DIM_2000EVALS_B     = [204, -0.2134, -0.3344, 2.3259];
PSO_10DIM_20000EVALS      = [ 53, -0.3488, -0.2746, 4.8976];
PSO_20DIM_40000EVALS      = [ 69, -0.4438, -0.2699, 3.3950];
PSO_20DIM_400000EVALS_A   = [149, -0.3236, -0.1136, 3.9789];
PSO_20DIM_400000EVALS_B   = [ 60, -0.4736, -0.9700, 3.7904];
PSO_20DIM_400000EVALS_C   = [256, -0.3499, -0.0513, 4.9087];
PSO_30DIM_60000EVALS      = [134, -0.1618,  1.8903, 2.1225];
PSO_30DIM_600000EVALS     = [ 95, -0.6031, -0.6485, 2.6475];
PSO_50DIM_100000EVALS     = [106, -0.2256, -0.1564, 3.8876];
PSO_100DIM_200000EVALS    = [161, -0.2089, -0.0787, 3.7637];
PSO_HANDTUNED             = [ 50,  0.7290,  1.4945, 1.4945];
PSO_DEFAULT               = PSO_20DIM_400000EVALS_A;

% Parameters for parallel version (above may also work):

PSO_PAR_5DIM_10000EVALS   = [ 72, -0.4031, -0.5631, 3.4277];
```

```
PSO_PAR_30DIM_60000EVALS  = [ 64, -0.2063, -2.7449, 2.3198];
PSO_PAR_DEFAULT           = PSO_PAR_5DIM_10000EVALS;

% ---------------------------------------------------------------
```

```
% ------------------------------------------------------
% SwarmOps - Heuristic optimization for Matlab
% Copyright (C) 2003-2010 Magnus Erik Hvass Pedersen.
% Please see the file license.txt for license details.
% SwarmOps on the internet: http://www.Hvass-Labs.org/
% ------------------------------------------------------

% Rosenbrock benchmark problem.
% Parameters:
%     x; position in the search-space.
%     data; data-struct for optimization problem.
% Returns:
%     fitness; the measure to be minimized.
function fitness = rosenbrock(x, data)
    fitness = 100*sum((x(1:end-1).^2 - x(2:end)).^2) + sum((x(1:end-1)-1).^2);
end

% ------------------------------------------------------
```

```
% ---------------------------------------------------------
% SwarmOps - Heuristic optimization for Matlab
% Copyright (C) 2003-2010 Magnus Erik Hvass Pedersen.
% Please see the file license.txt for license details.
% SwarmOps on the internet: http://www.Hvass-Labs.org/
% ---------------------------------------------------------

% Create data-struct for Rosenbrock problem.
% Parameters:
%     dim; the dimensionality of the search-space, e.g. 10.
%     maxEvaluations; the maximum number of fitness evaluations
%                     to perform in optimization.
% Returns:
%     data; the data-struct.
function data = rosenbrockdata(dim, maxEvaluations)
    data = benchmarkdata(dim, 0.001, maxEvaluations, 15, 30, -100, 100);
end

% ---------------------------------------------------------
```