REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique

End of studies project dissertation

For obtaining the state engineer diploma in Control Systems

On Maximum Power Point Tracking Quantum Algorithm

Realised by :
M. Habib Feraoun

Under the direction of :
Pr. Mohamed Tadjine (ENP)

Presented the 13 October 2022, in front of the jury :

Pr.	:	Omar STIHI	- President
Pr.	:	Mohamed TADJINE	- Promoter
Pr.	:	Djamel BOUDANA	- Examinator

ENP 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique

End of studies project dissertation

For obtaining the state engineer diploma in Control Systems

On Maximum Power Point Tracking Quantum Algorithm

Realised by :
M. Habib Feraoun

Under the direction of :
Pr. Mohamed Tadjine (ENP)

Presented the 13 October 2022, in front of the jury :

Pr.	:	Omar STIHI	- President
Pr.	:	Mohamed TADJINE	- Promoter
Pr.	:	Djamel BOUDANA	- Examinator

ENP 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique

Projet de Fin d'Etudes

Pour l'obtention du diplôme d'Ingénieur d'Etat en Automatique

Sur l'algorithme quantique de suivi du point de puissance maximale

Realisé par :
M. Habib Feraoun

Sous la direction de :
Pr. Mohamed Tadjine (ENP)

Présenté et soutenu publiquement le 13 October 2022, devant le jury :

Pr.	:	Omar STIHI	- Président
Pr.	:	Mohamed TADJINE	- Promoteur
Pr.	:	Djamel BOUDANA	- Examinateur

ENP 2022

Acknowledgment

First of all, we thank GOD Almighty Allah for giving us the courage and patience when developing this modest work. In particular, i offer my most sincere thanks and gratitude to my advisor Pr. Mohamed TADJINE , who helped me and gratified a large part of his attention, knowledge, scientific rigor and sensitivity, thank you for your help, your kindness and your support. These thanks also go to all those who participated in our training and guidance along the professional path in the past Three years.I would like to thank the professors who have honored to participate in the thesis jury, for the intention and the time allotted to the reading and judgment of this thesis. I also extend my deepest consideration to all those who, near or far, have contributed to the success of this work.

ملخص

لا شك أن الطاقة الشمسية ضرورية لعالم مثالي في المستقبل. طاقة نظيفة ومستدامة يمكن التحكم فيها باستخدام تقنية أشباه الموصلات من خلال الألواح الشمسية ، ومع ذلك يجب استغلالها على النحو الأمثل من خلال طريقة معينة تقودنا إلى الحوسبة وعلى وجه التحديد MPPT (تتبع نقطة الطاقة القصوى) على الرغم من تم تحسين مثل هذه الخوارزمية على مر السنين ومع أحدث نهج لحساب المعلومات الكميةْ من الممكن الحصول على MPPT الأمثل المطلق.

الغرض من هذا العمل هو تقديم الحوسبة الكمومية وتطوير نظريًا بعض خوارزميات الكم الشهيرة وتطبيق النهج الكمي لمشكلة الطاقة من خلال MPPT عن طريق تحسين سرب الجسيمات الكمومية باستخدام المحاكاة الكلاسيكية على كمبيوتر محلي والمحاكاة الاحتمالية الكمومية من IBM خلال كمبيوتر سحابة الكم مع مكتبة Qiskit

الكلمات الدالة:

الحوسبة ، الكم ، IBM ، Qiskit ، برمجة بايثون ، تحسين سرب الجسيمات ، البرمجة

Résumé

L'énergie solaire est sans aucun doute nécessaire pour un monde futur utopique. Une énergie contrôlable propre et durable utilisant la technologie des semi-conducteurs à travers des panneaux solaires, néanmoins elle doit être exploitée de manière optimale par une certaine méthode qui nous amène à l'informatique et précisément au MPPT (suivi du point de puissance maximum) bien que un tel algorithme a été amélioré au fil des ans et avec la nouvelle approche de calcul des «informations quantiques», il est possible d'avoir le MPPT optimal absolu.

Le but de ce travail est d'introduire l'informatique quantique et de développer théoriquement certains de ses célèbres algorithmes quantiques et d'appliquer l'approche quantique au problème de l'énergie via MPPT par l'optimisation de l'essaim de particules quantiques en utilisant la simulation classique sur un compilateur local et une simulation probabiliste quantique via IBM Ordinateur quantique cloud avec bibliothèque qiskit

Mots-clés: MPPT,informatique quantique,python,IBM qiskit,PSO,programmation

Summary

Solar energy is without doubt necessary for a utopian future world.clean,sustainable controllable energy using semiconductor technology through solar panels ,nevertheless it needs to be optimally exploited by a certain method which brings us to computing and precisely MPPT(maxmimum power point tracking) although such an algorithm has been improved through out the years and with the newest approach of computing 'quantum information' it is possible to have the absolute optimal MPPT.

The purpose of this work is to introduce quantum computing and develop theoretically some of it's famous Quantum Algorithms and applying the quantum approach to the energy problem through MPPT by the quantum particle swarm optimization using classical simulation on a local compiler and a quantum probabilistic simulation through IBM Cloud quantum computer with qiskit library

Keywords: MPPT,quantum computing,python,IBM qiskit,PSO,programming

List of Figures

Contents

Chapter 1

General Introduction

Solar energy is one of the most promising renewable energy sources due to its many advantages. Photovoltaic systems have experienced remarkable development in during the last decades. Like the maximum power point or MPP (Maximum Power Point) of a system photovoltaic varies according to changing environmental conditions, technologies and methods that track the maximum power point (Maximum Power Point Tracking) are necessary to obtain the maximum power from the photovoltaic systems. However, the supply voltage (P-V) curve shows multiple peaks in partial shade conditions. Under such conditions, many traditional methods of tracking maximum power points like disturbance and observation (disturbance and observation), as well as the incremental conductance, can become invalid because of the appearance of many local maxima.

Many advanced methods based on artificial intelligence such as networks of artificial neurons and control fuzzy logic can track the maximum power point global. However, they are not feasible in a complex environment as they require a elaborate training and broader experience (using machine learning and deep learning on massive data).

In addition ,One of the newest research interest is Quantum Computing and information .Recent physics Nobel prize winners present a scientific confirmation for the approach and theory in general ,researchers has been interested in both using the quantum attributes in classical algorithms or Trying to simulate directly in a powerful quantum computer to eventually enhance the current algorithms.

This work present an introduction of a quantum approach on a famous algorithm called particle swarm optimization applied to MPPT with comparaison, adding to it a quantum computer simulation based on a paper through IBM cloud quantum computer .

Chapter 2

Classical Computation

2.1 Introduction

- In all of the previous computation techniques in scientific domains The focus were always on the performance which covers the Information Capacity (Shanon's Law),Speed(Time of execution),Universality(Solving Problems and Tasks).Neverless the strength of actual supercomputers and computation techniques such as Artifitial intelligence,metaheuristic Algorithms : Genetic ,Bio inspired ..etc ,still there is some problems will never be solved unless the existence of a higher level computation

2.2 General Complexity

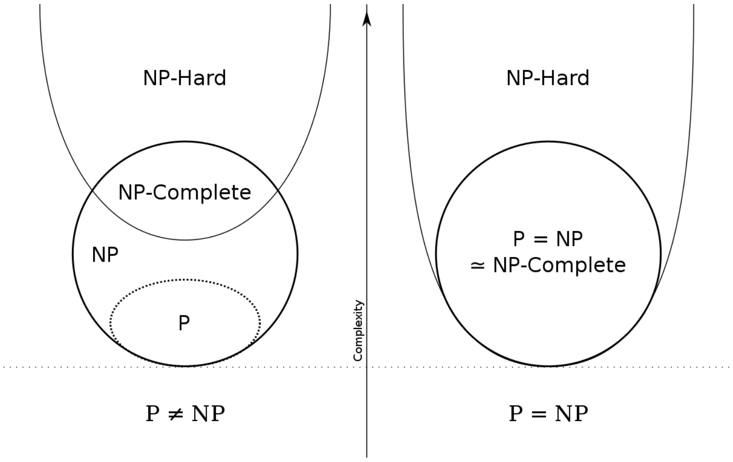


Figure 2.1: NP complexity

- In computational complexity theory, NP (nondeterministic polynomial time) is a complexity class used to classify decision problems. NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time by a deterministic Turing machine, or alternatively the set of problems that can be solved in

polynomial time by a nondeterministic Turing machine
Because of the many important problems in this class, there have been extensive efforts to find polynomial-time algorithms for problems in NP. However, there remain a large number of problems in NP that defy such attempts, seeming to require super-polynomial time. Whether these problems are not decidable in polynomial time is one of the greatest open questions in computer science (see P versus NP ("P=NP") problem for an in-depth discussion).
An important notion in this context is the set of NP-complete decision problems, which is a subset of NP and might be informally described as the "hardest" problems in NP. If there is a polynomial-time algorithm for even one of them, then there is a polynomial-time algorithm for all the problems in NP. Because of this, and because dedicated research has failed to find a polynomial algorithm for any NP-complete problem, once a problem has been proven to be NP-complete this is widely regarded as a sign that a polynomial algorithm for this problem is unlikely to exist.
However, in practical uses, instead of spending computational resources looking for an optimal solution, a good enough (but potentially suboptimal) solution may often be found in polynomial time. Also, the real life applications of some problems are easier than their theoretical equivalents.

2.3 Time Complexity

In computer science, the time complexity of an algorithm quantifies the amount of time taken by this algorithm to run as a function of the length of the string representing the input.
The time complexity of an algorithm is expressed using the notation O which excludes the terms with a lower degree taking into consideration only the term with the highest one.As anexample, if the time taken by an algorithm to run is $n^5 + 6n + 2$ then the time complexity will be $O(n^5)$

2.4 Hardware transition

The history of computing hardware starting at 1960 is marked by the conversion from vacuum tube to solid-state devices such as the transistor and later the integrated circuit. By 1959 discrete transistors were considered sufficiently reliable and economical that they made further vacuum tube computers uncompetitive. Computer main memory slowly moved away from magnetic core memory devices to solid-state static and dynamic semiconductor memory, which greatly reduced the cost, size and power consumption of computers.

The mass increase in the use of computers accelerated with 'Third Generation' computers. These generally relied on Jack Kilby's invention of the integrated circuit (or microchip), starting around 1966 in the commercial market. The basis of the fourth generation was the

Figure 2.2: Data General Nova

Figure 2.3: Intel 4004

invention of the microprocessor by a team at Intel.

Third generation minicomputers were essentially scaled-down versions of mainframe computers, whereas the fourth generation's origins are fundamentally different.

Microprocessor-based computers were originally very limited in their computational ability and speed, and were in no way an attempt to downsize the minicomputer. They were addressing an entirely different market.

Processing power and storage capacities have grown beyond all recognition since the 1970s, but the underlying technology has remained basically the same of large-scale integration (LSI) or very-large-scale integration(VLSI) microchips, so it is widely regarded that most of today's computers still belong to the fourth generation.

Chapter 3

Quantum Computing

3.1 Introduction

Information is stored, transmitted and processed by physical means. Thus, the concept of information and computation can be formulated in the context of a physical theory and the study of information requires ultimately experimentation. This sentence, innocuous at first glance, leads to non-trivial consequences.

Following Moore's law, about every 18 months microprocessors double their speed and, it seems, the only way to make them significantly faster is to make them smaller. In the not too distant future they will reach the point where the logic gates are so small that they consist of only a few atoms each. Then quantum-mechanical effects will become important. Thus, if computers are to continue to become faster (and therefore smaller), new, quantum technology must replace or supplement what we have now. But it turns out that such technology can offer much more than smaller and faster microprocessors. Several recent theoretical results have shown that quantum effects may be harnessed to provide qualitatively new modes of communication and computation, in some cases much more powerful than their classical counterparts.

This new quantum technology is being born in many laboratories. The last two decades have witnessed experiments in which single quantum particles of different kinds were controlled and manipulated with an unprecedented precision. Many "gedanken" experiments, so famous in the early days of quantum mechanics, have been carried out. New experimental techniques now make it possible to store and process information encoded in individual quantum systems. As a result we have a new, fledgling field of quantum information processing that represents a highly fertile synthesis of the principles of quantum physics with those of computer and information science. Its scope ranges from providing a new perspective on fundamental issues about the nature of physical law to investigating the potential commercial exploitation by the computing and communications industries. [1]

3.2 Basics of Quantum Computing

3.2.1 Qubits

Superposition

One of the main strength points of Quantum Computing is parallelism ,i.e: every atom can have both states with a probabilistic interpretation unlike the classical computation

$$c_1 |0\rangle + c_2 |1\rangle$$

where $c_1, c_2 \in \mathbb{C}$ and represent direction,magnitude of each configuration this concept was proven using the double slit experiment by young[2]

Overall Representation

A bit is the classical form of information and computation overall ,which leads to the idea of basing quantum computation on quantum bits called "Qubits"
Generally qubits are treated like mathematical objects specifically introducing the hilbert spaces ,In resume a qubit is just like classical bit which have two state 1 and 0 but in hilbert space ,plus the main difference is that it can have more states between 1 and 0 by having what it is called a "Superposition"
[1]

Algebraic notations

$x \in B$ such as $B = \{0, 1\}$ is equivalent in quantum dictionnary
$x \in H$ such as H is a Hilbert space
A Hilbert space is like vector space i.e: outer products , tensor product , orthonormality ,the main difference is the inner product which introduce the conjugate instead .A vector x in Hilbert space is denoted by the bra-ket notation $|x\rangle$ with states $|0\rangle$ and $|1\rangle$ as orthogonal basis vectors

Geometric representation

Complex representation

The state $|+\rangle$ represent the superposition of the state $|0\rangle$ and $|1\rangle$

$$|+\rangle = \alpha |0\rangle + \beta |1\rangle \tag{3.1}$$

in the figure $\alpha = \beta = \frac{1}{\sqrt{2}}$

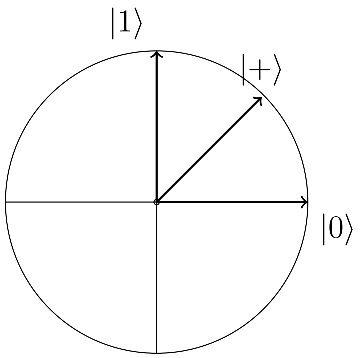


Figure 3.1: Complex representation

This state defines that we have probabilities for each state $|0\rangle$ and $|1\rangle$ equals to α^2 and β^2

Bloch sphere

In quantum mechanics and computing, the Bloch sphere is a geometrical representation of the pure state space of a two-level quantum mechanical system (qubit), named after the physicist Felix Bloch.

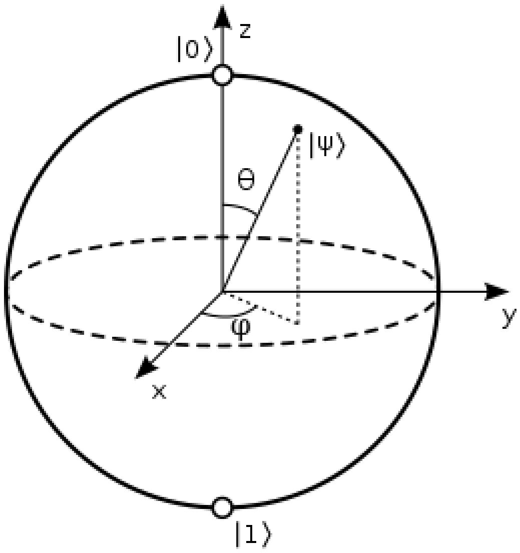


Figure 3.2: Bloch sphere

ϕ represents the wave function (quantum state) [1]

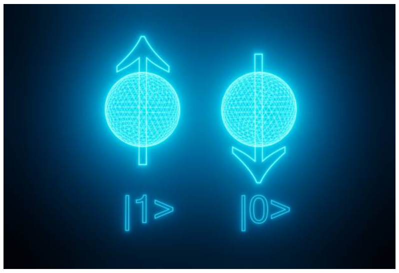


Figure 3.3: Entanglement

3.3 Multiple Qubits

3.3.1 Entanglement

“Hilbert Space is a big space”
Carlton Caves.
- Physically speaking it’s a group of particles where they interact and share a correlation in such a way that every particule’state is related to others’state the point of entangled states is the heart of quantum physics
An example for illustration :
Imagine you have two entangled particles which they are generated to have a zero spin in total , if one of the particles have positive direction in a specific axis then without measuring the other state , it’s in the negative direction

3.3.2 Formulation

Suppose we have two qubits. If these were two classical bits, then there would be four possible states, 00, 01, 10, and 11. Correspondingly, a two qubit system has four computational basis states denoted $|00\rangle, |01\rangle, |10\rangle, |11\rangle$A pair of qubits can also exist in superpositions of these four states, so the quantum state of two qubits involves associating a complex coefficient – sometimes called an amplitude – with each computational basis state, such that the state vector describing the two qubits is

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{3.2}$$

3.4 Mathematical model of quantum computing

3.4.1 Qubit

The qubit (quantum bit) - is the minimal unit of quantum information which describes the state of the simplest quantum system (just like 1 bit describes the state of the simplest classical system (fig. 2.4)).

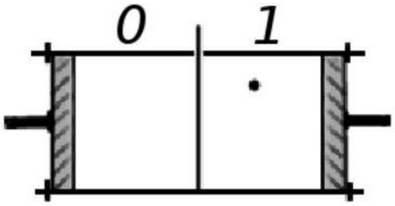


Figure 3.4: Szilard's engine

On the figure 2.5 you can see the quantum system where information is encoded by the polarization of a photon.

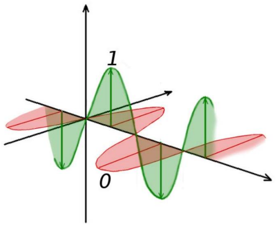


Figure 3.5: Quantum information

Mathematically qubit is a unitary vector in the 2-dimensional Hilbert's space. The real parts of coordinates in the example above can encode the angle of the photon polarization, while the imaginary parts - it's phase.

$$|\phi\rangle \in H, \quad \|\phi\| = 1, \quad \dim H = 2.$$

Hilbert's space is a vector space with inner product:

$$|x\rangle = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}, \quad |y\rangle = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}$$

$$\langle x \mid y \rangle = \sum_{i=1}^{n} x_i^* \cdot y_i$$

Brackets $|\cdots\rangle$ from now on will denote vectors (Dirac's notation in the name of the English physicist Paul Dirac, who invented and used it).

Inner (scalar) product allows us to define angles between vectors:

$$\cos\theta = \tfrac{|\langle x|y\rangle|}{\|x\|\cdot\|y\|},$$
$$\theta = \arccos\tfrac{|\langle x|y\rangle|}{\|x\|\cdot\|y\|}, \quad \theta \in \left[0, \tfrac{\pi}{2}\right]$$

In Hilbert's spaces all angles vary from 0 to $\pi/2$. Since qubits are unitary vectors, we can simplify by removing the norms:

$$\cos\theta = |\langle x \mid y\rangle|$$
$$\theta = \arccos|\langle x \mid y\rangle|$$

Orthogonal vectors have zero inner product:

$$|x\rangle \perp |y\rangle \Leftrightarrow \langle x \mid y\rangle = 0$$

The axes on the figure 2.5 are the projections of two orthogonal complex planes. Just as a photon (fig 2.4) a qubit can be in the infinite possible number of states.

Classical systems (like Szilard's engine) also can have enormous number of states but we choose to distinguish only few of them (digitization).

For quantum systems there's a similar concept - measurement.

3.4.2 Qubit Measurement

To obtain the information stored by a quantum system we have to measure it. To do that we have to choose an orthonormal basis in the system's state space. After measurement we obtain one of the vectors of this basis, and the system (subjectively for us) changes its state to this vector.

The probability of the basis vector to become our measurement outcome is defined by the coefficient before this vector in the state description before the measurement:

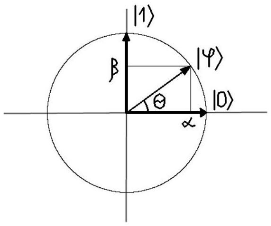


Figure 3.6: Qubit

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle,$$
$$\alpha, \beta \in \mathbb{C},$$
$$|\alpha|^2 + |\beta|^2 = 1$$
$$P(|0\rangle) = |\alpha|^2.$$
$$P(|1\rangle) = |\beta|^2.$$

Please note:

$$\begin{aligned}\alpha &= \langle 0 \mid \phi\rangle,\\ |\alpha| &= \cos\theta,\\ \beta &= \langle 1 \mid \phi\rangle,\\ |\beta| &= \sin\theta.\end{aligned}$$

In some sense measurement is similar to digitization - instead of the infinite number of possible states we again distinguish only two.

The choice of basis is important. Consider measuring the following qubit (which is either in the state $|\phi\rangle$ or $|\psi\rangle$ - we don't know which one is it.

$$\begin{aligned}|\phi\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\\ |\psi\rangle &= \frac{1}{\sqrt{2}}|1\rangle - \frac{1}{\sqrt{2}}|0\rangle.\end{aligned}$$

In the basis $|0\rangle, |1\rangle$ we obtain zero information about the state:

$$\begin{aligned}&P(|0\rangle \mid \phi) = |\langle\phi \mid 0\rangle|^2 = \frac{1}{2} = |\langle\psi \mid 0\rangle|^2 = P(|0\rangle \mid \psi),\\ &P(|1\rangle \mid \phi) = \frac{1}{2} = P(|1\rangle \mid \psi).\end{aligned}$$

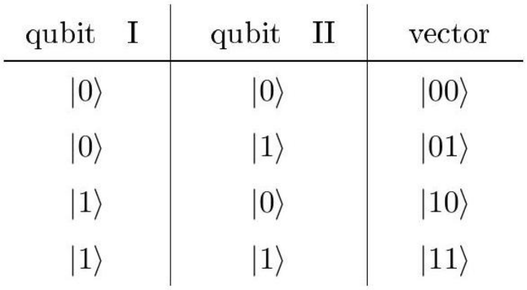
But with the Hadamard basis ($|+\rangle, |-\rangle$) we can find out which state was implemented by the system:

$$\begin{aligned}|+\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,\\ |-\rangle &= \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle,\\ P(|+\rangle \mid \phi) &= |\langle\phi \mid +\rangle|^2 = 1,\\ P(|+\rangle \mid \psi) &= |\langle\psi \mid +\rangle|^2 = 0\end{aligned}$$

3.4.3 Multiple qubits

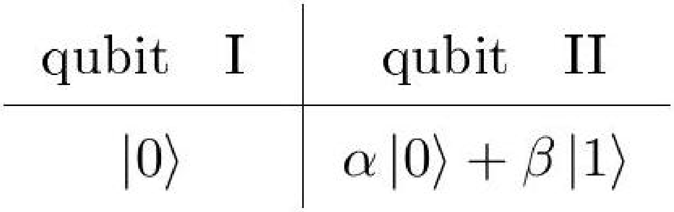
We are going to represent the composite state of several quantum systems as unitary vector in some Hilbert space. To do that we first introduce the basis of this new space:

qubit I	qubit II
$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$
$\|1\rangle$	$\|1\rangle$



qubit I	qubit II	vector
$\|0\rangle$	$\|0\rangle$	$\|00\rangle$
$\|0\rangle$	$\|1\rangle$	$\|01\rangle$
$\|1\rangle$	$\|0\rangle$	$\|10\rangle$
$\|1\rangle$	$\|1\rangle$	$\|11\rangle$

And then we describe this composite system in this basis:



qubit I	qubit II
$\|0\rangle$	$\alpha\,\|0\rangle + \beta\,\|1\rangle$

$$
\begin{aligned}
&P(|00\rangle) = |\alpha|^2,\\
&P(|01\rangle) = |\beta|^2,\\
&P(|10\rangle) = P(|11\rangle) = 0.\\
&|\alpha\gamma|^2 + |\alpha\delta|^2 + |\beta\gamma|^2 + |\beta\delta|^2 = 1.
\end{aligned}
$$

qubit I	qubit II	vector
$\alpha\|0\rangle + \beta\|1\rangle$	$\gamma\|0\rangle + \delta\|1\rangle$	$\alpha\gamma\|00\rangle + \alpha\delta\|01\rangle + \beta\gamma\|10\rangle + \beta\delta\|11\rangle$

Now we have to define how these new basis vectors are represented as columns:

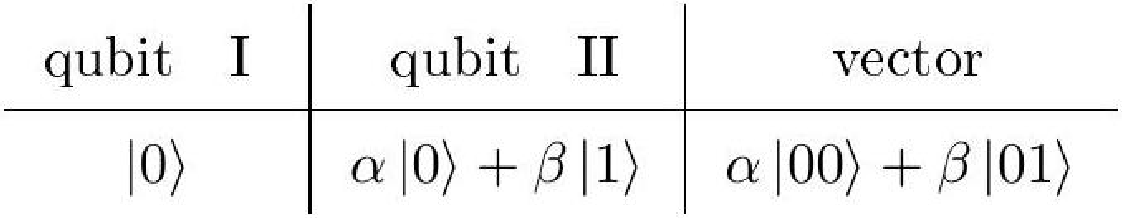
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \\ 0 \\ 1 \end{pmatrix}\Bigg) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$



qubit I	qubit II	vector
$\|0\rangle$	$\alpha\,\|0\rangle + \beta\,\|1\rangle$	$\alpha\,\|00\rangle + \beta\,\|01\rangle$

3.4.4 Measuring multiple qubits

[3] The most amazing thing about the introduced above way of describing multiple qubits systems is that any unitary vector in the constructed space describes some real quantum system that can be implemented on real particles.

$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle,$$
$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

But not all such vectors can be constructed as a tensor product of smaller systems. Consider for example the following set of states:

$$\alpha|00\rangle + \beta|11\rangle,$$
$$\alpha|01\rangle + \beta|10\rangle,$$
$$|\alpha|^2 + |\beta|^2 = 1.$$

These are the unitary vectors in a 4-dimensional Hilbert's space so they represent some quantum systems on 2 particles. But these vectors are not tensor products of 2-dimensional vectors, which means that those 2 particles don't have their separate states. The states of this kind are called the entangled states, and particles implementing these states - the entangled particles.

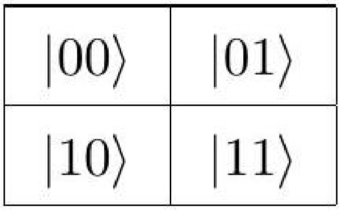
The composite systems can be measured just like the simple 1-qubit systems. To measure a system with 2 qubits we can measure each qubit separately. After measuring the first qubit (let it be $|0\rangle$ for example), we have the following description of the system:

$$|0\rangle \left(\frac{\alpha|0\rangle}{\sqrt{|\alpha|^2 + |\beta|^2}} + \frac{\beta|1\rangle}{\sqrt{|\alpha|^2 + |\beta|^2}} \right).$$

Measuring the first qubit doesn't alter the state of the second qubit, which is expectable, since those qubits reside on different particles, and when we measure the first particle we don't touch the second.

Now if we try to do the same thing with an entangled state, we can observe, that measuring of the first particle immediately assigns a state to the second particle (the spooky action at a distance).

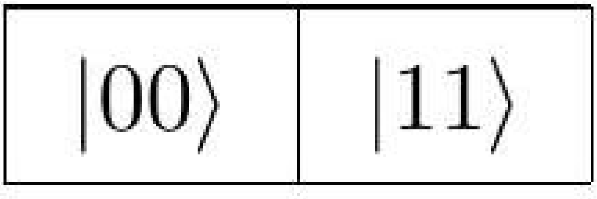
The many-worlds interpretation of quantum mechanics can describe this situation without any notion of action. For a not entangled state with 2 particles we have 4 different measurement outcomes, each existing in Multiverse:



$\|00\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$

The shares of the parts on this picture correspond to the squared coefficients . When we measure the first particle, our state splits on two, each observing only the half of this picture (the upper or the lower). The Multiverse for an entangled state looks like this:

There are only 2 types of universes and measurement of any particle defines for us subjectively the Universe we are in from now.



$\|00\rangle$	$\|11\rangle$

3.4.5 Quantum System Evolution

[3] The evolution of a quantum system can be described by a unitary operator. It means for us that any algorithm for processing quantum data is represented as a unitary operator in the state's space.

Operator U is unitary if and only if

$$UU^* = U^*U = I.$$

Another definition of unitarity:

$$\forall \phi \in H \quad \|U|\phi\rangle\| = \||\phi\rangle\|,$$
$$\forall \phi, \psi \in H \quad |\langle U \mid \phi\rangle|U|\psi\rangle\rangle| = |\langle \phi \mid \psi\rangle \ |$$

Examples.

3.4.6 Hadamard Tranform

[2] Unitarity of Hadamard transform:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H^* = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H,$$
$$H^*H = HH = \left(\frac{1}{\sqrt{2}}\right)^2 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = I.$$

Action of Hadamard transform on $|0\rangle$ and $|1\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle$$
$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle$$

Hadamard transform maps the basis $(|0\rangle, |1\rangle)$ to the Hadamard basis. Gate X

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Unitarity:

$$X^* = X,$$
$$XX = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = I.$$

Action:

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$
$$X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

Gate X is the quantum NOT gate.

3.5 Gate CNOT

CNOT is a 2-qubits gate:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Unitarity:

$$CNOT|00\rangle = |00\rangle,$$
$$CNOT|01\rangle = |01\rangle,$$
$$CNOT|10\rangle = |11\rangle,$$
$$CNOT|11\rangle = |10\rangle.$$

CNOT maps an orthonormal basis to an orthonormal basis, which means it's a unitary operator.

CNOT is the quantum conditional NOT operator.

To describe the quantum algorithms we are going to use the diagrams of the following type :

Horizontal lines are qubits (the most significant is on top), the operators are placed on these lines in the order of their application (from left to right). Figure 2.7 depicts the application of the Hadamard transform to 2 qubits $-|01\rangle$.

The matrix of this tranform is a 4×4 matrix. What is the look of this matrix?

$$H_2 = H \otimes H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} H & H \\ H & -H \end{pmatrix} =$$

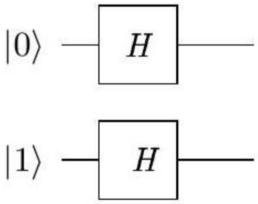


Figure 3.7: Quatum Algorithm

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

$$A|x\rangle \otimes B|y\rangle = (A \otimes B)|xy\rangle.$$

More examples:

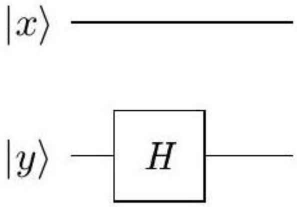


Figure 3.8: H gate on one qubit

$$I \otimes H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} H & 0 \\ 0 & H \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}.$$

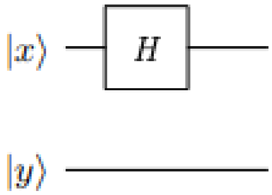


Figure 3.9: H gate on one qubit

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} I & I \\ I & -I \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}.$$

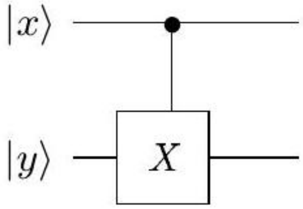


Figure 3.10: . CNOT, $|x\rangle-$ control, $|y\rangle$ - controlled

$$I \otimes CNOT = \begin{pmatrix} CNOT & 0 \\ 0 & CNOT \end{pmatrix}.$$

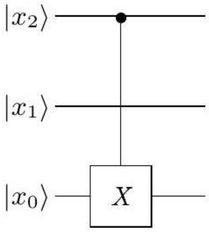


Figure 3.11: CNOT. 3 qubits

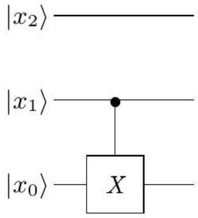


Figure 3.12: CNOT. 3 qubits

To discover the matrix let's use the following property:

$$Ae_k = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2k} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 1 \\ \cdots \\ 0 \end{pmatrix} = \begin{pmatrix} a_{1k} \\ a_{2k} \\ \cdots \\ a_{nk} \end{pmatrix}.$$

The operator from figure 2.12:

$$\begin{aligned} |000\rangle &\to |000\rangle, \\ |001\rangle &\to |001\rangle, \\ |010\rangle &\to |010\rangle, \\ |011\rangle &\to |011\rangle, \\ |100\rangle &\to |101\rangle, \\ |101\rangle &\to |100\rangle, \\ |110\rangle &\to |111\rangle, \\ |111\rangle &\to |110\rangle, \end{aligned}$$

$$\begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \end{pmatrix}.$$

3.5.1 Proof by induction

3.6 The Hadamard Transform for n qubits

$$H_n|x\rangle = \tfrac{1}{2^{n/2}} \textstyle\sum_{y=0}^{2^n-1} (-1)^{x \bullet y}|y\rangle,$$
$$x \bullet y = x_0y_0 \oplus x_1y_1 \oplus \cdots \oplus x_{n-1}y_{n-1}.$$

Base:

$$H_1|x\rangle = \frac{1}{2^{1/2}}\left(|0\rangle + (-1)^x|1\rangle\right) = \frac{1}{2^{1/2}}\sum_{y=0}^{1}(-1)^{x\bullet y}|y\rangle.$$

Induction step:

$$H_n|x\rangle = \frac{1}{2^{n/2}}\left(|0\rangle + (-1)^{x_{n-1}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + (-1)^{x_0}|1\rangle\right) =$$

$$= \frac{1}{\sqrt{2}}|0\rangle \otimes \frac{1}{2^{\frac{n-1}{2}}}\left(|0\rangle + (-1)^{x_{n-2}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + (-1)^{x_0}|1\rangle\right) +$$
$$+ \frac{1}{\sqrt{2}}(-1)^{x_{n-1}}|1\rangle \otimes \frac{1}{2^{\frac{n-1}{2}}}\left(|0\rangle + (-1)^{x_{n-2}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + (-1)^{x_0}|1\rangle\right) =$$
$$= \frac{1}{\sqrt{2}}|0\rangle \otimes H_{n-1}\left|x_{n-2}\cdots x_0\right\rangle + \frac{1}{\sqrt{2}}(-1)^{x_{n-1}}|1\rangle \otimes H_{n-1}\left|x_{n-2}\cdots x_0\right\rangle =$$
$$= \frac{1}{2^{n/2}}\Bigg(|0\rangle \otimes \sum_{y=0}^{2^{n-1}-1}(-1)^{x_0y_0\oplus x_1y_1\oplus\cdots\oplus x_{n-2}y_{n-2}}|y\rangle +$$
$$+(-1)^{x_{n-1}}|1\rangle \otimes \sum_{y=0}^{2^{n-1}-1}(-1)^{x_0y_0\oplus x_1y_1\oplus\cdots\oplus x_{n-2}y_{n-2}}|y\rangle\Bigg) =$$
$$= \frac{1}{2^{n/2}}\Bigg(\sum_{y=0}^{2^{n-1}-1}(-1)^{x_0y_0\oplus x_1y_1\oplus\cdots\oplus x_{n-2}y_{n-2}\oplus x_{n-1}\cdot 0}|0\rangle|y\rangle +$$
$$+ \sum_{y=0}^{2^{n-1}-1}(-1)^{x_0y_0\oplus x_1y_1\oplus\cdots\oplus x_{n-2}y_{n-2}\oplus x_{n-1}\cdot 1}|1\rangle|y\rangle\Bigg) =$$
$$= \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}(-1)^x \cdot y|y\rangle.$$

3.7 Deutsch Problem

3.7.1 Deutsch's Problem

Fromulation of the problem

The oracle (black box) function f maps 1 bit to 1 bit:

$$f : \{0,1\} \to \{0,1\}$$

There are only 4 functions of this type:

$$f(x) = 0$$
$$f(x) = 1$$
$$f(x) = x$$
$$f(x) = \bar{x}$$

The task is to find out which type of a function (constant or balanced) is implemented by the black box. The classical approach is to query the oracle twice - with inputs 0 and 1 , revealing not only the type of the function but the function itself.

Quantum Black Box

The definition of the quantum oracle, based on the function f :

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$
$$U_f\frac{1}{\sqrt{2}}|x\rangle(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}|x\rangle(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) =$$
$$= \frac{1}{\sqrt{2}}(-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)$$

Constant ($\mathbf{f}(\mathrm{x}) = \mathbf{0}$)

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus 0\rangle = |x\rangle|y\rangle$$
$$U_f = I$$

Constant ($\mathbf{f}(\mathbf{x}) = \mathbf{1}$)

$$|00\rangle \rightarrow |01\rangle,$$
$$|01\rangle \rightarrow |00\rangle,$$
$$|10\rangle \rightarrow |11\rangle,$$
$$|11\rangle \rightarrow |10\rangle,$$

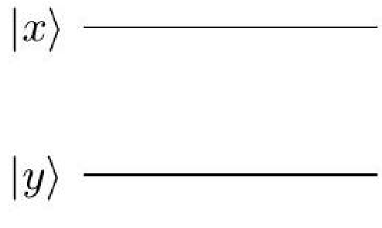


Figure 3.13: Circuit Scheme $U_f \cdot f(x) = 0$

Fig. 1. Circuit Scheme $U_f \cdot f(x) = 0$

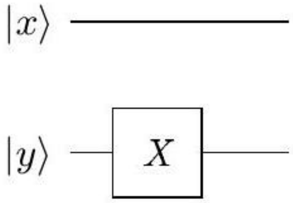


Figure 3.14: Circuit Scheme $U_f \cdot f(x) = 1$

Fig. 2. Circuit Scheme $U_f \cdot f(x) = 1$

$$U_f = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = I \otimes X$$

Balanced $(\mathbf{f}(\mathbf{x}) = \mathbf{x})$

$$|00\rangle \rightarrow |00\rangle,$$
$$|01\rangle \rightarrow |01\rangle,$$
$$|10\rangle \rightarrow |11\rangle,$$
$$|11\rangle \rightarrow |10\rangle,$$
$$U_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = CNOT$$

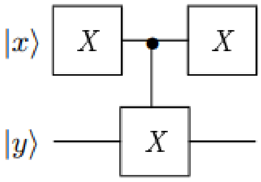


Figure 3.15: Circuit Scheme $U_f \cdot f(x) = x$ Balanced

Fig. 3. Circuit Scheme $U_f \cdot f(x) = x$ Balanced ($\mathbf{f}(\mathrm{x}) = \overline{\mathrm{x}}$) :

$$\begin{aligned} &|00\rangle \to |01\rangle, \\ &|01\rangle \to |00\rangle, \\ &|10\rangle \to |10\rangle, \\ &|11\rangle \to |11\rangle, \end{aligned}$$

$$U_f = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

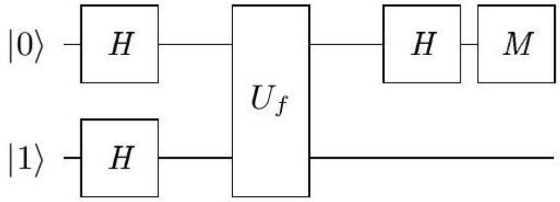


Figure 3.16: Circuit Scheme $U_f \cdot f(x) = \bar{x}$

3.7.2 Deutsch's Algorithm

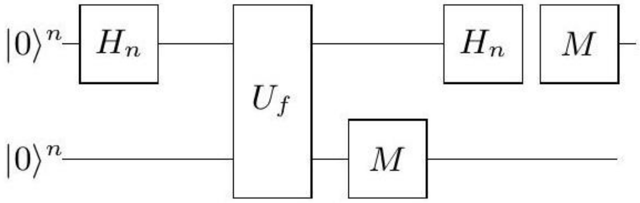


Figure 3.17: Deutsch's Algorithm

The algorithm action:

$$|01\rangle \xrightarrow{H_2} \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) =$$
$$= \frac{1}{2}|0\rangle(|0\rangle - |1\rangle) + \frac{1}{2}|1\rangle(|0\rangle - |1\rangle) \xrightarrow{U_f}$$
$$\xrightarrow{U_f} \frac{1}{2}(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + \frac{1}{2}(-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle) =$$
$$= \frac{1}{2}\left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right)(|0\rangle - |1\rangle).$$

If f is a constant, in the first qubit we have:

$$\pm\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)=\pm H(|0\rangle).$$

For the balanced function the same qubit will be:

$$\pm\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)=\pm H(|1\rangle).$$

So, measurement of the first qubit in the Hadamard basis gives us $|+\rangle$ for a constant function and $|-\rangle$ for a balanced one in just 1 query to the quantum oracle.

3.8 simon's problem

3.8.1 Simon's Problem

The Problem Formulation

An oracle function $f:\{0,1\}^n \to \{0,1\}^n$ has a period a in this sense:

$$\exists! a \neq 0 : \forall x f(x)=f(y) \iff y = x \oplus a.$$

We need to find a.

Classical approach needs in average 2^{n-2} oracle queries assuming we have the same amount of memory.

The quantum algorithm - figure 2.18

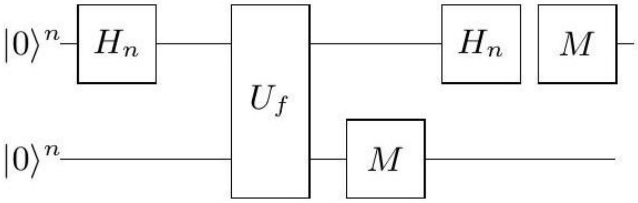


Figure 3.18: Simon's Algorithm

The algorithm action:

$$|0\rangle^n|0\rangle^n \xrightarrow{H_n} \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle|0\rangle^n \xrightarrow{U_f} \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle.$$

$$\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle = \frac{1}{2^{n/2}}\sum_{x\neq x\oplus a}(|x\rangle+|x\oplus a\rangle)|f(x)\rangle.$$

After the value register measurement:

$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus a\rangle)|f(x)\rangle.$$

The Hadamard transform on the input register:

$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus a\rangle) \xrightarrow{H_n} \frac{1}{2^{\frac{n+1}{2}}} \sum_{y=0}^{2^n-1} (-1)^{x \bullet y} |y\rangle + \frac{1}{2^{\frac{n+1}{2}}} \sum_{y=0}^{2^n-1} (-1)^{(x \oplus a) \bullet y} |y\rangle =$$
$$= \frac{1}{2^{\frac{n+1}{2}}} \sum_{y=0}^{2^n-1} \left((-1)^{x \bullet y} + (-1)^{x \bullet y \oplus a \bullet y}\right) |y\rangle =$$
$$= \frac{1}{\dots} \sum_{y: a \bullet y = 0} |y\rangle.$$

tion:

The sum contains only the vectors $|y\rangle$ which numbers satisfy the equation

$$a \bullet y = 0.$$

To find a we need n of linearly-independent y-s, so we need to run the algorithm several (O(n)) times.

3.9 Bernstein-Vazirani Problem

3.9.1 Bernstein-Vazirani Problem

The Problem Formulation

We know that an oracle function f is implemented like this:

$$f : \{0,1\}^n \to \{0,1\},$$
$$f(x) = a \bullet x.$$

We don't know the number a and our task is to find it.

The classical complexity - n oracle queries. The quantum complexity is just one quantum oracle query and the algorithm is the same as before .

$$\frac{1}{2^{\frac{n+1}{2}}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle(|0\rangle-|1\rangle)=$$
$$=\frac{1}{2^{\frac{n+1}{2}}}\sum_{x=0}^{2^n-1}(-1)^{a\bullet x}|x\rangle(|0\rangle-|1\rangle)=H_n|a\rangle\xrightarrow{H_n}$$
$$\xrightarrow{H_n}|a\rangle.$$

result.

After the measurement of the first n qubits we get $|a\rangle$ as the measurement

3.10 Grover algorithm

3.10.1 The Problem Formulation

An oracle function f maps n bit to n bit:

$$f:\{0,1\}^n\rightarrow\{0,1\}^n.$$
$$\exists!\omega:f(\omega)=a.$$

The task is to find ω.

It is more convenient for us to deal with function f_ω :

$$f_\omega(x)=\delta_{x=\omega}.$$

3.10.2 Quantum Oracle

$$U_f\frac{1}{\sqrt{2}}|x\rangle(|0\rangle-|1\rangle)=(-1)^{f(x)}\frac{1}{\sqrt{2}}|x\rangle(|0\rangle-|1\rangle).$$

The projection of U_f on the subspace of the argument ($|x\rangle$) :

$$U_\omega|x\rangle=(-1)^{f(x)}|x\rangle.$$
$$U_\omega=I-2|\omega\rangle\langle\omega|.$$
$$U_\omega|x\rangle=|x\rangle-2|\omega\rangle\langle\omega\mid x\rangle.$$

The Initial State

$$|s\rangle=H|0\rangle^n=\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle.$$

Operator U_s :

$$U_s=2|s\rangle\langle s|-I.$$

3.10.3 Grover's Iteration

$$R_{\text{grov}} = U_s U_\omega.$$

The action of the first Grover's iteration can be seen on figure 2.19. The vertical axis there is the desired $|\omega\rangle$, while the horizontal axis is the whole argument subspace except $|\omega\rangle$.

$$\sin\theta = |\langle s \mid \omega\rangle| = \frac{1}{2^{n/2}}.$$

Each Grover's iteration rotates the system in the direction $|\omega\rangle$ by the angle 2θ. Figure 2.20 shows the action of the second Grover's iteration.

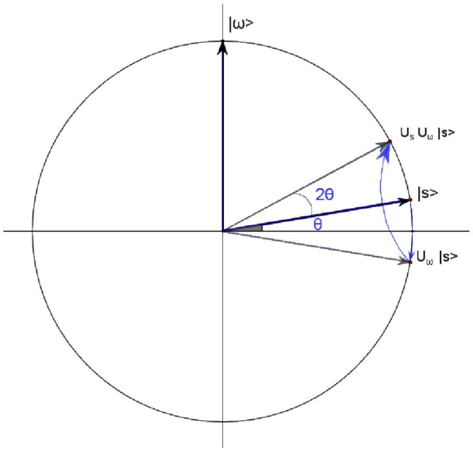


Figure 3.19: The first Grover's iteration

.[3]

The total number of iterations:

$$\theta + 2T\theta = \pi/2 \Longleftrightarrow T = \frac{\pi}{4\theta} - \frac{1}{2}.$$
$$T = \frac{\pi\sqrt{2^n}}{4} - \frac{1}{2} \approx \frac{\pi\sqrt{2^n}}{4}.$$

3.10.4 Grover's Algorithm Optimality

Let's consider the whole class of possible quantum algorithms which search for a special point of f_ω.[3]

1. $|\omega\rangle-$ the basis vector with number ω.

2. $U(\omega,t) = U_t U_\omega U_{t-1} U_\omega \cdots U_1 U_\omega-$ an algorithm with t calls of U_ω.

3. $|\psi_0\rangle -$ the initial state of the system.

4. $|\psi_t\rangle = U(\omega,t)\,|\psi_0\rangle$ - the state of the system after t calls of U_ω.

5. $T : U(\omega, T)\,|\psi_0\rangle = |\psi_T\rangle \approx |\omega\rangle$ - number of iterations (calls of U_ω) needed to get close enough to $|\omega\rangle$.

6. $|\psi_\omega\rangle = |\psi_T\rangle$.

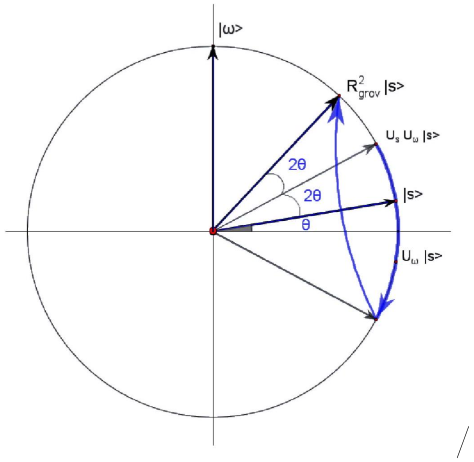


Figure 3.20: The second Grover's interation

Fig. 2. The second Grover's interation

8. $N = 2^n$. true:

We are going to find vector $|\phi\rangle$, for which the following double inequality is

$$4T^2 \geq \sum_{\omega=0}^{N-1} \|\,|\psi_\omega\rangle - |\phi\rangle\|^2 \geq 2N - 2\sqrt{N}.$$

Left and right part of the inequality don't depend on $|\phi\rangle$, thus:

$$4T^2 \geq 2N - 2\sqrt{N},$$

which tells us that for any algorithm $U(\omega, t)$ the number of iterations T needed to get close to $|\omega\rangle$ is not less than $O(\sqrt{N})$:

$$T \geq \sqrt{\frac{N}{2} - \frac{\sqrt{N}}{2}} \sim O(\sqrt{N}).$$

The Right Part For the different values ω vectors $|\psi_\omega\rangle$ form the basis which is almost orthonormal (we are going to consider it to be orthonormal):

$$\langle \psi_{\omega_i} \mid \psi_{\omega_j} \rangle =: \Delta_{i,j},$$
$$|\Delta_{i,j}| \approx 0$$

The right part :

$$\sum_{\omega=0}^{N-1} \| \left|\psi_\omega\right\rangle - \left|\phi\right\rangle \Bigg\|^2 = \sum_{\omega=0}^{N-1} \Bigg\| \psi_\omega \Bigg\|^2 + \sum_{\omega=0}^{N-1} \Bigg\| \phi\|^2 - \sum_{\omega=0}^{N-1} \langle \psi_\omega \mid \phi \rangle - \sum_{\omega=0}^{N-1} \langle \phi \mid \psi_\omega \rangle .$$

Let's decompose vector $|\phi\rangle$ in the basis $|\psi_\omega\rangle$:

$$|\phi\rangle = (x_0, x_1, x_2, \cdots x_{n-1}) ,$$
$$x_j = a_j + b_j i$$

From the previous we obtain:

$$\sum_{\omega=0}^{N-1} \| \left|\psi_\omega\right\rangle - |\phi\rangle \|^2 =$$
$$= 2N - \sum_{j=0}^{N-1} \left(x_j + x_j^*\right) = 2N - 2\sum_{j=0}^{N-1} \mathrm{Re}\,(x_j) = 2N - 2\sum_{j=0}^{N-1} a_j .$$

3.10.5 Lemma 1:

[3] For any set of real numbers $c_j, j = 1 \cdots K$

$$\left(\sum_{j=1}^{K} c_j\right)^2 \leq K \sum_{j=1}^{K} c_j^2$$

Proof:

$$\left(\sum_{j=1}^{K} c_j\right)^2 + \frac{1}{2}\sum_{i,j=1}^{K} (c_i - c_j)^2 =$$
$$= \sum_{i,j=1}^{K} c_i c_j + \frac{1}{2}K\sum_{i=1}^{K} c_i^2 + \frac{1}{2}K\sum_{j=1}^{K} c_j^2 - \sum_{i,j=1}^{K} c_i c_j =$$
$$= K\sum_{j=1}^{K} c_j^2$$

$$\left(\sum_{j=1}^{K} c_j\right)^2 + \frac{1}{2}\sum_{i,j=1}^{K} (c_i - c_j)^2 = K\sum_{j=1}^{K} c_j^2 \Longrightarrow$$
$$\Longrightarrow \left(\sum_{j=1}^{K} c_j\right)^2 \leq K\sum_{j=1}^{K} c_j^2 .$$

3.10.6 End of the proof.

$$\langle \phi \mid \phi \rangle = 1 \Longrightarrow \sum_{j=0}^{N-1} x_j x_j^* = 1 = \sum_{j=1}^{N-1} \left(a_j^2 + b_j^2 \right) \Longrightarrow \sum_{j=1}^{N-1} a_j^2 \leq 1.$$

From the inequality and Lemma 1:

$$N \geq N \sum_{j=1}^{N-1} a_j^2 \geq \left(\sum_{j=1}^{N-1} a_j \right)^2 \Longrightarrow \sqrt{N} \geq \sum_{j=1}^{N-1} a_j.$$

From above:

$$2N - 2 \sum_{j=0}^{N-1} a_j \geq 2N - 2\sqrt{N},$$

and the right part has been proved for any unitary vector $|\phi\rangle$.

The Left Part

$$U(\omega, t) = U_t U_\omega U_{t-1} U_\omega \cdots U_1 U_\omega.$$

Let's introduce vector $|\phi_t\rangle$:

$$|\phi_t\rangle = U_t U_{t-1} U_{t-2} \cdots U_1 |\psi_0\rangle .$$

Some auxiliary values:

$$E(\omega, t) := \| \left(U_\omega - I \right) |\psi_t\rangle \| = \| (I - 2|\omega\rangle\langle\omega| - I) |\psi_t\rangle \| = 2 \left| \langle \omega \mid \psi_t \rangle \right| .$$

$$F(t) := \| \, |\psi_t\rangle - |\phi_t\rangle \, \| =$$

$$= \| U_t U_\omega |\psi_{t-1}\rangle - U_t |\phi_{t-1}\rangle \|.$$

$$F(t) = \| U_t U_\omega |\psi_{t-1}\rangle - U_t |\psi_{t-1}\rangle + U_t |\psi_{t-1}\rangle - U_t |\phi_{t-1}\rangle \| \leq$$

$$\leq \| U_t \left(U_\omega - I \right) |\psi_{t-1}\rangle) \| + \| U_t \left(|\psi_{t-1}\rangle - |\phi_{t-1}\rangle \right) \| =$$

$$= \| \left(U_\omega - I \right) |\psi_{t-1}\rangle) \| + \| \, |\psi_{t-1}\rangle - |\phi_{t-1}\rangle \, \| = E(\omega, t-1) + F(t-1).$$

From above :

$$F(T) = \| \, |\psi_\omega\rangle - |\phi_T\rangle \, \| \leq 2 \sum_{t=1}^{T} |\langle \omega \mid \psi_{t-1} \rangle| \Longrightarrow$$

$$\Longrightarrow \| \, |\psi_\omega\rangle - |\phi_T\rangle \, \|^2 \leq 4 \left(\sum_{t=1}^{T} |\langle \omega \mid \psi_{t-1} \rangle| \right)^2 .$$

Let's apply Lemma 1:

$$F(T) \leq 4T \sum_{t=1}^{T} \left|\langle \omega \mid \psi_{t-1} \rangle\right|^2 .$$

$$\sum_{\omega=0}^{N-1} \| \, |\psi_\omega\rangle - |\phi_T\rangle \, \|^2 \leq 4T \sum_{\omega=0}^{N-1} \sum_{t=1}^{T} \left|\langle \omega \mid \psi_{t-1} \rangle\right|^2 =$$

$$= 4T \sum_{t=1}^{T} \sum_{\omega=0}^{N-1} \left|\langle \omega \mid \psi_{t-1} \rangle\right|^2 =$$

$$= 4T \sum_{t=1}^{T} 1 = 4T^2.$$

since $\sum_{\omega=0}^{N-1} \left|\langle \omega \mid \psi_{t-1} \rangle\right|^2$ is the squared norm of $|\psi_{t-1}\rangle$.

The left part of the inequality has been proved.

3.10.7 Are Quantum Computers Always Better?

Let's consider all functions which map n bits to 1 bit.

$$f(x) : \{0,1\}^n \to \{0,1\}$$

For each such function f we define the string x_f :

$$x(f) = x_{N-1} x_{N-2} \cdots x_0,$$
$$N = 2^n, x_i = f(i).$$

The quantum oracle U_f :

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle.$$
$$\forall i \in \{0, \cdots N-1\},$$
$$U_f |i\rangle |y\rangle = x_i |i\rangle |y \oplus 1\rangle + (1 - x_i) \, |i\rangle |y\rangle.$$

It appears that after T oracle queries each basis vector in the system state has a coefficient defined by a polynomial of power T of different x_i.

Let's denote $\tilde{x}_i = 1 - 2x_i$ and consider the "Parity" function:

$$\text{Parity}(\tilde{x}) = \prod_{i=0}^{N-1} \tilde{x}_i,$$

Now we are ready to formulate the task: having the quantum oracle U_f we need to determine the "Parity" of f.

Suppose that we have developed some effective quantum algorithm which solves this problem with less than $N/2$ oracle queries, thus overspeeding a classical computer at least two times.

After we ran this algorithm, each vector in the resulting state will have coefficient defined by a polynomial of x_i of power T. The probability for each of these vectors to become our measurement outcome is thus a polynomial of power $2T < N$. Some of these measurement outcomes indicate for us that f is "even". The probability of measuring any of them we denote P_{even}^{2T} .

Let's consider the following sum over all functions:

$$\sum_{\tilde{x}} P_{\text{even}}^{2T}\left(\tilde{x}\right) \prod_{i=0}^{N-1} \tilde{x}_i .$$

We can split the sum by any random index j : the first term - the sum over all f for which $f(j) = 0$, the second term - the sum over all f for which $f(j) = 1$:

$$(11) = \sum_{\tilde{x}:x_j\tilde{=}1} P_{even}^{2T}(\tilde{x}) \prod_{i \neq j} \tilde{x}_i - \sum_{\tilde{x}:x_j\tilde{=}-1} P_{even}^{2T}(\tilde{x}) \prod_{i \neq j} \tilde{x}_i .$$

Since $2T < N$, for any monomial M from P_{even}^{2T} we can choose index j for which x_j is not present in M. Thus for this M the sum is 0 :

$$\sum_{\tilde{x}:x_j\tilde{=}1} M(\tilde{x}) \prod_{i \neq j} \tilde{x}_i - \sum_{\tilde{x}:x_j\tilde{=}-1} M(\tilde{x}) \prod_{i \neq j} \tilde{x}_i =$$
$$= \sum_{\tilde{x}} M(\tilde{x}) \prod_{i \neq j} \tilde{x}_i - \sum_{\tilde{x}} M(\tilde{x}) \prod_{i \neq j} \tilde{x}_i = 0.$$

Since this is true for any monomial M, we conclude that the whole sum equals to 0.

Let's rewrite our discovery in a slightly different way:

$$0 = \sum_{\tilde{x}} P_{\text{even}}^{2T}\left(\tilde{x}\right) \prod_{i=0}^{N-1} \tilde{x}_i =$$
$$= \sum_{\tilde{x}:\tilde{x}\text{ even}} P_{\text{even}}^{2T}\left(\tilde{x}\right) - \sum_{\tilde{x}:\tilde{x}\text{ odd}} P_{\text{even}}^{2T}\left(\tilde{x}\right) \Longrightarrow$$

$$\Longrightarrow \sum_{\tilde{x}:\tilde{x}\text{ even}} P_{\text{even}}^{2T}\left(\tilde{x}\right) = \sum_{\tilde{x}:\tilde{x}\text{ odd}} P_{\text{even}}^{2T}\left(\tilde{x}\right).$$

So we can conclude that if the number of iterations T is less than $\frac{N}{2}$ then the probability to obtain the result which we interpret as "function is even" is equal for "even" and "odd" function. No speedup this time.

3.11 Shor algorithm

3.11.1 Factoring and the RSA

The RSA Algorithm

Let p and q be some big different prime numbers, $N = pq$. Euler's theorem:

$$\forall a < N : (a, N) = 1,$$
$$a^{\Phi(N)} = 1(N),$$

where $\Phi(N)$ is the Euler's function.

$$\Phi(N) = (p-1)(q-1).$$

If $e < N, GCD(e, N) = 1, \mathrm{GCD}(e, \Phi(N)) = 1$ then

$$\exists d < N : ed = 1(\Phi(N)),$$

or

$$\exists d, k : ed + \Phi(N)k = 1.$$

Expression above is the linear representation of $GCD(e, \Phi(N))$. We can find it with the Extended Euclidian algorithm.

$m < N, GCD(m, N) = 1$ - some secret message, $e, N-$ public key,

$d, N-$ private key.

Public key (e, N) can be widely shared while the numbers $d, \Phi(N), p, q$ must be kept in secret.

Encoding:

$$\text{encode }(m) = m^e(N),$$

Decoding:

$$\text{decode (encode }(m)) = (\text{encode}(m))^d(N) = \left(m^e\right)^d(N) =$$
$$= m^{ed}(N) = m^{1+\Phi(N)k}(N) = mm^{\Phi(N)k}(N) \overset{(1)}{=} m.$$

Example:

$$p = 11, q = 13,$$
$$N = pq = 143,$$
$$\Phi(N) = 10 \cdot 12 = 120,$$
$$e = 17.$$

The tuple $\{N, e\} = \{143, 17\}$ is the public key. Let's encode the message "7":

$$7^{17}(143) = 7 \cdot 49^8(143) = 7 \cdot 2401^4(143) = 7 \cdot 113^4(143) =$$

$$= 7 \cdot (-30)^4(143) = 7 \cdot 900^2(143) = 7 \cdot 42^2(143) = 7 \cdot 48(143) = 50(143).$$

(2) :

" 50 " is our encrypted message. To decrypt it we need to find d which satisfies

$$\begin{gathered} 17d = 1 + 120k \\ 17 \cdot 7 = 119 = 120 - 1 \\ 17 \cdot (-7) = 1 + 120 \cdot (-1), \\ d = -7 = 113(120) \end{gathered}$$

The tuple $\{N, d\} = \{143, 113\}$ is our private key:

$$50^{113} = 7(143).$$

Factoring and Period Finding

Shor's algorithm is designed to search for the period of a periodic function. Here we'll learn how it helps us with factoring.[4]

Again p and q are big different primes, $N = pq$. For a randomly chosen number $a < N$: $\text{GCD}(a, N) = 1$ we define function $f_a(x)$:

$$f_a(x) = a^x(N).$$

f_a is a periodic function, it's period $-r$ is the order of the element a in the ring $\mathbb{Z}_N$:

$$\begin{gathered} a^r = 1(N), \\ \forall r_1 < r \quad a^{r_1} \neq 1(N). \end{gathered}$$

Let's assume we have applied the Shor's algorithm and found r for f_a.

If the number r happens to be even,

$$r = 0(2).$$

then

$$\begin{gathered} a^r - 1 = 0(N) \\ \left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right) = Nk \end{gathered}$$

$\left(a^{r/2} - 1\right) \neq 0(N)$, since $a^{r/2} \neq 1(N)$. Otherwise $r/2$ would be the order of a.

If, even more,

$$\left(a^{r/2}+1\right) \neq 0(N)$$

then $\left(a^{r/2}-1\right)\left(a^{r/2}+1\right)$ is a multiple of N, but neither $\left(a^{r/2}-1\right)$ nor $\left(a^{r/2}+1\right)$ are multiples of N separately. So:

$$p, q = GCD\left(a^{r/2} \pm 1, N\right).$$

How likely the previous conditions are met for a random number a ? Let's consider a_1 and a_2 :

$$a_1 = a(p),$$
$$a_2 = a(q)$$

r_1 and r_2 are the orders of a_1 and a_2 in the rings $\mathbb{Z}_p$ and $\mathbb{Z}_q$.

$$a_1^r(p) = a^r(p) = 1$$

because $a^r = 1(pq)$. Since r_1 is the order of a_1 in $\mathbb{Z}_p$, the number r is a multiple of r_1. The same reasoning for $\mathbb{Z}_q$ gives us r is a multiple of r_2.

For any number s which is multiple of both r_1 and r_2, it is also a multiple of r :

$$a^s = pk_1 + 1,$$
$$a^s = qk_2 + 1,$$

since s is a multiple of both r_1 and r_2.

$$pk_1 = qk_2.$$

p is coprime to q, so k_1 is a multiple $q(7)$. We can now rewrite :

$$a^s = pq\frac{k_1}{q} + 1 \Longrightarrow a^s = 1(pq) \Longrightarrow r \mid s.$$

Since any number which is a multiple of both r_1 and r_2, is also a multiple of r, we conclude that

$$r = \mathrm{LCM}\left(r_1, r_2\right).$$

Let's decompose r_1 and r_2 as a multiplication of some power of 2 and some odd factor:

$$r_1 = 2^{c_1} \text{ odd }_1, c_1 \geq 0$$
$$r_2 = 2^{c_2} \text{ odd }_2, c_2 \geq 0$$

If the numbers c_1 and c_2 are different (for example if $c_1 > c_2$), then both the conditions are met:

$$r = 2r_2int,$$

$$a^{r/2} = a^{r_2int} = 1(q) \Longrightarrow a^{r/2} \neq -1(q) \Longrightarrow a^{r/2} \neq -1(pq).$$

The numbers c_1 and c_2 are defined by the random choice of a. Let's estimate the likelihood of them being different $(c_1 \neq c_2)$:

$$p = 2^{k_p}s_p + 1, \quad s_p = 1(2).$$

Let b be the primitive element of the field $\mathbb{Z}_p$, different powers of b generate the whole field. The order of b is $2^{k_p}s_p$. The random choice of a defines the random choice of a_1, thus the random choice of $m_p \in \left\{1, \cdots, 2^{k_p}s_p\right\}$:

$$a_1 = b^{m_p}(p).$$

Since r_1 is the order of a_1 :

$$a_1^{r_1}(p) = 1 \Longrightarrow b^{m_p \cdot r_1} = 1 = b^{2^{k_p}s_p} \Longrightarrow m \cdot r_1 = 0\left(2^{k_p}s_p\right).$$

It appears that $m_p r_1$ is a multiple of 2^{k_p}, with m_p chosen randomly from $\left\{1, \cdots, 2^{k_p}s_p\right\}$. The same reasoning for $\mathbb{Z}_q$ gives us that for some other number m_q from $\left\{1, \cdots, 2^{k_q}s_q\right\}$ also defined by the random choice of $a, m_q r_2$ is a multiple of 2^{k_q}. This means that the powers of 2 in r_1 and r_2 are defined by the powers of 2 in two random numbers $-m_p$ and m_q. The probability of coincidence for r_1 and r_2 to have the same power of 2 in them has it's maximum when $k_q = k_p = 1$. In this case this probability is $\frac{1}{2}$ which is equal to the probability for two random numbers to have the same oddity.

This all means that with the high enough probability (at least $\frac{1}{2}$) we can successfully factor N if we find the period r of the function f_a defined by some random number a.

Quantum Fourier Transform

The Shor's algorithm uses an operator we didn't meet before - the Quantum Fourier Transform, QFT :

$$N := 2^n,$$

$$QFT|x\rangle = \frac{\|x\|}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{xy}{N}} |y\rangle.$$

QFT unitarity

$$\begin{aligned}\|QFT|x\rangle\|^2 &= \frac{\|x\|^2}{2^n}\Bigg\langle \sum_{y=0}^{2^n-1} e^{2\pi i\frac{xy}{N}} \mid y \Bigg\rangle \Bigg| \sum_{y=0}^{2^n-1} e^{2\pi i\frac{xy}{N}} \Bigg| \, y \Bigg\rangle \Bigg\rangle = \\ &= \frac{\|x\|^2}{2^n}\sum_{y=0}^{2^n-1} e^{(-2\pi i+2\pi i)\frac{xy}{N}} \langle y \mid y\rangle = \\ &= \frac{\|x\|^2}{2^n}\sum_{y=0}^{2^n-1} 1 = \frac{\|x\|^2}{2^n}2^n = \|x\|^2.\end{aligned}$$

QFT doesn't alter vector's norm. From now on we are going to omit the norm in (8), since we deal only with unitary vectors.

$$\|x_1\| = \|x_2\| = 1, \langle x_1 \mid x_2\rangle = 0$$

$$\langle QFT \mid x_1\rangle \, |QFT|x_2\rangle\rangle = \frac{1}{2^n}\Bigg\langle \sum_{y_1=0}^{2^n-1} e^{-2\pi i\frac{x_1y_1}{N}} \mid y_1 \Bigg\rangle \Bigg| \sum_{y_2=0}^{2^n-1} e^{2\pi i\frac{x_2y_2}{N}} \Bigg| \, y_2 \Bigg\rangle \Bigg\rangle =$$

$$= \frac{1}{2^n}\sum_{y_1=y_2} \Big\langle e^{-2\pi i\frac{x_1y_1}{N}} \mid y_1 \Big\rangle \Big| e^{2\pi i\frac{x_2y_2}{N}} \Big| \, y_2 \Big\rangle\Big\rangle + \frac{1}{2^n}\sum_{y_1\neq y_2} \Big\langle e^{-2\pi i\frac{x_1y_1}{N}} \mid y_1 \Big\rangle \Big| e^{2\pi i\frac{x_2y_2}{N}} \Big| \, y_2 \Big\rangle\Big\rangle =$$

$$= \frac{1}{2^n}\sum_{y_1=y_2} \Big\langle e^{-2\pi i\frac{x_1y_1}{N}} \mid y_1 \Big\rangle \Big| e^{2\pi i\frac{x_2y_2}{N}} \Big| \, y_2 \Big\rangle\Big\rangle = \frac{1}{2^n}\sum_{y=0}^{2^n-1} e^{2\pi i\frac{(x_2-x_1)y}{N}}.$$

The expression for the sum of geometrical progression:

$$\sum_{y=0}^{2^n-1} a^y = \frac{a^{2^n}-1}{a-1}.$$

By combining previous results we obtain:

$$(9) = \frac{e^{2\pi i(x_2-x_1)}-1}{e^{2\pi i\frac{(x_2-x_1)}{N}}-1} = \frac{1-1}{e^{2\pi i\frac{(x_2-x_1)}{N}}-1} = 0.$$

QFT maps orthonormal basis to orthonormal basis, so it's unitary..[4]

QFT Implementation

$$\begin{aligned}QFT|x\rangle = \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1} e^{2\pi i\frac{xy}{N}}|y\rangle &= \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1} e^{2\pi i\frac{xy(N)}{N}+k\cdot 2\pi i}|y\rangle = \\ &= \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1} e^{2\pi i\frac{xy(N)}{N}}|y\rangle.\end{aligned}$$

$xy(N) - xy$ modulo N.

Let's write down the binary representations of x and y :

$$x = x_0 + x_1 2 + x_2 2^2 + \cdots + x_{n-1} 2^{n-1},$$
$$y = y_{n-1} 2^{n-1} + y_{n-2} 2^{n-2} + \cdots + y_1 2 + y_0.$$
$$\frac{xy(N)}{N} = y_{n-1}\frac{x_0}{2} + y_{n-2}\left(\frac{x_0}{4} + \frac{x_1}{2}\right) + \cdots + y_0\left(\frac{x_0}{2^n} + \frac{x_1}{2^{n-1}} + \cdots + \frac{x_{n-1}}{2}\right).$$

then :

$$QFT|x\rangle =$$
$$= \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i\left(y_{n-1}\frac{x_0}{2} + y_{n-2}\left(\frac{x_0}{4}+\frac{x_1}{2}\right) + \cdots + y_0\left(\frac{x_0}{2^n} + \frac{x_1}{2^{n-1}} + \cdots + \frac{x_{n-1}}{2}\right)\right)} |y\rangle =$$
$$= \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i y_{n-1}\frac{x_0}{2}} e^{2\pi i y_{n-2}\left(\frac{x_0}{4}+\frac{x_1}{2}\right)} \dots e^{2\pi i y_0\left(\frac{x_0}{2^n}+\frac{x_1}{2^{n-1}}+\cdots+\frac{x_{n-1}}{2}\right)} |y\rangle.$$

$$QFT|x\rangle =$$
$$= \frac{1}{\sqrt{2}} e^{2\pi i \cdot 0 \cdot \frac{x_0}{2}} |0\rangle \otimes \frac{1}{2^{\frac{n-1}{2}}} \sum_{y=0}^{2^{n-1}-1} e^{2\pi i y_{n-2}(\cdots)} \dots e^{2\pi i y_0(\cdots)} |y_{n-2} y_{n-1} \cdots y_0\rangle +$$
$$+ \frac{1}{\sqrt{2}} e^{2\pi i \cdot 1 \cdot \frac{x_0}{2}} |1\rangle \otimes \frac{1}{2^{\frac{n-1}{2}}} \sum_{y=0}^{2^{n-1}-1} e^{2\pi i y_{n-2}(\cdots)} \dots e^{2\pi i y_0(\cdots)} |y_{n-2} y_{n-1} \cdots y_0\rangle =$$
$$= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \frac{x_0}{2}} |1\rangle\right) \otimes \frac{1}{2^{\frac{n-1}{2}}} \sum_{y=0}^{2^{n-1}-1} e^{2\pi i y_{n-2}(\cdots)} \dots e^{2\pi i y_0(\cdots)} |y_{n-2} y_{n-1} \cdots y_0\rangle.$$

$$QFT|x\rangle =$$
$$= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \frac{x_0}{2}} |1\rangle\right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \left(\frac{x_0}{4}+\frac{x_1}{2}\right)} |1\rangle\right) \otimes \cdots$$
$$\cdots \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \left(\frac{x_0}{2}+\cdots+\frac{x_{n-1}}{2}\right)} |1\rangle\right).$$

It appears that $QFT|x\rangle$ can be decomposed on a tensor product of n 1-qubit operators.

Let's define a set of new 1-qubit gates R_k :

$$R_k = \left(\begin{array}{cc} 1 & 0 \\ 0 & e^{2\pi i \frac{1}{2^k}} \end{array} \right).$$

Figure 2.21 shows the circuit scheme of $QFT|x\rangle$ with 3 qubits, with the use of R_k H gates. This scheme changes the order of qubits - the most significant qubit is mapped to the least significant one.

The upper line in the scheme corresponds to the first 1-qubit operator in

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \frac{x_0}{2}} |1\rangle\right) = \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{x_0} |1\rangle\right) = H |x_0\rangle,$$

The second line of the scheme corresponds to the second 1-qubit operator in

$$H\,|x_1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{x_1}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\frac{x_1}{2}}|1\rangle\right) \xrightarrow{R_2(x_0)}$$

$$\xrightarrow{R_2(x_0)} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{x_0}{4}+\frac{x_1}{2}\right)}|1\rangle\right),$$

And finally the third line and the third operator:

$$|x_2\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\frac{x_2}{2}}|1\rangle\right) \xrightarrow{R_3(x_0)}$$

$$\xrightarrow{R_3(x_0)} \frac{1}{\sqrt{2}}\left(\,|0\rangle + e^{2\pi i(\frac{x_2}{2}+\frac{x_0}{8})}\,|1\rangle\,\right) \xrightarrow{R_2(x_1)}$$

$$\xrightarrow{R_2(x_1)} \frac{1}{\sqrt{2}}\left(\,|0\rangle + e^{2\pi i(\frac{x_2}{2}+\frac{x_1}{4}+\frac{x_0}{8})}\,|1\rangle\,\right).$$

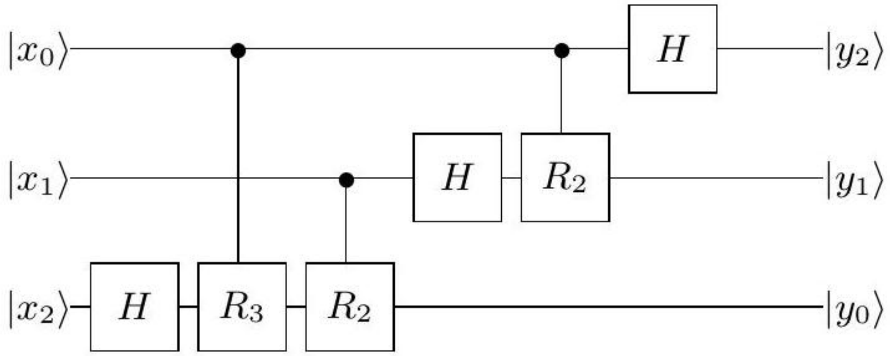


Figure 3.21: QFT, 3 qubits

If we generalize fig. 1 on n qubits, the complexity of the QFT implementation becomes

$$\sum_{j=1}^{n} j \approx O\left(n^2\right)$$

of R_k and H gates.

The Shor's Algorithm

The circuit scheme of the Shor's algorithm - figure 2.22

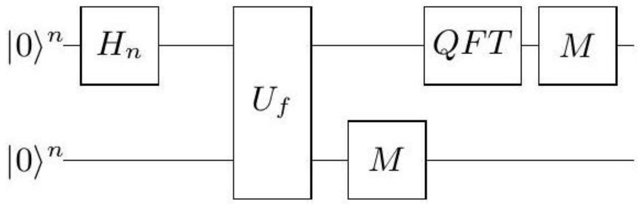


Figure 3.22: Shor's Algorithm

The action of the algorithm:

$$|0\rangle^n|0\rangle^n \xrightarrow{H_n} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle|0\rangle^n \xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle \xrightarrow{\text{Measure } Y}$$

$$\xrightarrow{\text{Measure } Y} \frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |x_0 + jr\rangle \, |f\left(x_0\right)\rangle \, .$$

A - the number of all values which are mapped by f into $f\left(x_0\right)$:

$$A \approx \frac{N}{r},$$
$$N - r \leq Ar \leq N + r.$$

The QFT action:

$$\frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |x_0 + jr\rangle \xrightarrow{QFT} \frac{1}{\sqrt{AN}} \sum_{j=0}^{A-1} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x_0 y}{N}} e^{2\pi i \frac{jry}{N}} |y\rangle =$$
$$= \frac{1}{\sqrt{AN}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x_0 y}{N}} \sum_{j=0}^{A-1} e^{2\pi i \frac{jry}{N}} |y\rangle.$$

The probability of any particular y to be the measurement outcome:

$$P(y) = \frac{1}{AN} \left| e^{2\pi i \frac{x_0 y}{N}} \right|^2 \left| \sum_{j=0}^{A-1} e^{2\pi i \frac{jry}{N}} \right|^2 =$$
$$= \frac{1}{AN} \left| \sum_{j=0}^{A-1} e^{2\pi i \frac{jry}{N}} \right|^2 .$$

Let's denote

$$\theta_y = 2\pi \frac{ry(N)}{N}.$$

The sum after QFT action is the sum of the geometrical progression:

$$\sum_{j=0}^{A-1} e^{2\pi i \frac{jry}{N}} = \sum_{j=0}^{A-1} e^{\theta_y j} = \frac{e^{A\theta_y i} - 1}{e^{\theta_y i} - 1}.$$

We are going to estimate the probability of the measurement of the special class of values y, which we call "good", and which are defined by .

$$A\left|\theta_y\right| \in [0, \pi].$$

For "good" y-s we have:

$$\left|e^{A\theta_y i} - 1\right| \geq \frac{2A\left|\theta_y\right|}{\pi}.$$

The proof of previous inequality:

$$A\left|\theta_y\right| =: x$$

$$\left|e^{xi} - 1\right| = \left|\cos x - 1 + i\sin x\right| = \sqrt{(\cos x - 1)^2 + \sin^2 x} =$$

$$= \sqrt{2 - 2\cos x} = \sqrt{2\left(\cos^2\frac{x}{2} + \sin^2\frac{x}{2}\right) - 2\cos^2\frac{x}{2} + 2\sin^2\frac{x}{2}} = 2\sin\frac{x}{2},$$

since sine is positive on the interval $[0, \pi/2]$.

So, we need to prove this:

$$\sin\frac{A\left|\theta_y\right|}{2} \geq \frac{A\left|\theta_y\right|}{\pi}.$$

On the boundaries of $[0, \pi]$ we have an equality:

$$\sin 0 = 0,$$
$$\sin\frac{\pi}{2} = \frac{\pi}{\pi} = 1.$$

Sine is a convex function on $[0, \pi]$, the right part of the inequality is a linear function, so the inequality is true for the whole interval $[0, \pi]$.

Also we can note that

$$\left|e^{\theta_y i} - 1\right| \leq \left|\theta_y\right|.$$

since chord is shorter than its arc (figure 2.23):

From the inequalities above we can estimate the probability of the "good" y measurement:

$$P(y) \geq \frac{1}{AN}\left(\frac{2A\left|\theta_y\right|}{\pi}\right)^2\left(\frac{1}{\left|\theta_y\right|}\right)^2 = \frac{4A}{\pi^2 N} \approx \frac{4}{\pi^2 r}.$$

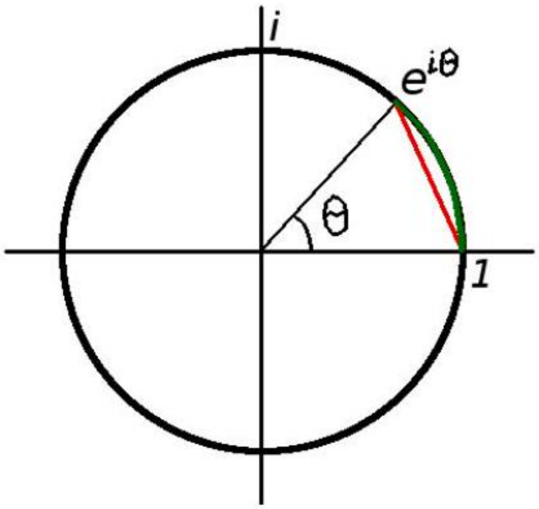


Figure 3.23: Values from (18) on the complex plane

The "good" y-s are:

$$\begin{gathered}A\left|\theta_y\right| \in [0,\pi] \iff \\ \iff -\pi \le 2\pi\frac{A\cdot yr(N)}{N} \le \pi \iff \\ \iff -\frac{1}{2} \le \frac{yr(N)}{r} \le \frac{1}{2} \iff \\ \iff -\frac{r}{2} \le yr(N) \le \frac{r}{2} \iff \\ \iff Nk - \frac{r}{2} \le yr \le Nk + \frac{r}{2}.\end{gathered}$$

The inequality has only 1 solution y_k for any $k \in \mathbb{N} \cup \{0\}$

$$\begin{gathered}y_0 = 0, \\ y_r = N = 0(N) = y_0(N).\end{gathered}$$

That means that there are only r of different "good" y-s.

$$P\left(y = {}'\mathrm{good}'\right) = \frac{4}{\pi^2 r}r = \frac{4}{\pi^2} \approx 0,406\cdots$$

Why "good" y is good?

It appears that QFT, which is applied at the end of the Shor's algorithm, dramatically increases the amplitudes of the certain vectors - those we have called "good". It's time to find out why they are "good". First we will divide by Nr

$$-\frac{r}{2} \leq yr(N) \leq \frac{r}{2} \Longleftrightarrow$$
$$\Longleftrightarrow \frac{k}{r} - \frac{1}{2N} \leq \frac{y}{N} \leq \frac{k}{r} + \frac{1}{2N} \Longleftrightarrow$$
$$\Longleftrightarrow \left|\frac{y}{N} - \frac{k}{r}\right| \leq \frac{1}{2N}.$$

We can see that in a close (of size $1/N$) neighborhood of $\frac{y}{N}$ resides the number $\frac{k}{r}$, which is extremely interesting to us because of its denominator.

If we assume that

$$r < \sqrt{N},$$

then in this neighborhood $\frac{k}{r}$ is the only number with the denominator less than $\sqrt{N}$:

$$a, b, r_1, r_2 \in \mathbb{N}, a \neq b,$$
$$r_1 < \sqrt{N}, r_2 < \sqrt{N}$$
$$\left|\frac{a}{r_1} - \frac{b}{r_2}\right| = \frac{|a \cdot r_2 - b \cdot r_1|}{r_1 \cdot r_2} \geq \frac{1}{r_1 \cdot r_2} \geq \frac{1}{\sqrt{N}\sqrt{N}} = \frac{1}{N}.$$

So, after the measurement of y, if this y is "good" we can search for a fraction with denominator less than $\sqrt{N}$ in the $1/N$-neighborhood of the fraction $\frac{y}{N}$. For example by the continued fraction method:

$$n = 10, 2^n = N = 1024, \sqrt{N} = 32.$$

Measured y is 139 :

$$\frac{y}{N} = \frac{139}{1024}$$
$$\frac{139}{1024} = \frac{1}{\frac{1024}{139}} = \frac{1}{7 + \frac{51}{139}} \approx \frac{1}{7}$$
$$\frac{1}{7 + \frac{1}{\frac{139}{51}}} = \frac{1}{7 + \frac{1}{\frac{139}{51}}} = \frac{1}{7 + \frac{1}{2 + \frac{37}{51}}} \approx \frac{2}{15}.$$
$$\frac{1}{7 + \frac{1}{2 + \frac{37}{51}}} = \frac{1}{7 + \frac{1}{2 + \frac{1}{51}}} = \frac{1}{7 + \frac{1}{2 + \frac{1}{1 + \frac{14}{37}}}} \approx \frac{3}{22}.$$

3.11.2 Implementation Example

$$p = 3, q = 5, N = pq = 15,$$
$$\Phi(N) = (5-1)(3-1) = 8$$
$$a = 7, f(x) = 7^x(15).$$
$$f(x) = 7^x(15) = \left(7^8\right)^{x_3} \cdot \left(7^4\right)^{x_2} \cdot \left(7^2\right)^{x_1} \cdot \left(7^1\right)^{x_0}(15) = \left(7^2\right)^{x_1} \cdot \left(7^1\right)^{x_0}(15).$$

since 4 is the order of 7 in $\mathbb{Z}_{15}$, and $7^4 = 1(15)$.

$$\left(7^2\right)^{x_1} \cdot \left(7^1\right)^{x_0} (15) = (4)^{x_1} \cdot (7)^{x_0}(15).$$

The algorithm implementation on the quantum computer simulator - figure 2.24

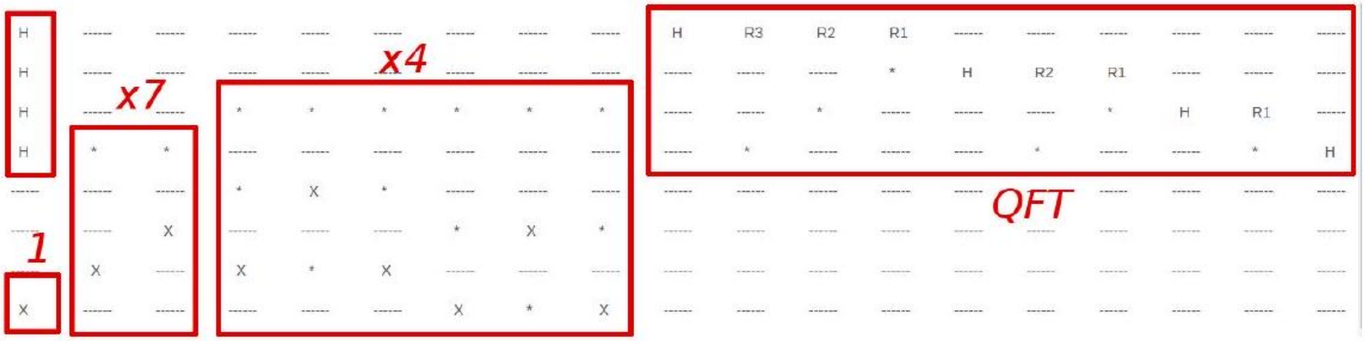


Figure 3.24: Shor's Algorithm. $f(x) = 7^x(15)$

We read value 0 x 0100 = 4 on the output.

$$\frac{y}{2^n} = \frac{4}{2^4} = \frac{1}{4}.$$

1/4 has small enough denominator so we suppose it is $\frac{k}{r}$ and $4-$ is r :

$$f(4) = 7^4(15) = 1.$$

OK, r is indeed the period.

Now let's search for p and q :

$$GCD\left(7^{\frac{4}{2}} + 1, 15\right) = GCD(50, 15) = 5,$$
$$GCD\left(7^{\frac{4}{2}} - 1, 15\right) = \mathrm{GCD}(48, 15) = 3.$$

p and q are found. Shor's algorithm works!

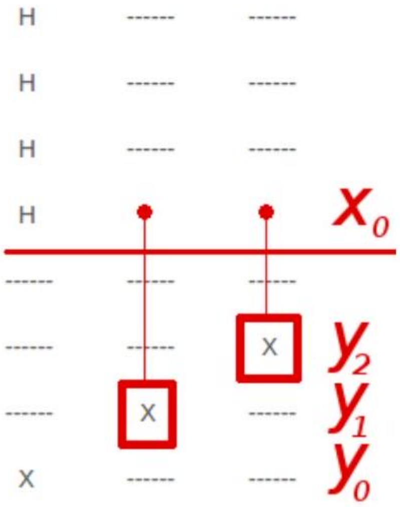


Figure 3.25: Operation ×7

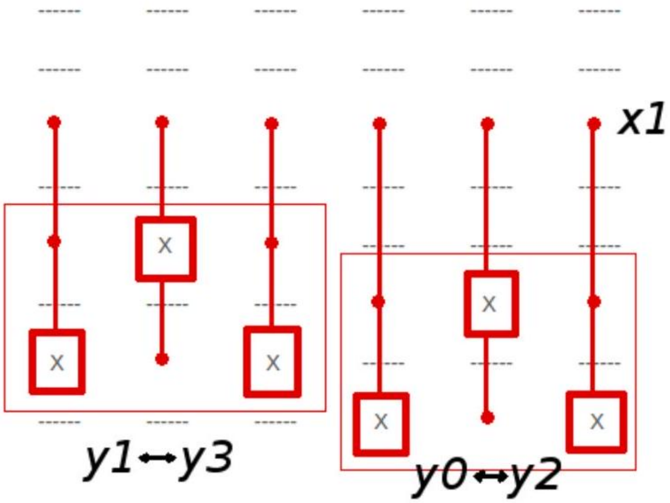


Figure 3.26: Operation ×4

3.12 Quantum Hardware

3.12.1 Complexity

1. Isolation

   The most common challenge is isolation. Heat and light can cause quantum decoherence, where qubits lose their quantum properties (superposition and entanglement), as well as the information they store. Typically, quantum computers are stored at almost 0 Kelvin (273.15 Celsius).[5]

2. Signal control

   To change the state of a qubit, it needs to be rotated (flipped by the logic gate).

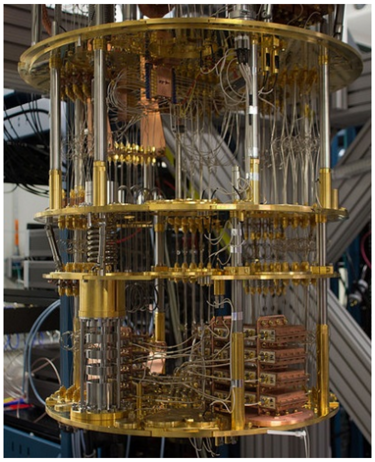


Figure 3.27: IBM Q prototype

These rotations are prone to error. For instance, if an algorithm requires a number of qubits to be rotated by 90 degrees but an error causes a rotation by 90.1 degrees, the accumulation of qubits rotating at this rate of error will result in an incorrect output.[5]

3. Quantum error correction QEC approaches

   Quantum error correction (QEC) is necessary to protect quantum information from errors due to decoherence and other quantum noise. In classical computers, error correction employs redundancy which means copying and storing the bits used to encode a given amount of information many times and checking whether they are the same. If changes are found then the majority of identical information is the true version.[5]

   However, quantum information cannot be copied due to the no-cloning theory which states that it is impossible to create an identical copy of an arbitrary quantum state. Another challenge that faces quantum error correction is the problem of wavefunction collapse. In a classical computing, arbitrary properties can be measured without compromising the encoded information. In quantum computing, measuring qubits as a part of the error correction procedure must be carefully done in order not to cause the wavefunction to collapse and erase the encoded information.[5]

3.12.2 The Quantum Computer, DIY

To assemble the simple quantum computer for the Deutsch's algorithm demonstration we need (figure 2.27):

1. The laser. In this example we use 650 nm (red) laser pointer, 100 mW
2. The linear polarizer 3. 2 beam-splitters, red light (since the laser is red), non-polarizing
3. 2 metal mirrors

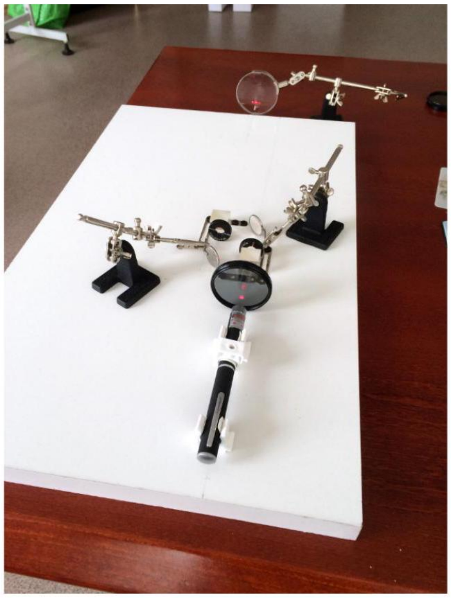


Figure 3.28: 2-qubit Quantum Computer (not a universal one, however)

4. 2 half-waveplates for 650 nm (since the laser gives us this wavelength)

5. A heavy flat board (chipboard works fine for us)

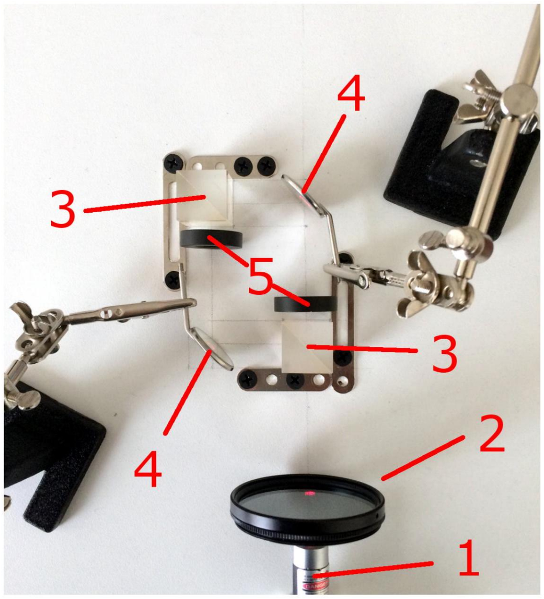


Figure 3.29: Quantum Computer (a closer look)

3.12.3 Half-waveplates

Waveplate delays the propagation of photons, polarized orthogonal to its optical axis. Half-waveplate delays the photon on exactly half of its period (thus it depends on the wavelength of light we are going to process) - this half-period delay is identical to multiplication of the wave by -1.

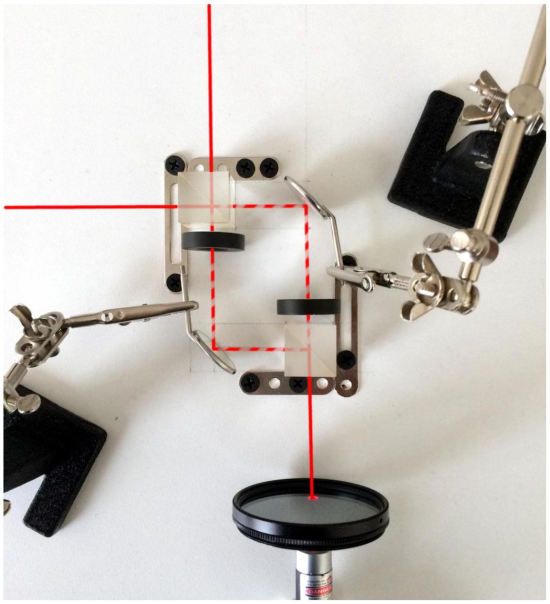


Figure 3.30: Mach-Zehnder Interferometer Action

Action of a half-waveplate on a photon of corresponding wavelength is reflection of its polarization over the optical axis of the waveplate.

Both qubits in our quantum computer are carried by 1 photon, the first qubit is the photon's polarization, the second qubit - the photon's path in the interferometer:

$|0\rangle$ - left path,

$|1$ - right path.

$|0\rangle$ - horizontal polarization,

$|1\rangle$ - vertical polarization.

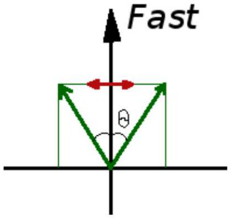


Figure 3.31: Waveplate $\lambda/2$

3.12.4 Computer Action. The Polarizer

The linear polarizer plays the role of the Hadamard transform on the second qubit, since the photons which pass it are polarized along its optical axis which we choose to be this - (fig 2.31).

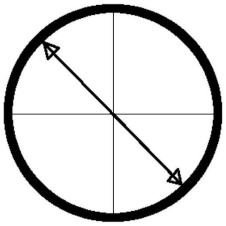


Figure 3.32: Polarizer Orientation

The state after the polarizer:

$$\frac{1}{\sqrt{2}}|0\rangle(|0\rangle - |1\rangle).$$

3.12.5 Computer Action. The 1st Beam-splitter

The first beam-splitter works as the Hadamard transform on the first qubit:

$$\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle).$$

3.12.6 Computer Action. Oracle

The oracle is implemented via the half-waveplates with optical axes oriented perpendicularly to the photons polarization (figure 2.32).

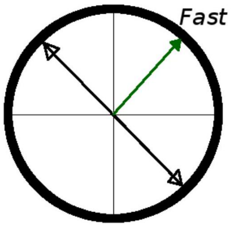


Figure 3.33: Waveplates Orientation

This orientation of the half-waveplates implements the multiplication of the state by (-1) :

$$|x\rangle(|0\rangle-|1\rangle) \to -|x\rangle(|0\rangle-|1\rangle).$$

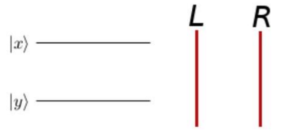


Figure 3.34: Circuit Scheme and its Implementation $(f(x)=0)$

3.13 Computer Action. The 2nd Beam-splitter

The second beam-splitter plays the role of the Hadamard transform on the first qubit.

Computer Action. The Measurement

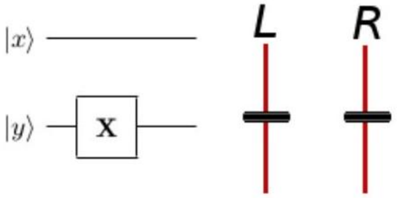


Figure 3.35: Circuit Scheme and its Implementation $(f(x)=1)$

So, after the whole set is adjusted and tested we need only 1 photon to find out which function type is implemented inside of the interferometer.

3.13.1 Solving Deutsch problem

The Problem Formulation

An oracle (black box) function $f : \{0,1\}^n \to \{0,1\}$ maps n bit to 1 bit. We know from a trusted source (which we are not going to reveal here) that f is either a constant function or

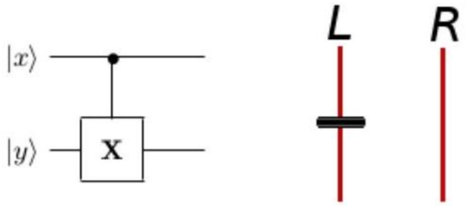


Figure 3.36: Circuit Scheme and its Implementation ($f(x) = x$)

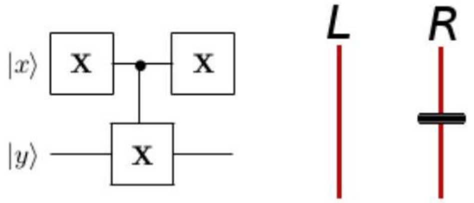


Figure 3.37: Circuit Scheme and its Implementation ($f(x) = \bar{x}$)

a balanced one (a balanced function returns 1 on exactly one half of its inputs). We need to know which case is it - function of what type is implemented in the black box.

To prove that f is a constant in classical case we would need $2^{n-1} + 1$ oracle queries. The quantum algorithm (figure 2.39) needs only one.

The algorithm action:

$$|0\rangle^n|1\rangle \xrightarrow{H_{n+1}} \frac{1}{2^{\frac{n+1}{2}}} \sum_{x=0}^{2^n-1} |x\rangle(|0\rangle - |1\rangle) \xrightarrow{U_f}$$

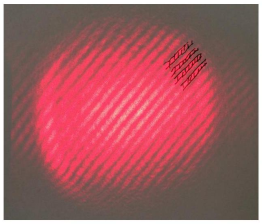


Figure 3.38: Interference. f is a constant ($f(x) = \bar{x}$)

$$\xrightarrow{U_f} \frac{1}{2^{\frac{n+1}{2}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle).$$

For $f =$ const we can take $(-1)^{f(x)}$ out of the sum and for the first n qubits we have:

$$\frac{1}{2^{n/2}}(-1)^f \sum_{x=0}^{2^n-1} |x\rangle = (-1)^f H_n|0\rangle^n$$

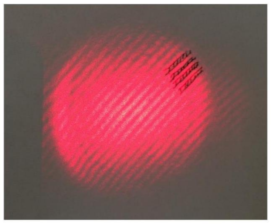


Figure 3.39: Interference. f is balanced

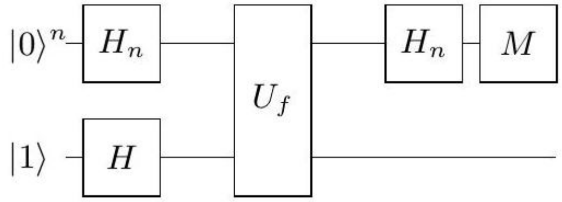


Figure 3.40: The Deutsch-Jozsa Algorithm

For a balanced function we have

$$\frac{1}{2^{n/2}}\left(\sum_{x:f(x)=0}|x\rangle - \sum_{x:f(x)=1}|x\rangle\right)$$

$$H_n|x\rangle = \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}(-1)^{x\bullet y}|y\rangle = \frac{1}{2^{n/2}}\left(|0\rangle^n + \sum_{y=1}^{2^n-1}(-1)^{x\cdot y}|y\rangle\right).$$

Since the number of terms in both sums is the same, all vectors $|0\rangle^n$ will selfdestruct giving us a zero probability to obtain such vector as the measurement result.

So, if in the end we measure $|0\rangle^n$ then f is a constant. Anything else will tell us that f is balanced.

3.14 Conclusion

In this chapter we introduced quantum computing and some of the famous algorithms .Explaining theoretically each algorithm using Hilbert space and differential geometry with examples

In addition we presented a small quantum computer prototype and it's application to solve Deutsch 's problem

Chapter 4

Maximum power point tracking

4.1 Introduction

The energy area is one of the crucial fields in the world right now for electricity needs especially a majority depending on the non renewable energy which is unsustainable.This made researchers to build up strategies to increase the efficiency of electric systems by turning also to viable energies (Solar ,Wind,Hydraulics)
in another words to replace fossil fuel resources a solar energy technology has been adopted nevertheless the big challenge is efficiency

At present, most of the street lights are powered through conventional energy sources. But, there is a huge disparity between the growing demand and available resources. Hence demand for renewable energy that is, Solar power based street lights are increased.

The Solar powered devices are best suitable in places where there is scarcity of renewable resources. The initial cost of these solar powered devices is more. Therefore, the integration of a robust MPPT controller is necessary. Maximum power point tracking (MPPT) consists of extracting the maximum power from the PV generator. With MPPT more devices can be powered with less wattage panel.

4.2 Photo-voltaic Systems

4.2.1 Process of PV cells

PV cells are made of specific materials like silicium ,germanuim called semiconductors .The energy of photons carried by solar arrays invokes electrons of Silicium and highs up their level energy .If this energy is higher than the Si 's gap energy then a displacement will happen from the band of valance to the conduction making pairs of holes and excited electrons. Applying an external voltage will force the electrons to create a unidirectionnal flow of continued current.

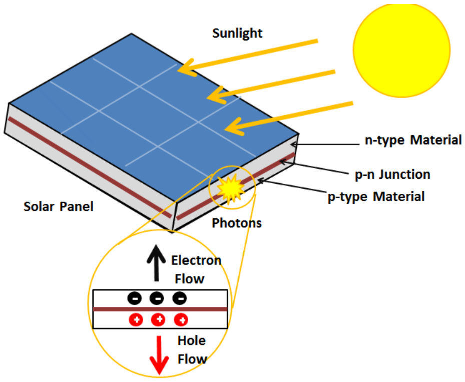


Figure 4.1: photovoltaic effect

In other words when illuminating a PN junction (device in which the doping of the semiconductor suddenly changes from a P-type to an N-type), the electron-hole pairs that are created in the space charge region of the junction are immediately separated by the electric field which prevails in this region, and driven into the neutral zones on either side of the junction. If the device is isolated, it a potential difference appears across the terminals of the junction (photo voltage); if he is connected to an external electrical load, we observe the passage of a current then that no voltage is applied to the device. This is the basic principle of a cell photovoltaic.

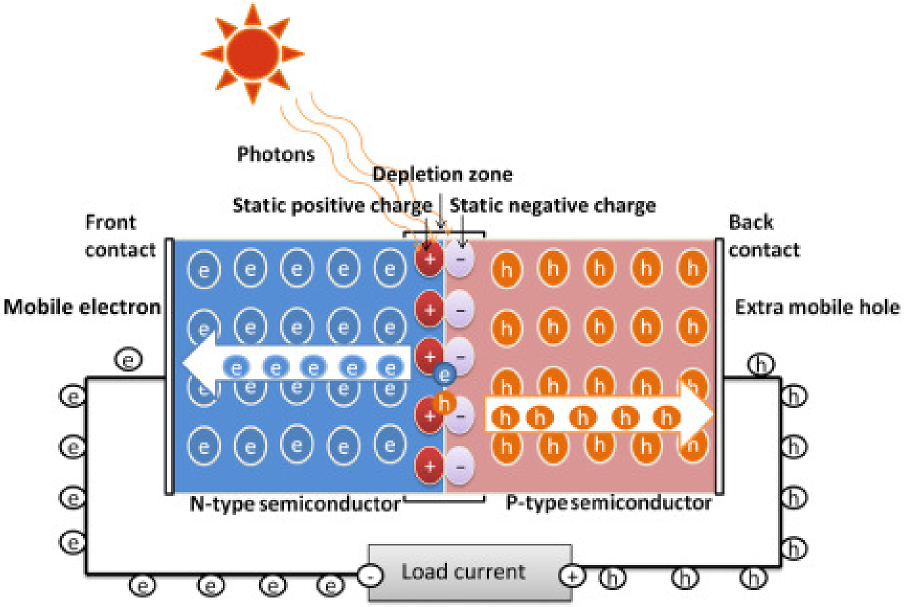


Figure 4.2: PV cell P-N junction

4.2.2 characteristics of a cell

The basic characteristics of a solar cell are the short-circuit current (ISC), the open-circuit voltage (VOC), the fill factor (FF) and the solar energy conversion efficiency (). The influence of both the diode saturation current density and of ISC on VOC, FF and is analyzed for ideal solar cells.

Short circuit current

The short-circuit current is the current through the solar cell when the voltage across the solar cell is zero (i.e., when the solar cell is short circuited). Usually written as ISC, the short-circuit current is shown on the IV curve below.

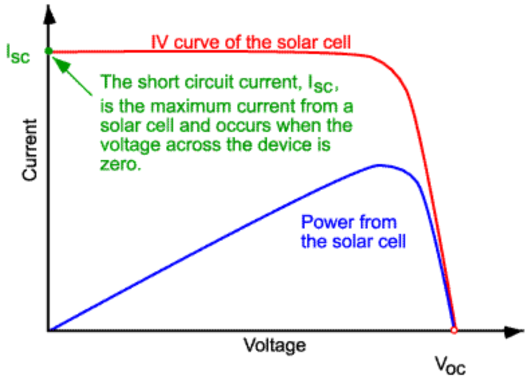


Figure 4.3: Current Curve showing short circuit current

The short-circuit current ISC is due to the generation and collection of light-generated carriers. For an ideal PV cell with moderate resistive loss, ISC and the light-generated current are identical (ISC is the largest current which may be drawn from the solar cell)

open circuit voltage

The open-circuit voltage, VOC, is the maximum voltage available from a solar cell, and this occurs at zero current. The open-circuit voltage corresponds to the amount of forward bias on the solar cell due to the bias of the solar cell junction with the light-generated current. The open-circuit voltage is shown on the IV curve below.[6]

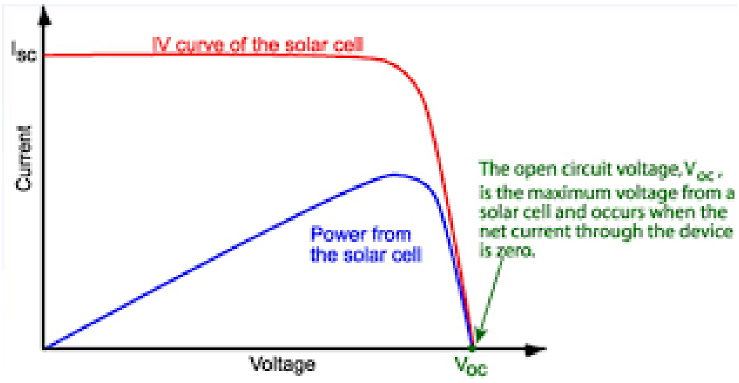


Figure 4.4: Current Curve showing Open circuit voltage

V_oc is given when the current is zero which can be expressed in the PV cell equation :

$$V_{OC} = \frac{nkT}{q} \ln\left(\frac{I_L}{I_0} + 1\right) \tag{4.1}$$

fill factor

The Fill Factor (FF) is typically a measure of the Efficiency of a solar PV module.[7]

FF is the ratio of maximum power (Pmax) to the product of VOC IS i.e:

$$FF = \frac{P_{MP}}{V_{OC} \times I_{SC}} \tag{4.2}$$

as it shown in the down below graph

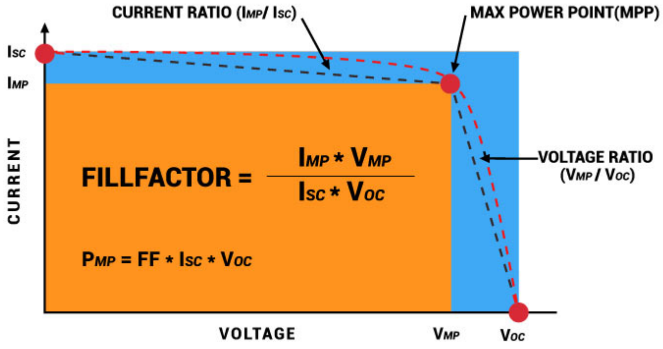


Figure 4.5: Fill Factor of Sollar cell

conversion efficiency

The Efficiency of a solar cell is a determination of a solar panel's power-producing capacity.[8]

It is the ratio of the highest power to the input power.

$$\eta = \frac{V_{OC} I_{SC} FF}{P_{in}} \tag{4.3}$$

Voc is the open-circuit voltage;

Isc is the short-circuit current;

FF is the fill factor and

is the efficiency.

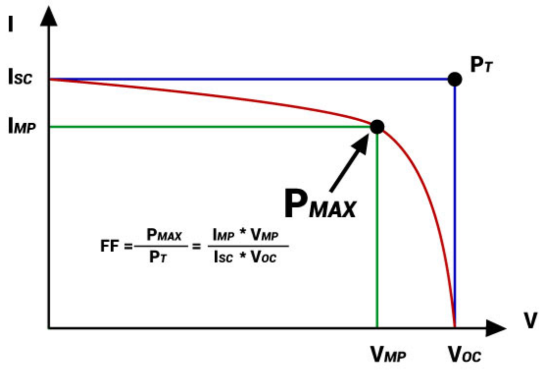


Figure 4.6: current curve

4.2.3 Modelisation of PV cells

The mathematical models for PV arrays are based on the theoretical equations that describe the functioning of the PV cells and can be developed using the equivalent circuit of the PV cells. The empirical models rely on different values extracted from the I-V curve of the PV arrays and they approximate the characteristic equation of the solar panels using an analytical function

A simple equivalent circuit model for a PV cell consists of a real diode in parallel with an ideal current source. The ideal current source delivers current in proportion to the solar flux to which it is exposed. There are two conditions of interest for the actual PV and for its equivalent circuit, which are:

1. the current that flows when the terminals are shorted together (the short-circuit current, Isc);

2. the voltage across the terminals when the leads are left open (the open-circuit voltage, Voc).

When the leads of the equivalent circuit for the PV cell are shorted together, no current flows in the (real) diode since Vd = 0, so the whole current from the ideal source flows through the shorted leads. Since the short-circuit current must equal Isc, the magnitude of the ideal current source itself must be equal to Isc. When the leads from the PV cell are left open, the load current, I, is null and the V on the load is equal to Voc = Vd..[9]

4.2.4 PV generator

Until the right moment There are multiple configurations of photovoltaic fields in the literature, the best known of which are:

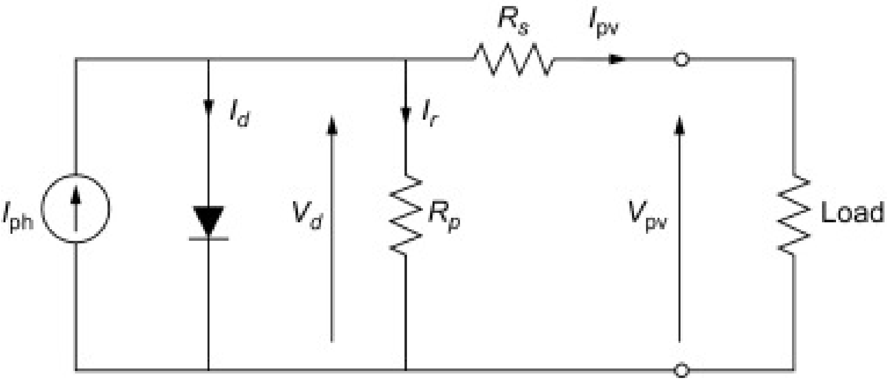


Figure 4.7: PV circuit

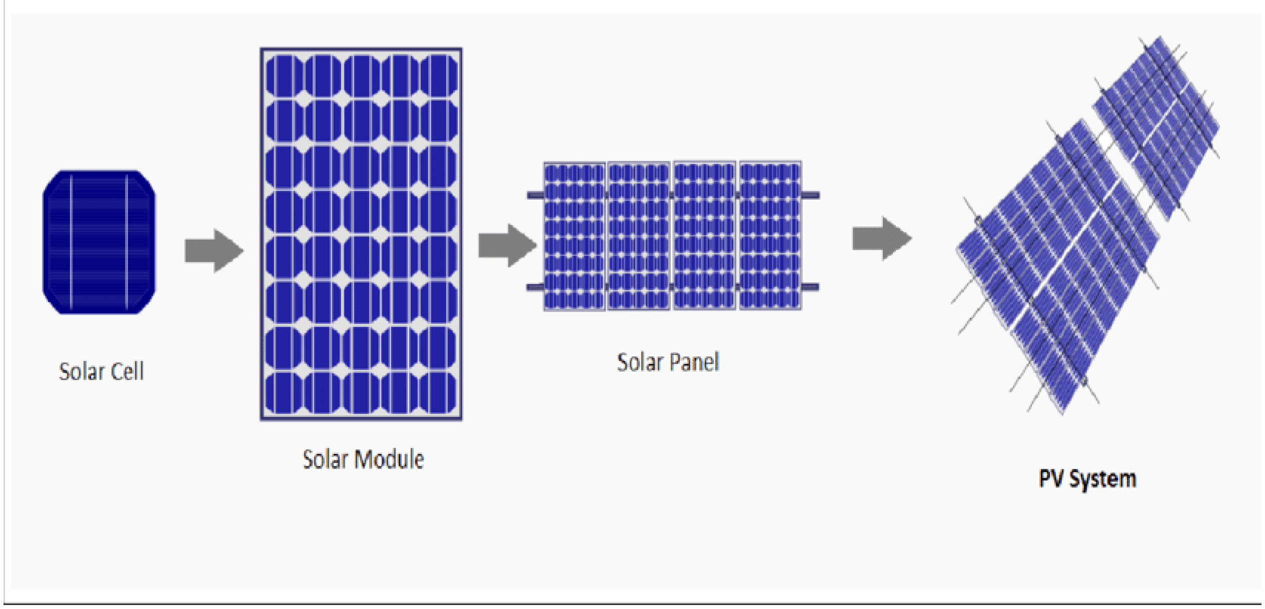


Figure 4.8: PV system

1. **Series-parallel:**

   This is the most validated configuration. The panels are connected in series (chain), then these chains are connected to each other in parallel . The power produced at the output is increased but the disadvantage of this configuration is that it saves a lot of losses of powers due to panels mounted in series. It is recommended in such configuration to limit the number of photovoltaic panels per string and to favor a greater number of parallel connections. .[10]

2. **Total cross Tied:**

   It is achieved by connecting all the panels in parallel photovoltaic on the same line of the different strings . That gives a solar field in the form of a matrix containing several nodes. The sum of the currents in the different nodes and the voltage of the panels photovoltaics mounted in parallel are equal. This configuration in the majority

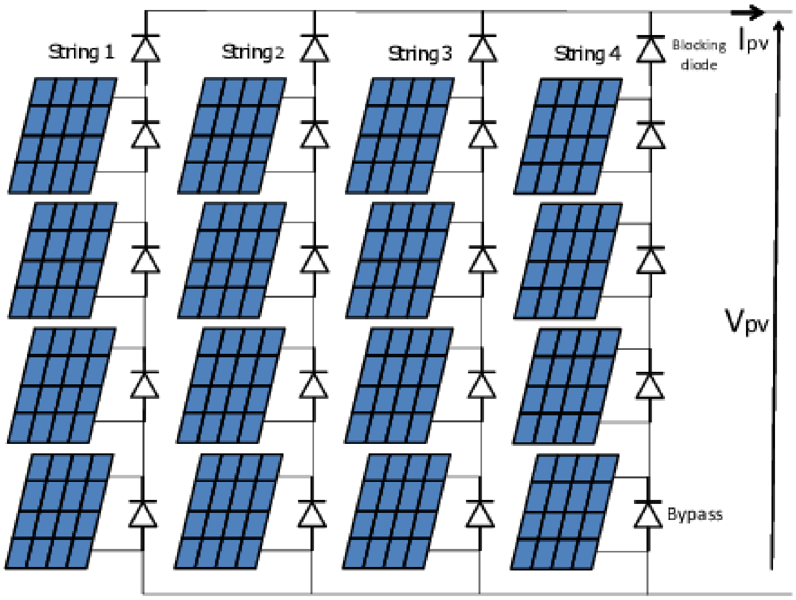


Figure 4.9: Series parallel combination for the PV generator

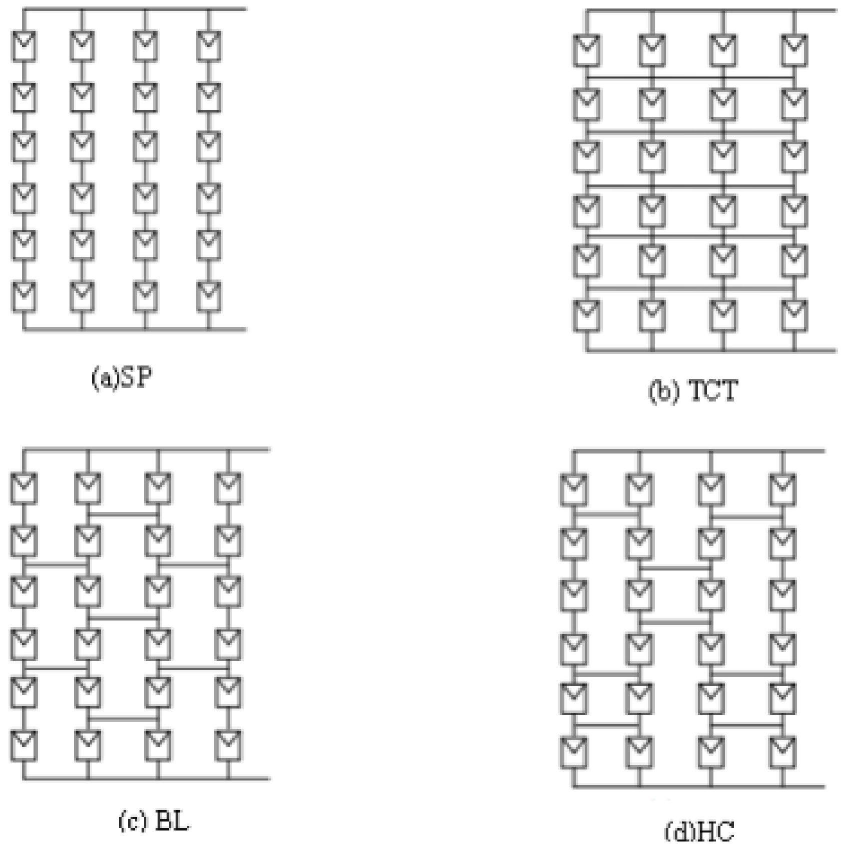
of shading models tested give better results compared to the other three configurations. This is due to multiple connections which make it possible not to activate the bypass diodes in all the shade conditions. This allows voltage and current to be increased while minimizing power losses. The downside of this setup is that it is more expensive to implement. .[10]

3. **Bridge Link:**

   This configuration is made up of several islands . Each island is made up of two parallel strings each carrying two panels solar panels in series with links interposed between the bridges. It also allows to increase the voltage and the current while limiting the power losses in the photovoltaic system. Compared to the total cross tied configuration, the bridge link configuration is less efficient for cases of partial shade but presents a better performance than the total cross tied when it is subject to full sunlight. This is because it requires less spinning, therefore less power loss. Compared to configuration series-parallel, it is much more expensive to implement but has less loss than the series-parallel configuration. .[10]

4. **Honey Comb:** It is made up of two parallel strings made up of three modules connected in series . Faced with total configurations cross tied and bridge link, the Honey Comb configuration is recognized as having average performance because it registers a little more power loss.

   On the other hand, it has a better performance for a solar field. laid out and connected asymmetrically or when the number of columns receiving the same sunshine is greater than the number of rows..[10]



(a)SP

(b) TCT

(c) BL

(d)HC

Figure 4.10: Different types of Of PV generators

4.2.5 Voltage-current

The IV curve of a solar cell is the superposition of the IV curve of the solar cell diode in the dark with the light-generated current. The light has the effect of shifting the IV curve down into the fourth quadrant where power can be extracted from the diode. Illuminating a cell adds to the normal "dark" currents in the diode so that the diode law becomes:

$$I = I_0 \left[\exp\left(\frac{qV}{nkT} \right) - 1 \right] - I_L \tag{4.4}$$

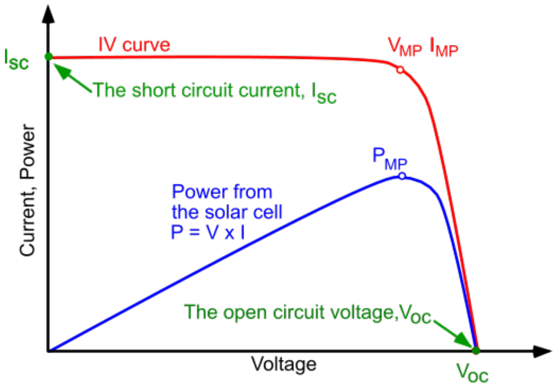


Figure 4.11: current curve

Current voltage (IV) cure of a solar cell. To get the maximum power output of a solar cell it needs to operate at the maximum power point, PMP.

4.2.6 converters

A typical output voltage of PV panels can be on the order of 30 V, and it is too low for being converted to AC and fed to the grid. Therefore DC/DC conversion is often a necessary step before the DC current from the PV system is supplied to the inverter. Most of power conditioning units include some type of DC/DC converter.

an ideal case, when input power is equal to the output power. In reality, there are always conversion losses, which lead to typical efficiencies in the range 90-95

DC/DC conversion allows keeping the voltage on the PV and voltage on the load separately controlled. There two main types of DC/DC converters depending on the direction of voltage change: boost converters transform smaller voltage to higher voltage and buck converters transform higher voltage to lower voltage

4.2.7 Buck Boost Converter

The topology of the Buck-Boost converter has the following electrical components:

- a DC input voltage
- a load resistor R,
- a filter capacitor C,
- an inductor L
- and two complementary switches

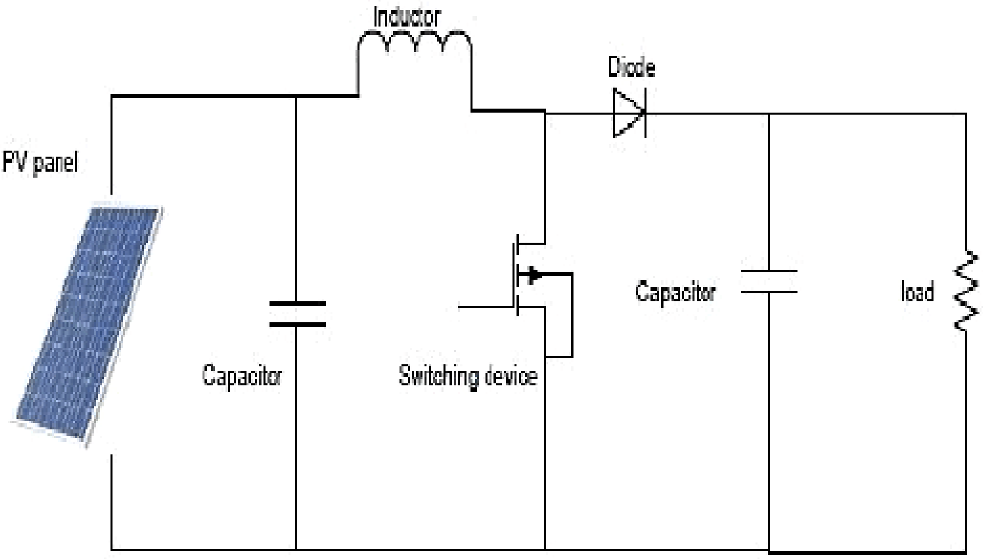


Figure 4.12: PV panel connected with dc-dc boost converter

Figure 4.13: Buck Boost Converter Component

the buck-boost is a non-isolated power converter. However, the buck-boost is capable to provide a voltage both higher and lower than the input source. To analyze the operation of the buck-boost converter, we will apply the same reasoning used in the Voltage Booster article. The circuit is shown in the Figure 1, and it is interesting to notice that the number of components is the same as the buck and the boost, but the functionality is slightly different: the polarity of the output voltage is opposite to the input

In case conduction mode we obtain a duty cycle which is designed to obtain a certain voltage

$$D = \frac{V_o}{V_o - V_i} \tag{4.5}$$

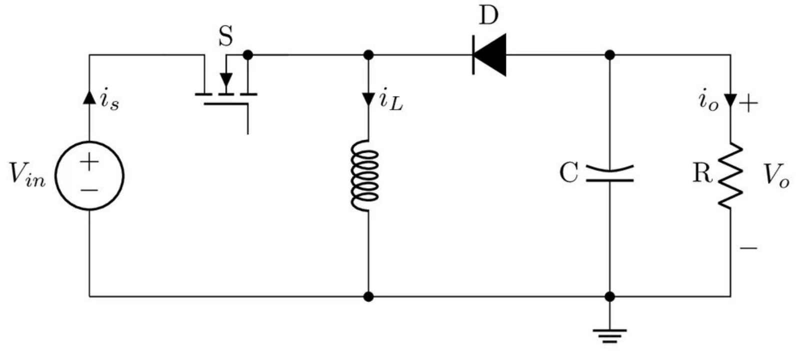


Figure 4.14: Buck Boost Converter

4.3 MPPT Algorithms

4.3.1 General process

It is a method to determine the point at which the maximum power generated by the solar panels. Based on research conducted by Priananda and Sulistyowati [11], one of the advantages of the use of the MPPT is quick to satisfy the condition of equilibrium photovoltaics for the conditions required by the load and which can be filled with solar panels. MPPT requires two components supporting the input current to operate

4.3.2 Conventional Methods(Closed Loop

Pertub and Observe

The PO algorithm is arguably the easiest to implement in systems PV. Moreover, it is the most used method.

The principle of this algorithm is to perform a disturbance on the voltage of the PV panel while acting on the duty cycle D. Indeed, following this disturbance, we calculate the power supplied by the PV panel at time (k), then we compare it to the previous instant (k − 1). If the power increases, we approach the point of maximum power, 'MPP' and the variation of the duty cycle is maintained in the Same direction. On the contrary, if the power decreases, we move away from the power point maximum, 'MPP'. Then, we must reverse the direction of the variation of the duty cycle.[12]

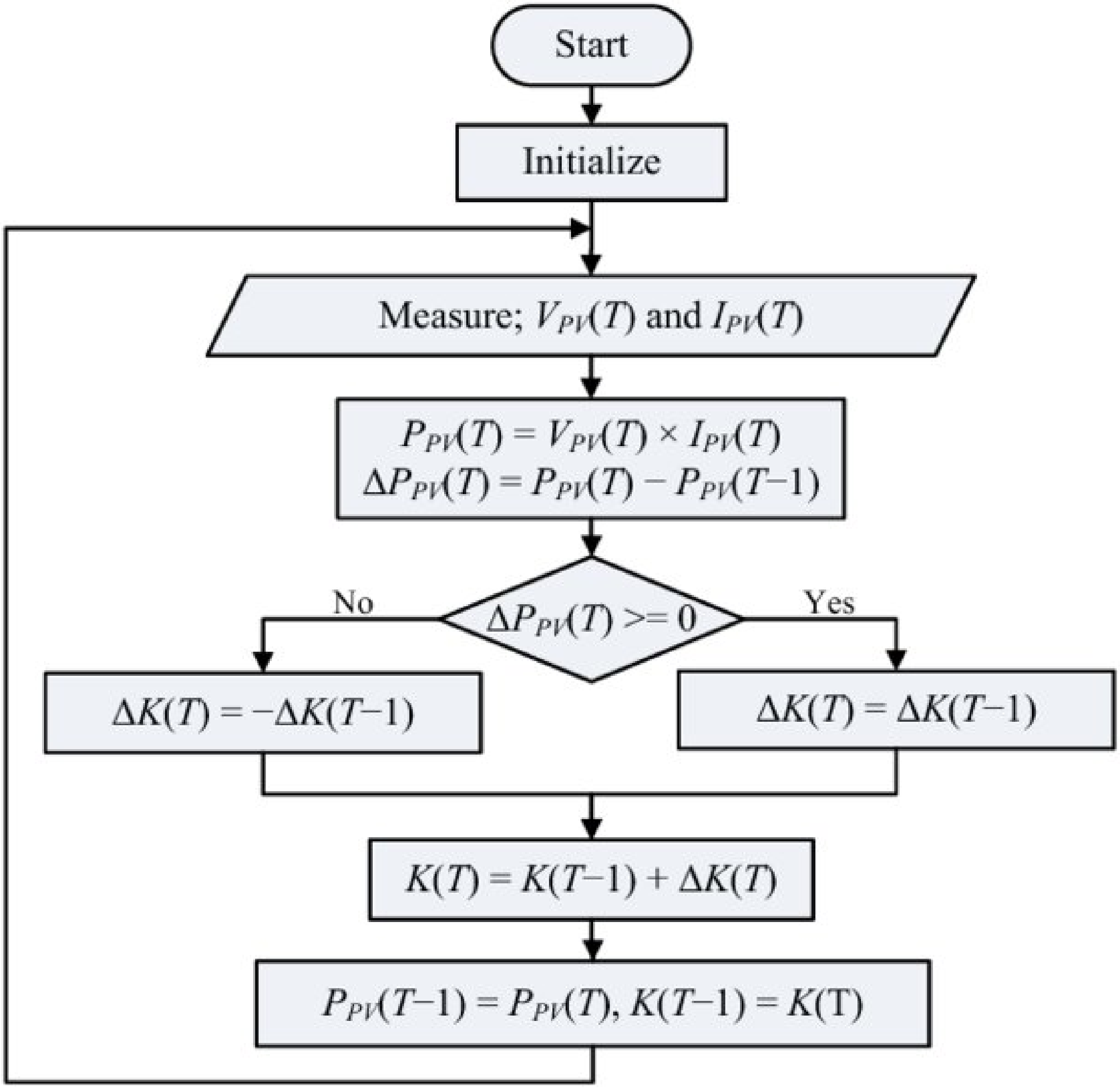

Start
Initialize
Measure; $V_{PV}(T)$ and $I_{PV}(T)$
$P_{PV}(T) = V_{PV}(T) \times I_{PV}(T)$
$\Delta P_{PV}(T) = P_{PV}(T) - P_{PV}(T-1)$
No
$\Delta P_{PV}(T) >= 0$
Yes
$\Delta K(T) = -\Delta K(T-1)$
$\Delta K(T) = \Delta K(T-1)$
$K(T) = K(T-1) + \Delta K(T)$
$P_{PV}(T-1) = P_{PV}(T), K(T-1) = K(\mathrm{T})$

Figure 4.15: Pertub and observe Algorithm flowchart

Incremental Conductance

The principle of this algorithm is based on the knowledge of the value of the conductance $G = \frac{I}{v}$ and on the increment of the conductance dG to deduce the position of the point of operation relative to the maximum power point, 'MPP'. If the increment of the conductance (dG) is greater than the opposite of the conductance (G), the ratio is reduced cyclic.

On the other hand, if the conductance increment is less than the opposite of the conductance, the duty cycle is increased. This process is repeated until reaching the maximum power point, 'MPP'[12]

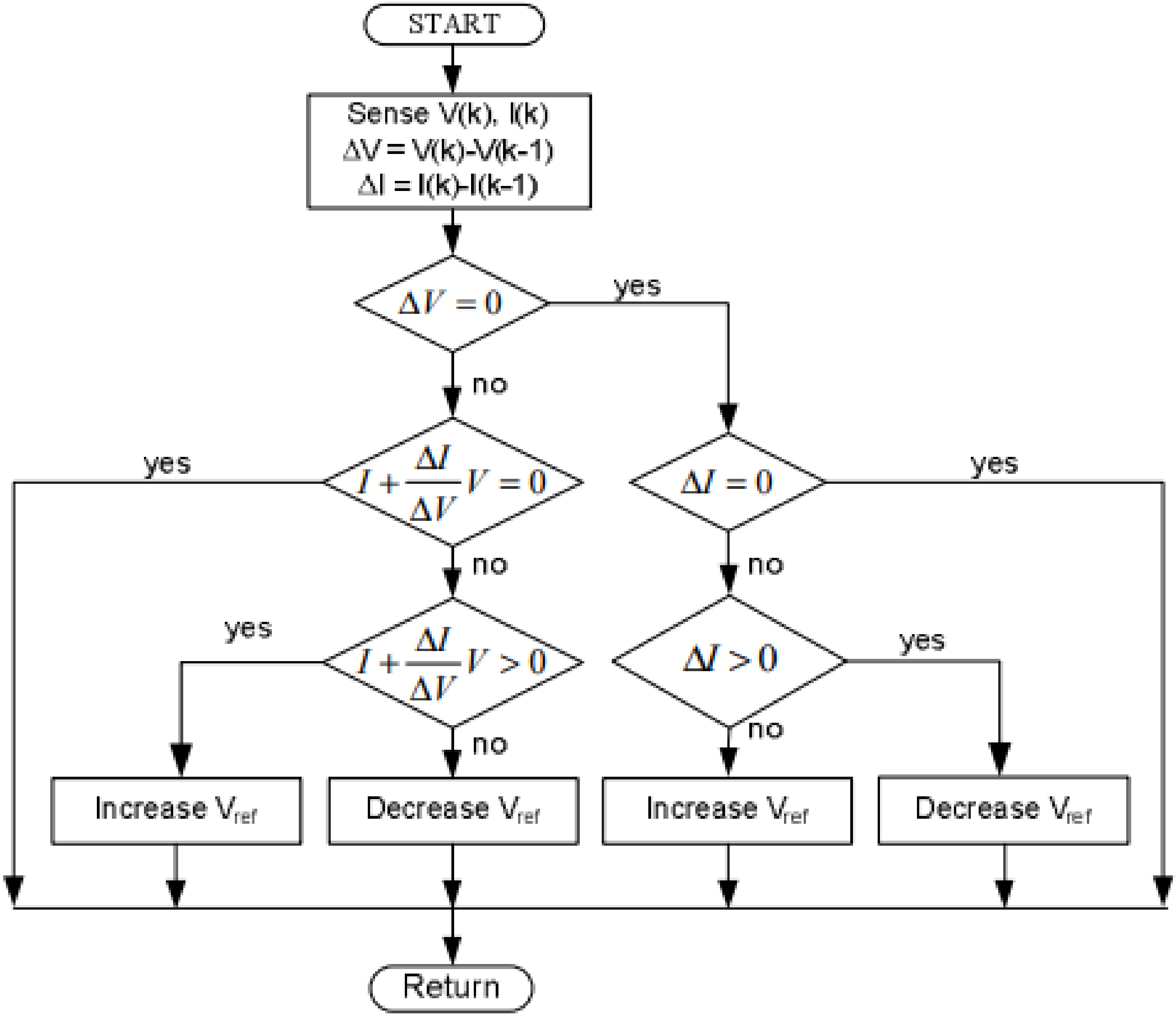


Figure 4.16: Incremental Conductance Algorithm flowchart

adding to pertube and observe , incremental inductance there is also the algorithms based on measuring open circuit voltage and short circuit current

Hill climbing

The basic idea of the HC (Hill Climbing) method is the same as PO method. It tests if P(n) is greater than P(n-1) or not, to reach MPP. The PO method uses instead a test on dP/dV to determine whether the maximum power point has been found or not. However, the HC method uses a test condition on P(n)-P(n-1).

in addition there is also the robust and adaptive control techniques such as : Extremum Seeking Control Method (ESC),RSMCA: Robust Sliding Mode Control Algorithm,RUCA: Robust Unified Control Algorithm

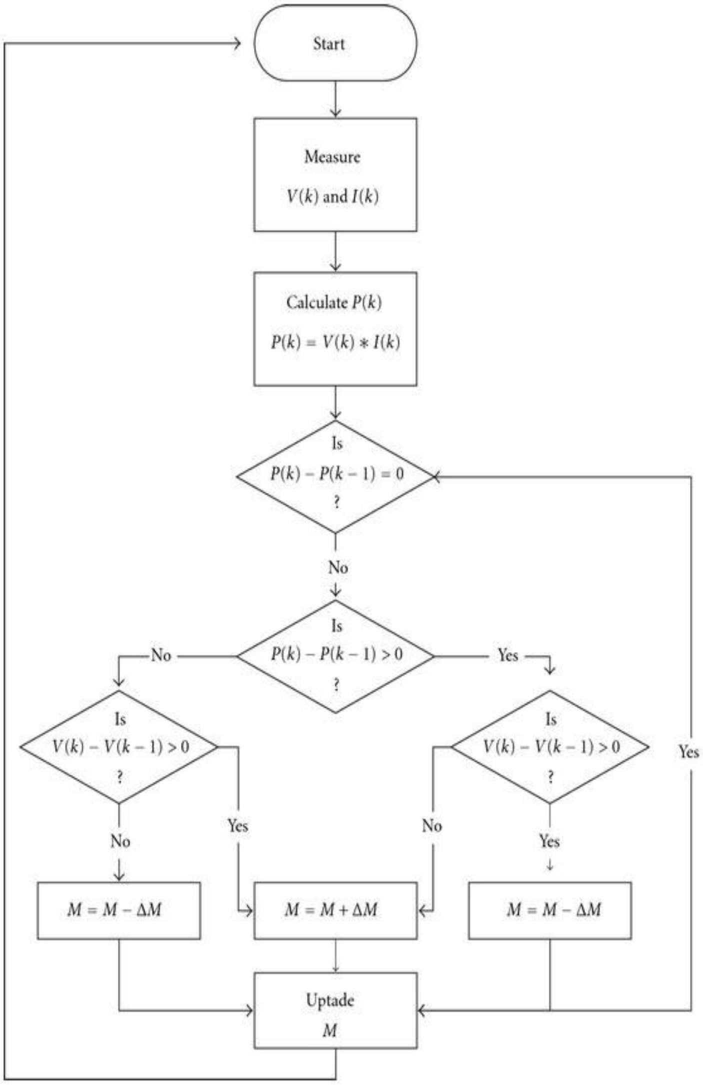


Figure 4.17: Hill and climbing Algorithm flowchart

4.3.3 Inconventional Methods(Open Loop)

Fixed Operating Point

The problem is to find the voltage VMPP or the current IMPP at which the PV array delivers the maximum power under a given temperature T and irradiance G. Then the method automatically puts the PV in this condition. The first remark was that the MPP varies in a small region and that on the left part of the P-V characteristic, the slope P/V is roughly constant.

Artificial Neural Network(ANN) for MPPT

ANN are well known to provide universal approximators providing non-linear models which are complementary to the conventional modeling techniques. Back propagation ANN are used as pattern classifier or as non-linear layered feed-forward networks to give a global approximations to a non-linear inputoutput mapping (Reisi et al., 2013). The first application of ANNs to MPPT, has been proposed by Hiyama et al (Hiyama and Kitabayashi, 1997).

In general a three layer structure, i.e. input layer, hidden layer and output layer are used with the back propagation. After a good learning, ANN are able to make generalizations in regions of the phase space where little is known or no data are available. The Neural network is composed by neuron cells, placed in 3 layers (or may be more) connected to all neurons through weights see figure below. The input variables are PV parameters like VOC and ISC, atmospheric data (Irradiance and Temperature). The output of ANN gives reference signals, like the reference voltage or the duty cycle signal used to drive the power converter to operate at or close to the MPP.

The three layers of neural network have a hyper tangent sigmoid function (Noguchi et al., 2002). The algorithm used for training is back-propagation. The back-propagation training algorithm needs inputs and the desired output to adapt the weight by MSE. The characteristics of a PV array are nonlinear and time-varying, this implies that the neural network has to be trained to guarantee accurate tracking of MPP. This is a time consuming process. Note also that it can use as inputs the voltage and current measurement, to become a closed loop method or a combination of both.

Fuzzy Logic MPPT

Fuzzy logic controllers offer the advantage of working capability with imprecise inputs, and do not need an accurate mathematical model. They can handle nonlinearities, and have fast convergence. Their learning ability and accuracy depend on the number on the fuzzy levels and the the membership functions. The decision-making uses rules specified by a set of IF–THEN statements to define the control which produce the desired behavior. The defuzzification stage, operates the reverse function to get numerical variables for analog control using the membership function.

In order to track MPP, the error is computed based on irradiance and temperature or instantaneous values such as power and voltage (Algazar et al., 2012). The output signal is either the duty cycle itself, or VMPP and IMPP reference to generate the duty cycle. The membership function associated with fuzzification and defuzzification, as well as the antecedent and the consequent fuzzy rules are determined by trial and error. This can be time-consuming. This method can be used in open loop or in closed loop when using as feedback (in real time) the output variables like current, voltage and the power.

Bio Inspired Algorithms based MPPT

the methods bio-inspired are considered to be very effective in treating the characteristic curve P-V in partial shade conditions. Because they deal with problems correctly nonlinear and stochastic optimization and show excellent performance without involving heavy mathematical calculations, which results in simplicity of calculation, easy understanding, reliability and quick response.

Among the controllers designed based on bio-inspired algorithms, we can cite:

- MPPT controller based on the Ant Colony Algorithm (ACO) [13],
- MPPT controller based on Particle Swarm Optimization (PSO) algorithm [14],
- MPPT controller based on the Artificial Bee Colony (ABC) algorithm [15],
- MPPT controller based on the Cuckoo Search Algorithm (CSA) [16],
- MPPT controller based on the Firefly Algorithm (FA) [17],
- Bat Swarm Optimization Algorithm based MPPT controller (BSO) [17],
- MPPT controller based on Flower Pollination Algorithm (FPA) [17],
- MPPT controller based on the Gray Wolf Optimization (GWO) algorithm [17],
- MPPT controller based on the Generalized Pattern Search (GPS) algorithm [17],
- MPPT controller based on Shuffled Frog Leaping Algorithm (SFL) [17]

4.4 Conclusion

In this chapter we Introdued photovoltaic systems from it's different components and circuits to the MPPT algorithm which is used to attain the maximum power of the panel ,presenting the different algorithms used before to the state of the art

In the next chapter we will be focused on the Particle Swarm optimization adding to it a quantum behaved variation

Chapter 5

Quantum MPPT Algorithm

5.1 Introduction

This nature is all the more present for quantum-scale objects. When the idea of a wave-particle duality emerged, scientists and engineers started implementing this idea for real-world applications. One of those applications is the QPSO algorithm that is based on Shrödinger's equation and quantum mechanics. Simulating the behavior of human intelligence, instead of that of a bird flock or fish schooling, requires the thinking mode of an individual of the social organism which is not sufficient to be described by using a linear evolution equation. It is believed that human thinking is uncertain as if a particle having quantum behavior . This is why the QPSO algorithm has hugely improved the PSO algorithm.

5.2 Quantum Behaved Particle swarm optimization

5.2.1 Description

QPSO is proposed by Sun et al in 2004 and is based on the elemental theory of particle swarm and rules of quantum mechanics. Here all the particles have the features of quantum deportment. Moreover, researches in the past shows that global optimization performance of QPSO is superior to that of the standard PSO algorithm.[18]

In general QPSO is an introduction of quantum computing to the particle swarm starting from a mechanical point of view that the particle in the space has a quantum behavior.

5.2.2 From PSO to QPSO

In the quantum model of PSO called QPSO ,the state of the particle is defined by a wave function (Schrodinger 's equation) $|\psi]^2$ instead of the position and velocity,in addition the dynamic behavior of the particle is widely divergent from the classical one in that the exact

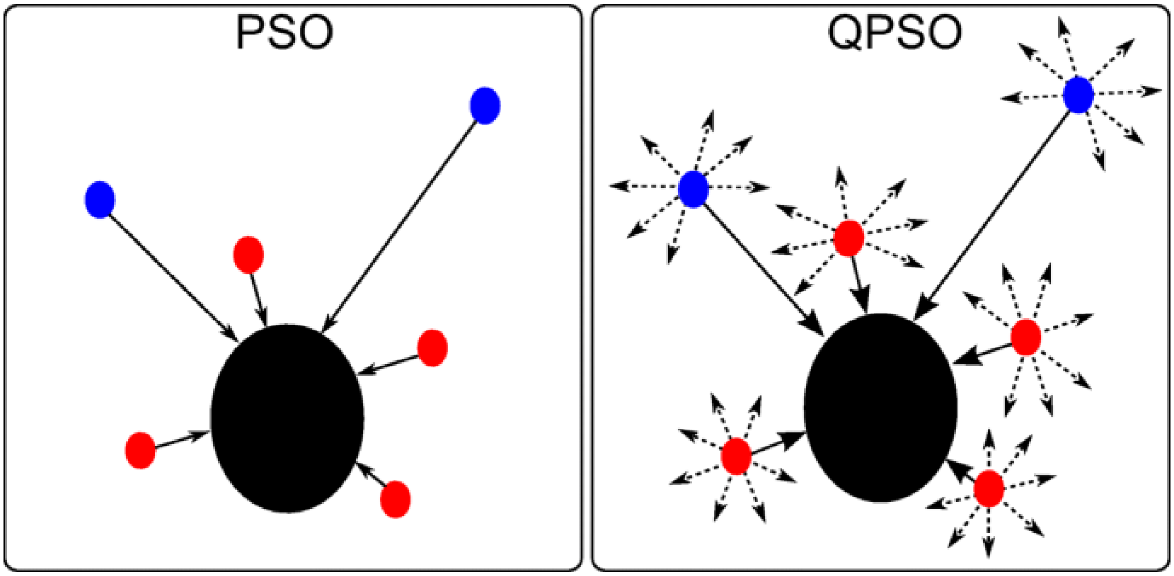


Figure 5.1: From PSO to QPSO

values of x_i and v_i cannot be determined simultaneously (Heisenberg) in this context the probability of particle's location is defined by the density probability $|\psi]^2$

In other words the particle swarm is applied to a quantum space defined by a wave function

$$|\psi]^2 dxdydz = Qdxdydz \tag{5.1}$$

Q is probability density function which satisfies the normalisation condition

From the recent literature about quantum particle swarm optimisation there is 03 different variants :

Quantum Delta Particle Swarm optimization QDPSO : Delta is for the Potential well function of the particle

Revised Quantum Particle Swarm optimization RQPSO :it's an update for the previous approach introducing the mean best to the formula

Gaussian Quantum Particle Swarm optimization GQPSO : We change the formula of the local attractor and particle position by introducing a normal distribution

for this work we the first variant using the delta potential function

The protocol dealing with the movement of the particles in QPSO is quite different from the particles in standard PSO. In accordance with the unpredictability theory of quantum world, particle becomes visible at any position of search space with a certain probability resulting in the fact that the position and velocity of that particle cannot be found at the same instant. Different steps associated with QPSO are detailed below:[18]

let's assume

$n = number of particles$

$D = dimension associated with the problem$

$X_i = (x_i 1, x_i 2, ..., x_i d)$

$pbest = P_i = (p_i1, p_i2, ..., p_id)$
$gbest = p_g = (p_g1, p_g2, ..., p_gd)$

1. Generate initial population of particles randomly between the minimum and the maximum operating limits in the D-dimensional space.[19]

2. Estimate the fitness value of every particle.[19]

3. Present fitness value of the particle is compared with the personal best (pbest) of every particle. If the present fitness value of the particle is superior, then allocate the present fitness value to pbest and allocate the present coordinates to pbest coordinates.[19]

4. Compute the local attractor from the pbest and global best

   $$L_a = \frac{c_1 r_1 p_i + c_2 r_2 p_g}{c_1 r_1 + c_2 r_2} \tag{5.2}$$

   $$L_{ai} = \phi p_i + (1 - \phi) p_g \tag{5.3}$$

   $\phi = U(0, 1)$

   Using the local attractor means that each particle takes the weighted average between the global best position and pbest position

5. In the overall population, determine the present best fitness and its coordinates. If the present best fitness value is superior to global best (gbest), then allocate the present best fitness value to gbest and allocate the current coordinates to gbest coordinates.[19]

6. update local attractor and position

   The position of particle is updated using the Monte Carlo method:

   $$x_{id} = p_{id} \pm \beta.|p_{id} - x_{id}|.ln(\frac{1}{u})$$

(5.4)

   β is the expansion/shrinkage factor (positive real number) set as :

   $$\beta = 0.5.(\frac{T - t}{T} + 1) \tag{5.5}$$

   This factor is used to balance between the global and local searching ability ,t is the current iteration ;T is the maximum iteration

The local attractor is used to guarentee the convergence of the Particle to destined position,replacing it in the 4.4 equation will result:

$$\mathrm{x}_{id} = L_{ai} - \beta.|L_{ai} - x_{id}|.ln(\tfrac{1}{u}) \quad foru < 0.5$$
$$x_{id} = L_{ai} + \beta.|L_{ai} - x_{id}|.ln(\tfrac{1}{u}) \quad otherwise$$

(5.6)

7. the criteria of convergence is the maximum iteration number ,when the algorithm attain it ,the process will be stopped [19]

5.2.3 Application to MPPT

PROCESS

Photovoltaic cell characteristic is non-linear whose output power varies as function of the irradiation and temperature. The reason for the non-linear is because, the value of irradiation and temperature changes throughout the day, thus decreasing the output efficiency. Also, the efficiency of these Photovoltaic modules is not satisfied by the power requirement. Hence, an algorithm to increase the efficiency of the PV module needs to be designed to solve the discrepancies among the efficiency of the cell and the power requirement. Hence, the objective of this project is to design and model the Particle Swarm Optimization assisted MPPT algorithm and enhance the efficiency of the photovoltaic system.

Step 1 (Parameter Selection): In the proposed system, the particle position is defined as the duty cycle value d of the dc–dc converter, and the fitness value evaluation function is chosen as the generated power P. From the algorithm point of view, a larger number of particles result in more accurate MPP tracking even under complicated shading patterns. However, a larger number of particles also lead to longer computation time. Therefore, a trade-off should be made to ensure good tracking speed and accuracy.[20]

Step 2 (QPSO Initialization): In QPSO initialization phase, particles can be placed on fixed position or be placed in the space, randomly. Basically, if there is information available regarding the location of the Global MPP in the search space, it makes more sense to initialize the particles around it. The particles are initialized on fixed positions which cover the search space [Dmin, Dmax] . Dmax and Dmin are the maximum and minimum duty cycle of the utilized dc-dc converter, respectively.

Step 3 (Fitness Evaluation): The goal of the proposed MPPT algorithm is to maximize the generated power PPV. After the digital controller output, the PWM command according to the position of particle i (which represents the duty cycle command), VPV and current IPV can be measured and these values can then be utilized to calculate the fitness value PPV of particle i. It should be noted that in order to acquire correct samples, the time interval between successive particle evaluations has to be greater than the power converter's settling time.

Step 4 (Update Individual and Global Best Data): If the fitness value of particle i is better than the best fitness value in history $p(best, i)$, set current value as the new $p(best, i)$. Then, choose the particle with the best fitness value of all the particles as the g_{best}. This step is similar to step 3 of the standard PSO method.

Step 5 (Position of Each Particle): After all the particles are evaluated, the position of each particle in the swarm should be updated.

Step 6 (Convergence Determination): if the maximum number of iterations is reached, the proposed MPPT algorithm will stop and output the obtained g$_bestsolution$.

Step 7 (Re-initialization): Typically, QPSO method is used to solve problems that the optimal solution is time invariant. However, in this application, the fitness value (global maximum available power) often changes with environments as well as loading conditions. In such cases, the particles must be reinitialized to search for the new GMPP again. The following constraint is utilized to detect the insulation change and shading pattern changes.

5.2.4 Simulation results

In this section we will apply the Classical swarm optimization optimization using the approach used in the paper [14] and the quantum appraoch in standard and non standard Conditions.

Using a Simulink block for the PV model and Matlab functions for both PSO and QDPSO we obtain as follow.

The software used is Matlab SIMULINK ,creating 4 PV modules in series

Bypass Diode

A Bypass Diode is used in general to ensure safety of the circuit in case of partial shading ,Hot spots in some PV panel .It provides a weak resistance path for electrons

Also it recovers the power loss in Partial Shading conditions

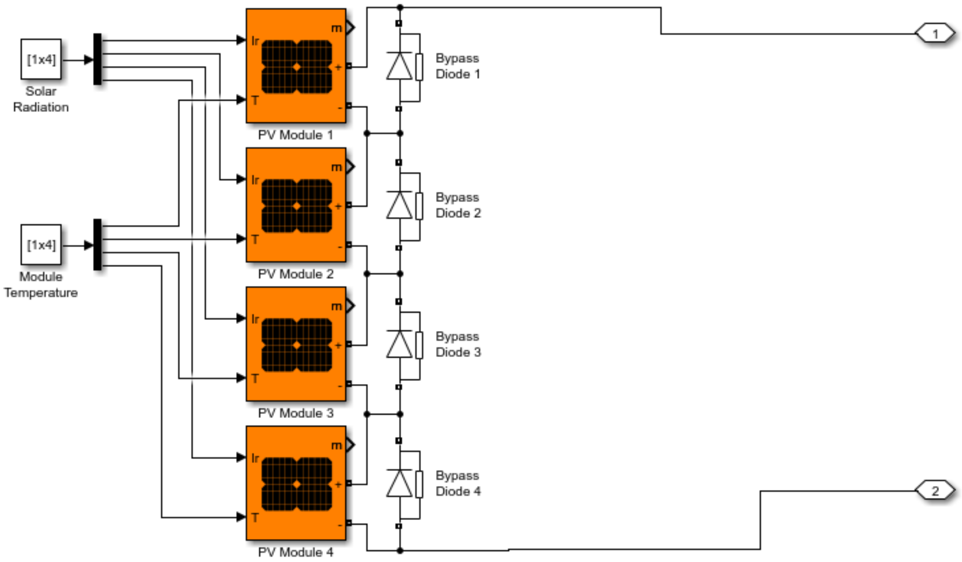


Figure 5.2: Simulink Block of PV array

The Full System Block for PSO

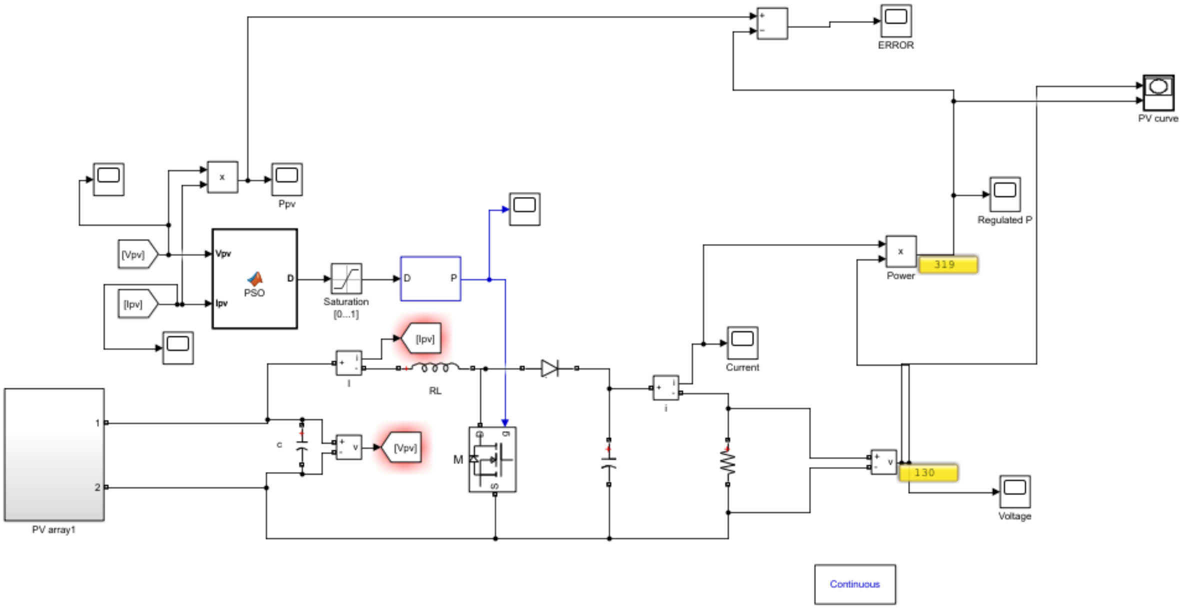


Figure 5.3: PV PSO MPPT system

The Full System Block for QPSO

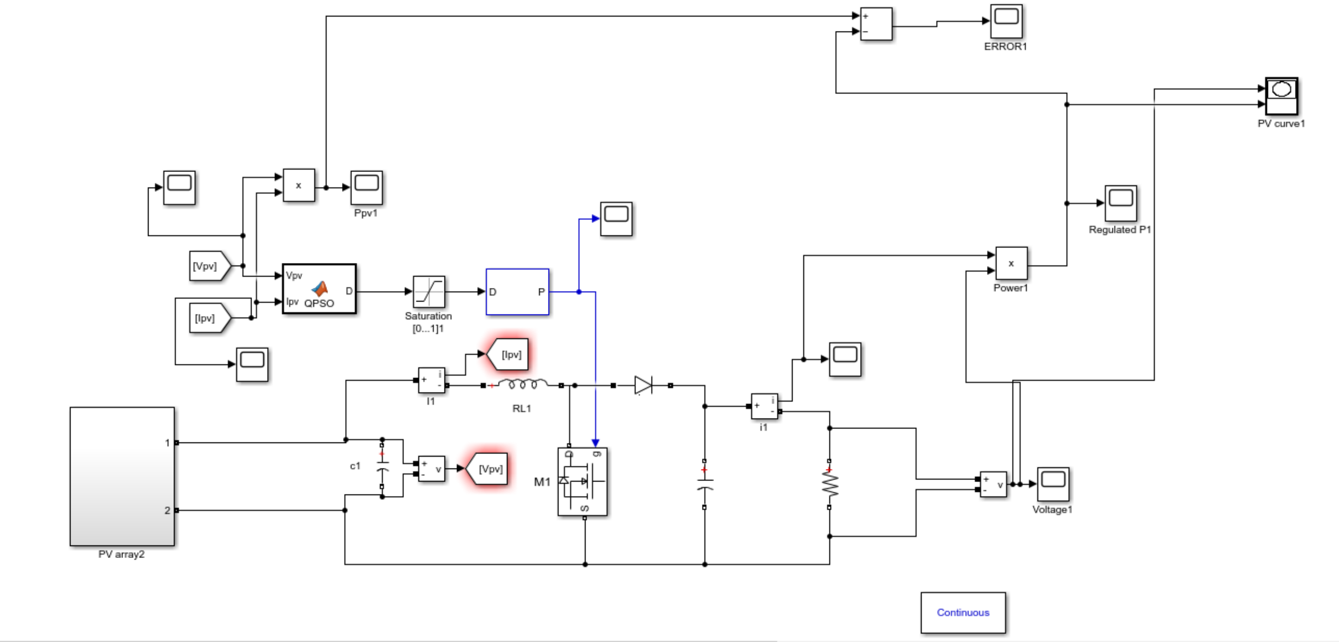


Figure 5.4: PV QPSO MPPT system

1. T=25°,Ir=500 W/m^2 (Standard Conditions)

   Photovoltaic power curve in standard conditions

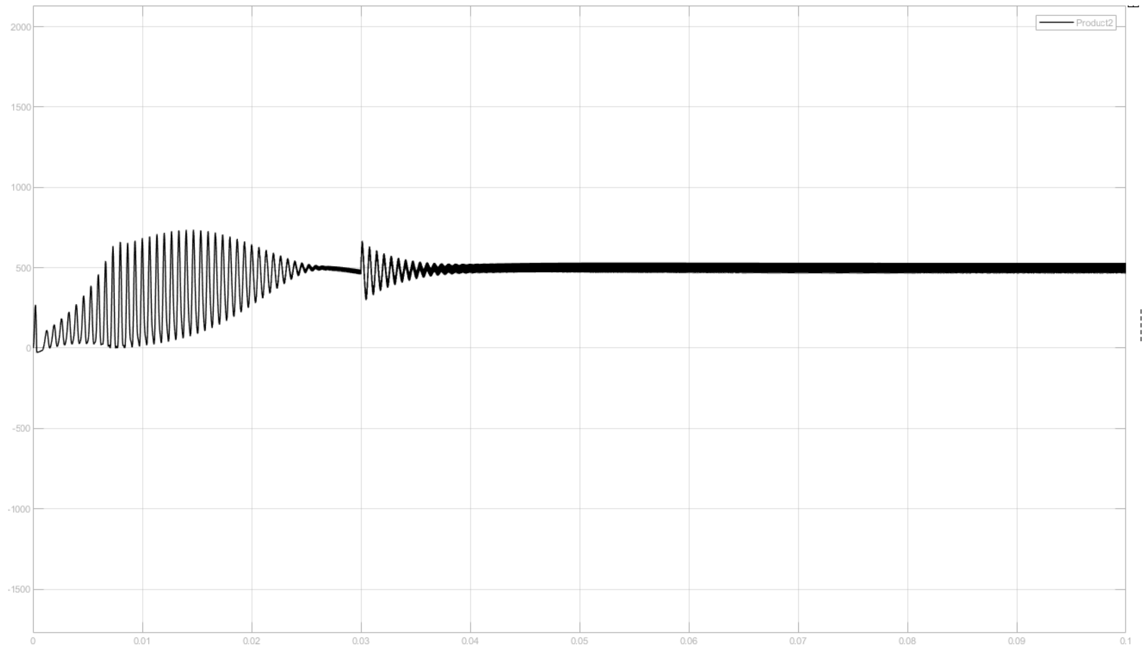


Figure 5.5: PV Power

PSO MPPT Algorithm :

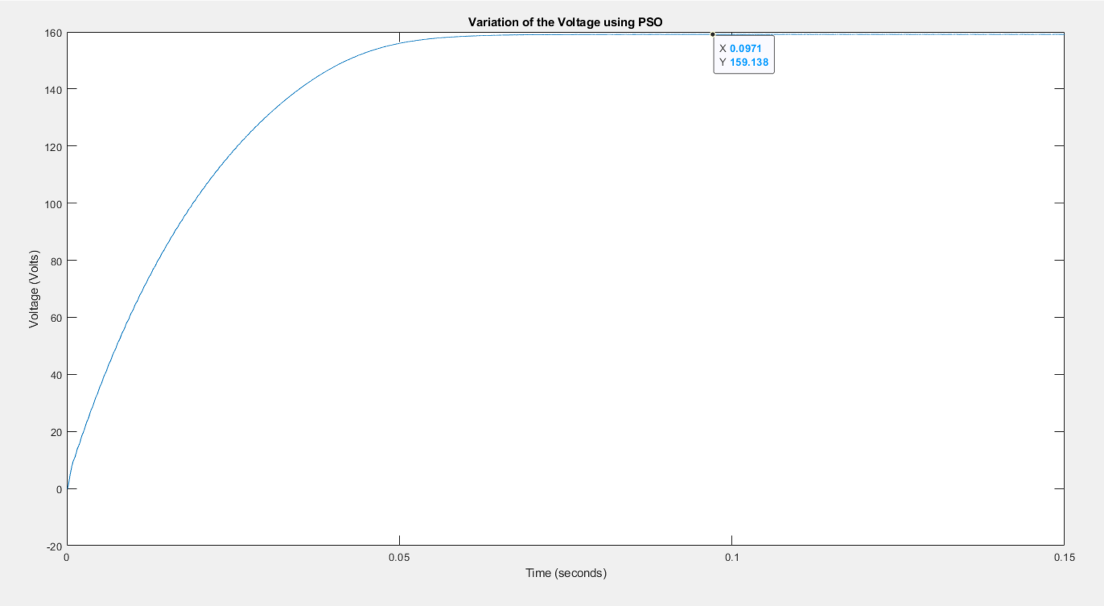


Figure 5.6: Voltage

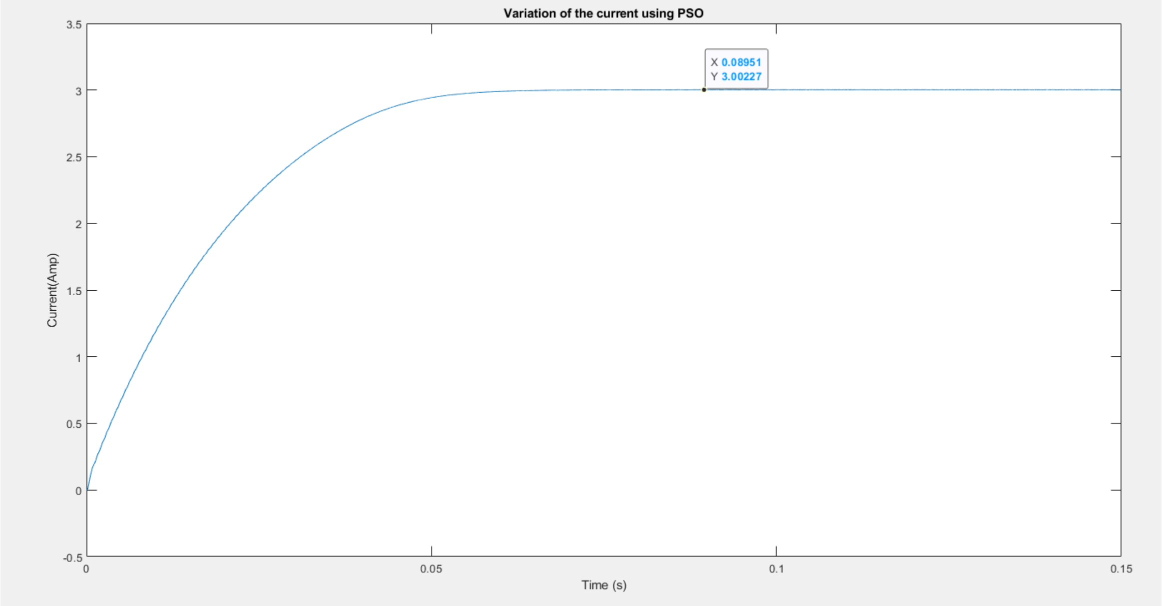


Figure 5.7: Current

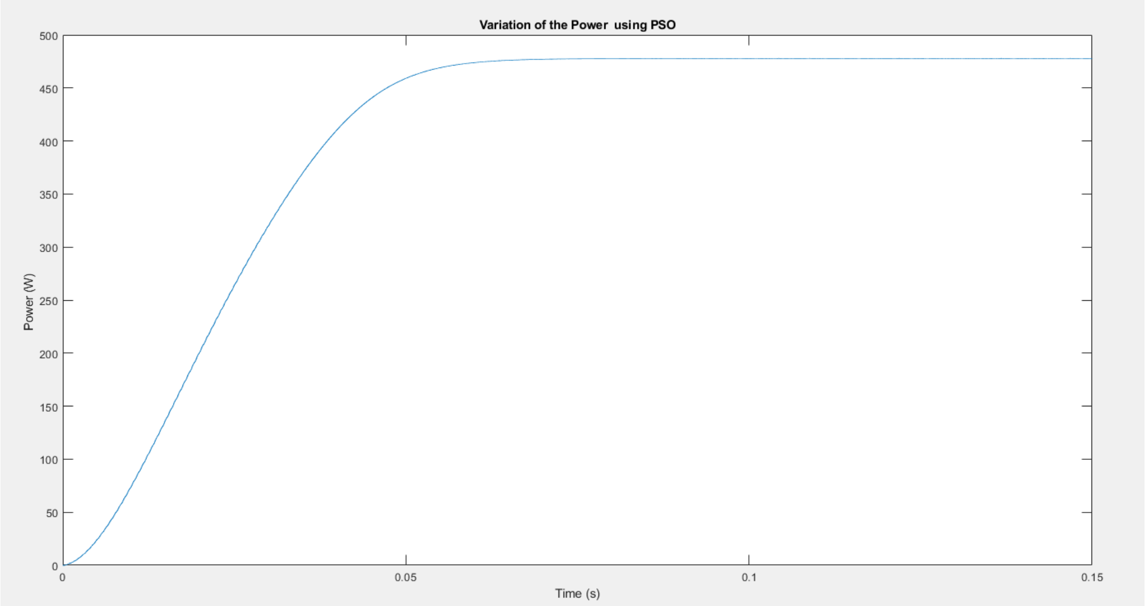


Figure 5.8: Power

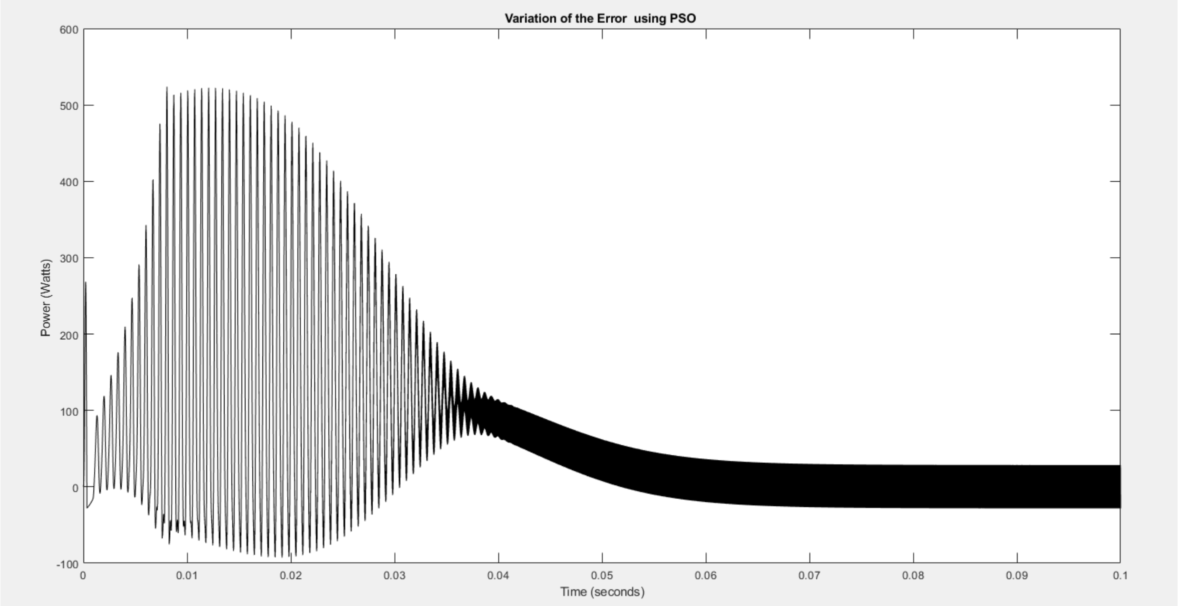


Figure 5.9: Error between Power of PV and generated using PSO

QPSO MPPT Algorithm :

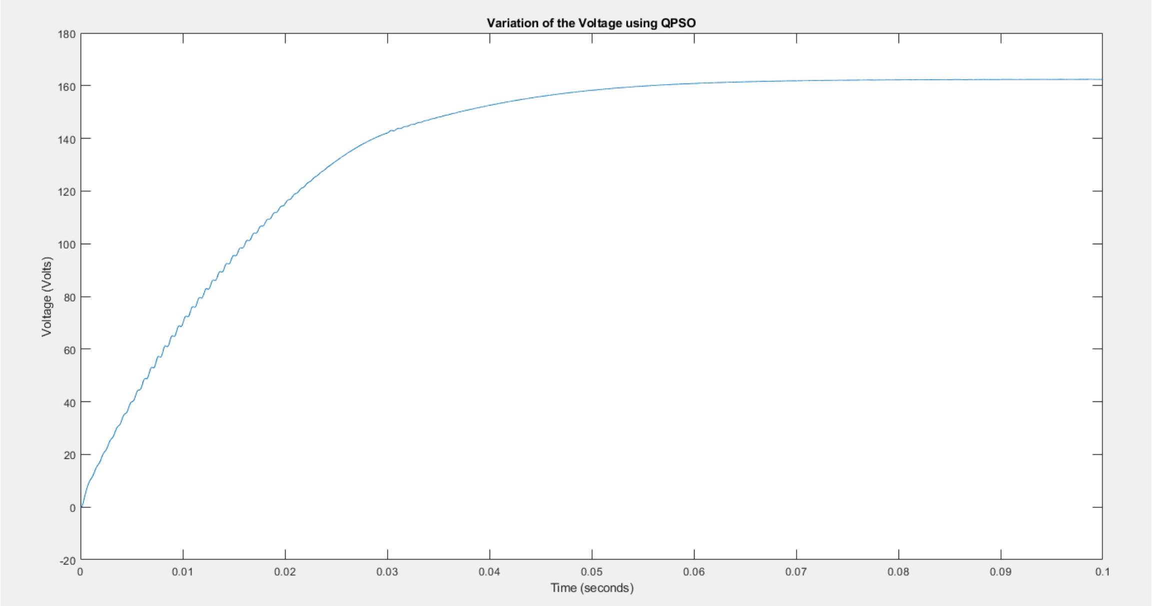


Figure 5.10: Voltage

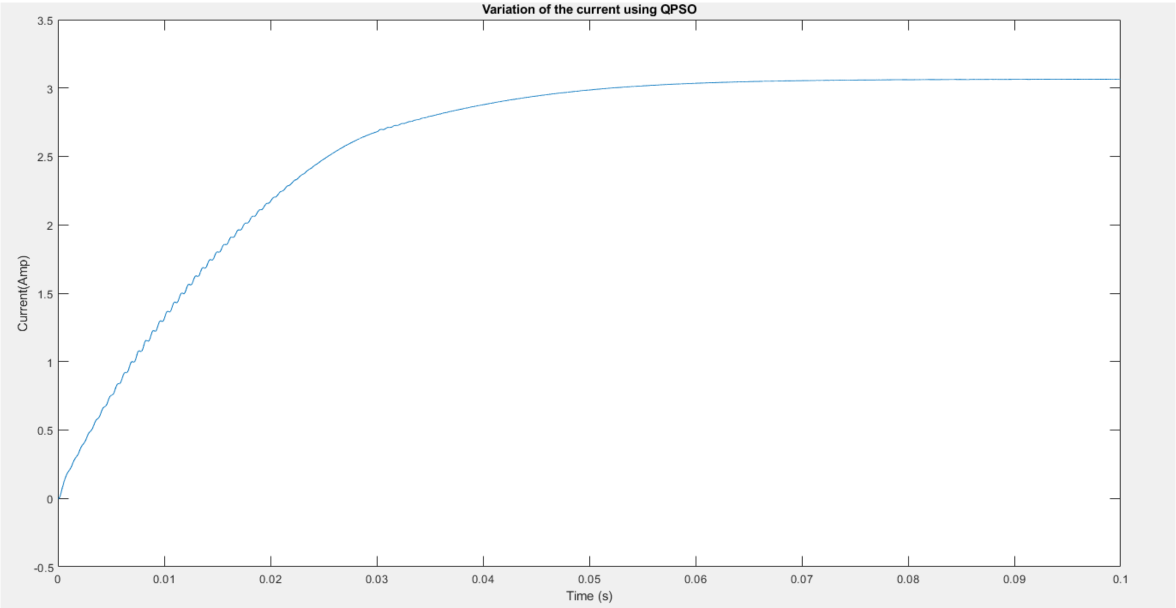


Figure 5.11: Current

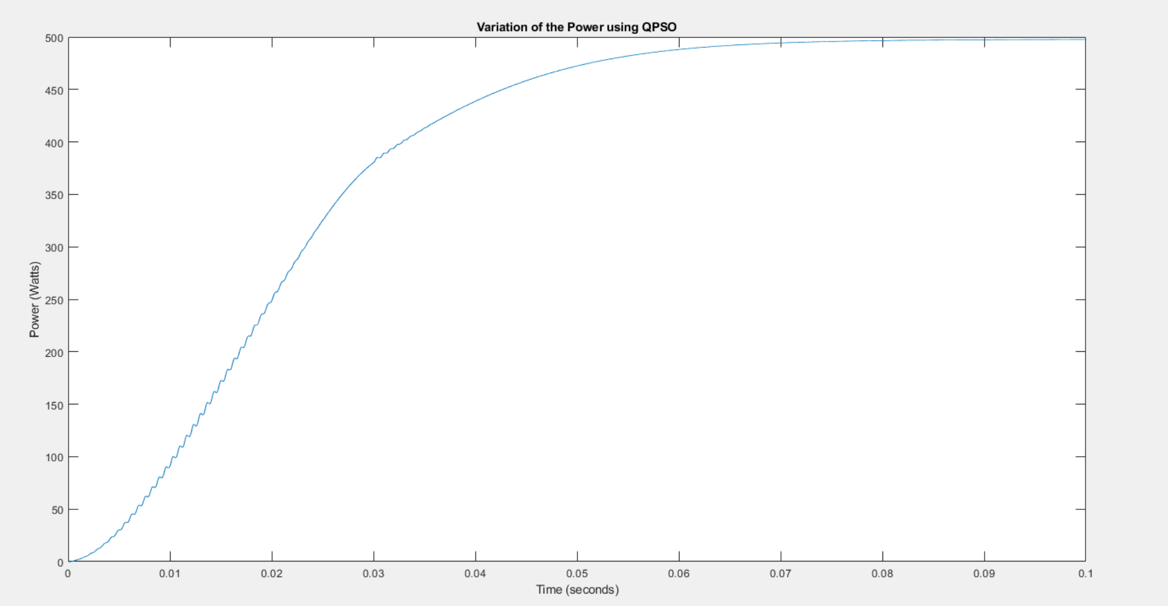


Figure 5.12: Power

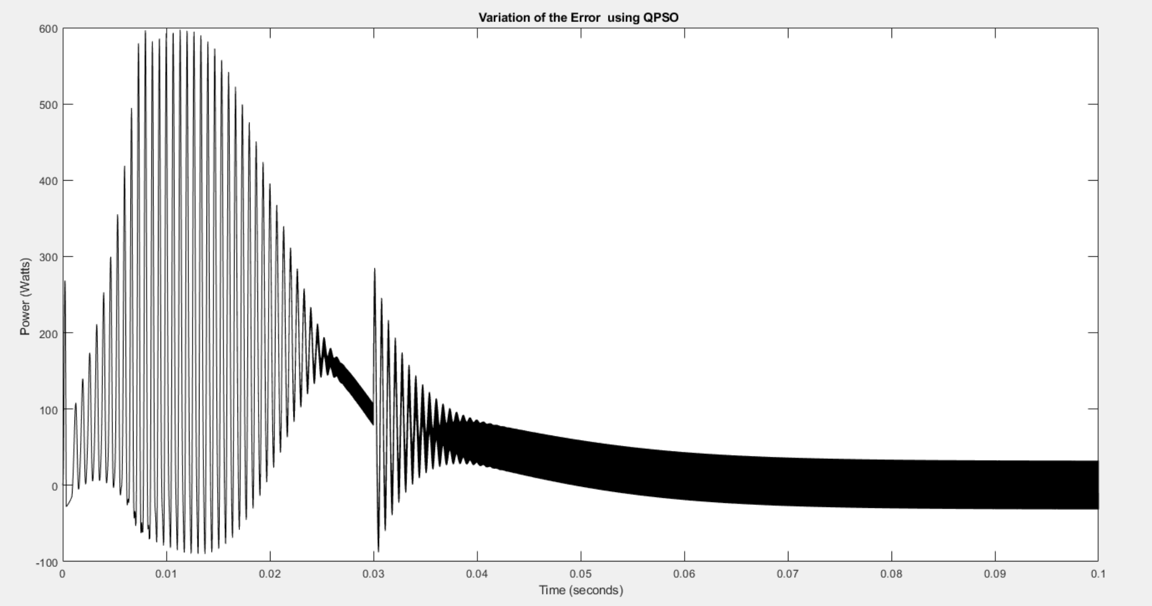


Figure 5.13: Error between Power of PV and generated using PSO

QPSO Vs PSO :

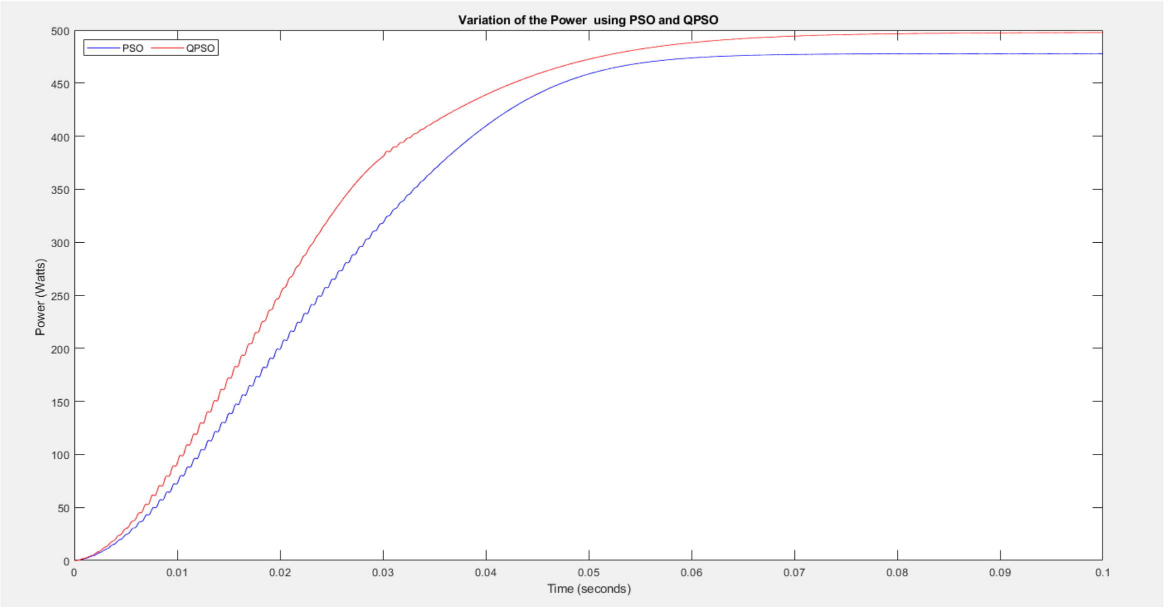


Figure 5.14: Variations of power using PSO and QPSO

Discussion

- Time convergence for PSO curve is about 0.07 seconds ,reaching P=478 Watt

- Time convergence for QPSO curve is about 0.04 seconds ,reaching P=500 Watt

-the power of PV panel reaches it's maximum at 500 Watt,so we observe that QPSO results are more precise and faster

2. T=40°between 0.04 and 0.075,Ir=500 W/m^2 (High Temperature)

Photovoltaic power curve in High Temperature

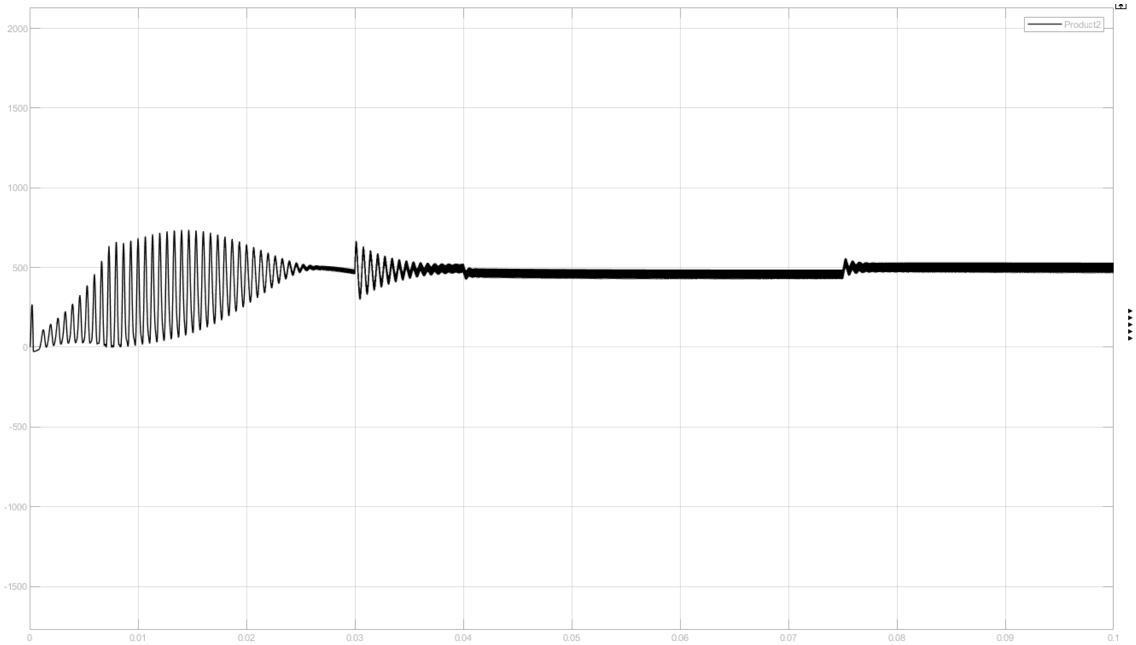


Figure 5.15: PV Power

PSO MPPT Algorithm :

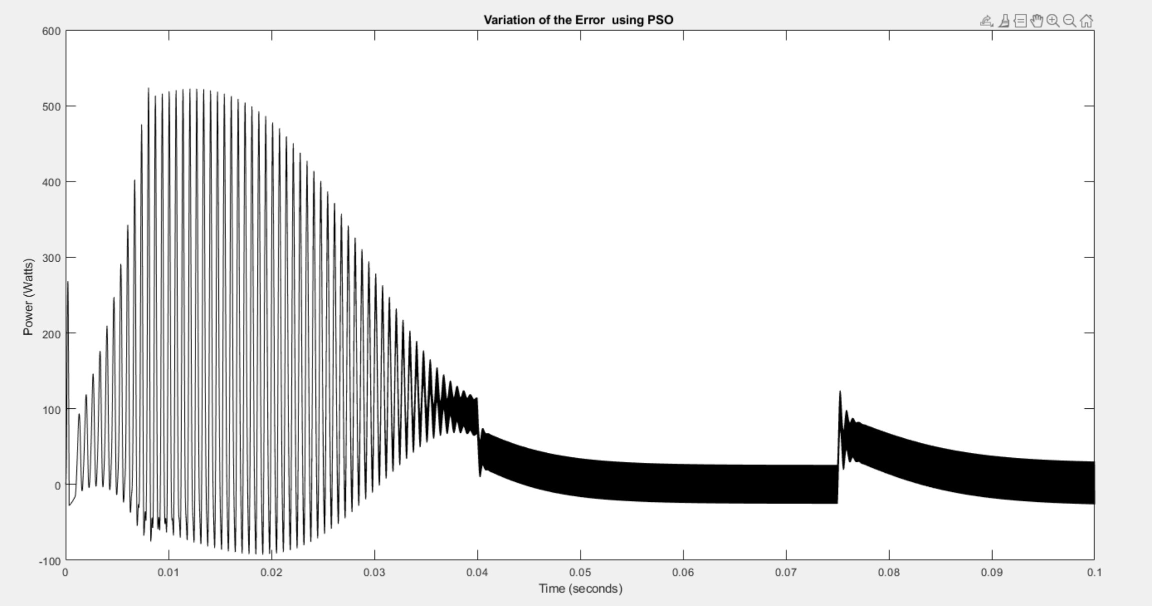


Figure 5.16: Error between Power of PV and generated using PSO

QPSO MPPT Algorithm :

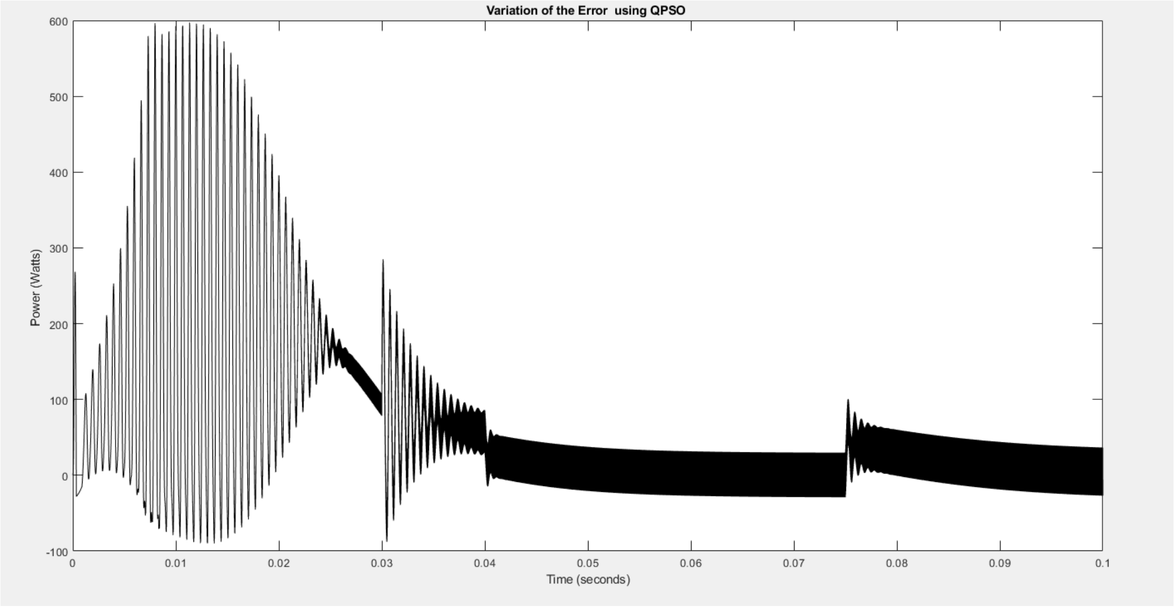


Figure 5.17: Error between Power of PV and generated using QPSO

QPSO Vs PSO :

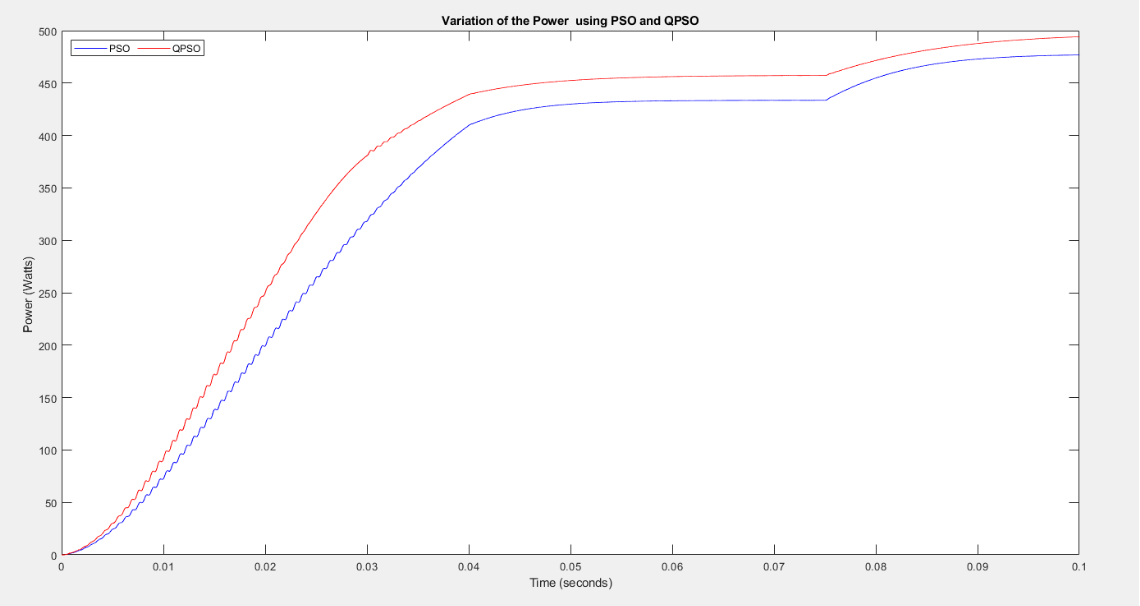


Figure 5.18: Variations of power using PSO and QPSO

Discussion

- Time convergence for PSO curve is about 0.07 seconds ,reaching P=478 Watt after the temperature perturbation

- Time convergence for QPSO curve is about 0.04 seconds ,reaching P=500 Watt after the temperature perturbation

-While the temperature perturbation ,the decrease for QPSO curve is less than the PSO curve -the power of PV panel reaches it's maximum at 500 Watt,so we observe that QPSO results are robust and precise

3. T=25°,Ir=500 W/m^2 For PV module 1 and 4 ,Ir=400 W/m^2 For PV module 2 and 3 (Partial Shading)

   PSO MPPT Algorithm :

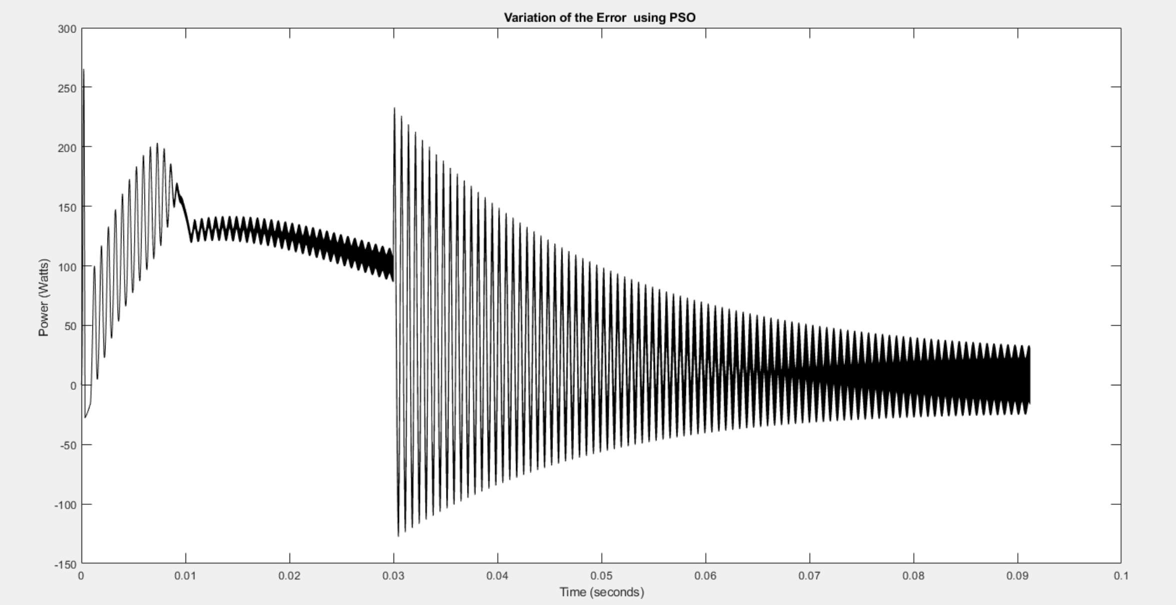


Figure 5.19: Error between Power of PV and generated using PSO

QPSO MPPT Algorithm :

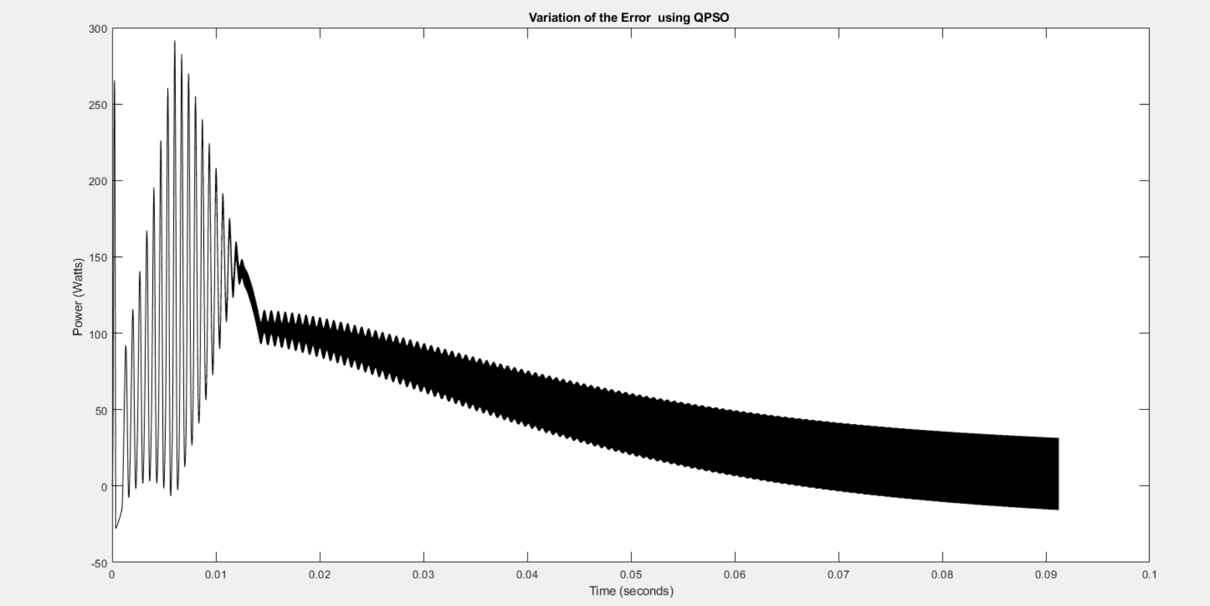


Figure 5.20: Error between Power of PV and generated using QPSO

QPSO Vs PSO :

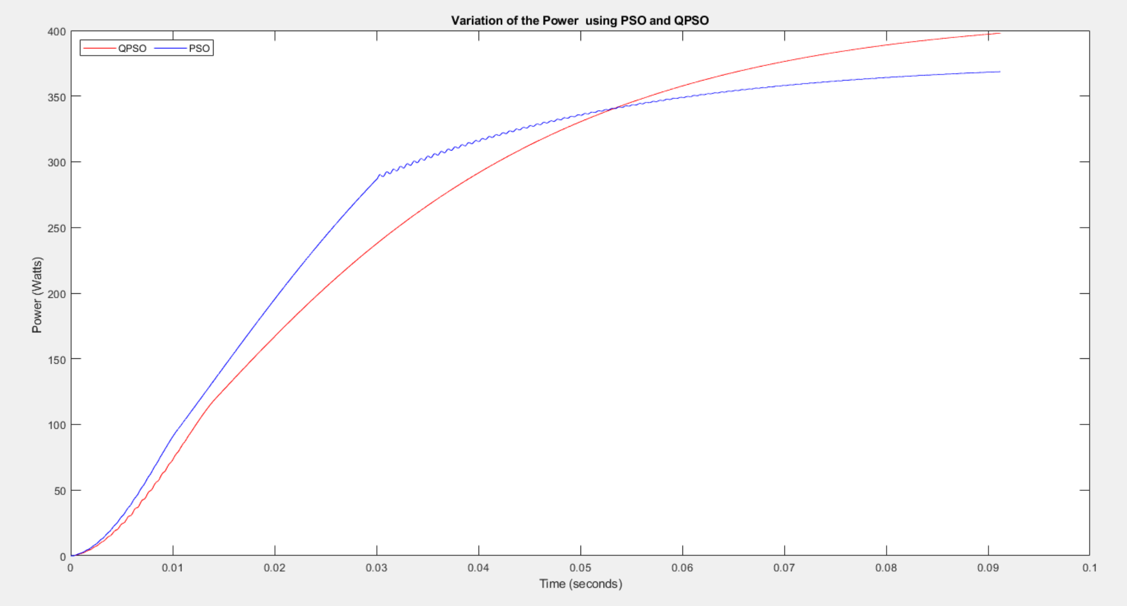


Figure 5.21: Variations of power using PSO and QPSO

Discussion

-We observe that the speed of QPSO decreased with the partial shading but the precision is still high ,nevertheless PSO algorithm maintained the speed while loosing the precision

4. changing expansion factor ,$Iter_1 = 1500$,$Iter_2 = 500$

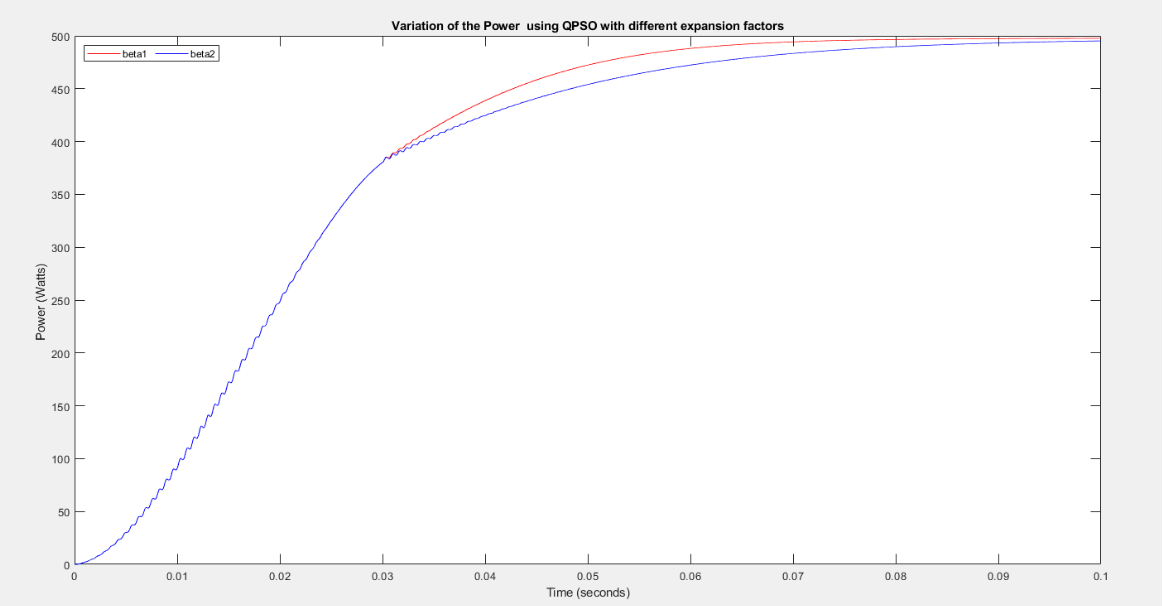


Figure 5.22: Power

Discussion

-The more iteration you make the better beta will be adapted and that implements a higher accuracy

5.2.5 General Discussion

Knowing that the global MPP for the system is 500V which is Shown in in the Ppv curve .

For the standard variations we see that QPSO 's graph is fast and precise than the PSO 's

Increasing the Temperature in a period will decrease the power in general ,which is shown in the previous curves ,nevertheless ,the QPSO 's curve recovered quicker and accurate than the PSO which shows good robustness for both algorithms but with an upper hand to Quantum approach

Partial Shading decrease the power which is also demonstrated in the previous curves ,The observation showed that QPSO results were more accurate than the classical swarm approach ,In the other Part some speed is lost .

The only adjustable parameter in QPSO is β the expansion factor , which is relative to the iteration number .Increasing it from 500 to 1500 made the curve more accurate to the MPP goal

5.2.6 Comparative Study

Algorithms	Accuracy	Convergence Time	Robustness	Parametric Controllability
PSO	good	fast	good	Position and Velocity to update,plus too many parameters
QPSO	high	Super Fast(standard)	highly robust	Only position to update , One parameter to adjust "the expansion factor"

5.3 Simulation of Quantum swarm particle optimization on a quantum computer

To realize the quantum-behaved particle swarm optimization algorithm, we need to implement Schrödinger's equation on a quantum computer. On each iteration, the wave function that describes the particles is updated until the probability density function collapses to one point. We start by discretizing space. Hence, the particles move on a lattice with l lattice sites in each direction.

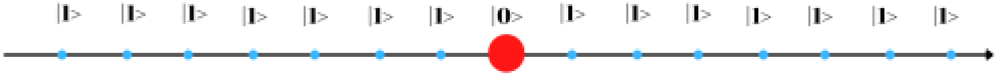


Figure 5.23: Represenation of a particle on 1D lattice

Let's first consider one particle moving in a one-dimensional space without any potential. Each lattice point is associated with a qubit. The qubit is either $|0>$ or $|1>$ depending on whether the particle is situated on that point.[21]

We consider the following equation that is capable of describing the Schrödinger equation: $\phi(t+2) = M1M2\phi(t)$.

$$M_1 = \begin{pmatrix} b & a & 0 & 0 & ... & 0 & 0 & 0 & 0 \\ a & b & 0 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & b & a & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & a & b & ... & 0 & 0 & 0 & 0 \\ & & & & ... & & & & \\ 0 & 0 & 0 & 0 & ... & b & a & 0 & 0 \\ 0 & 0 & 0 & 0 & ... & a & b & 0 & 0 \\ 0 & 0 & 0 & 0 & ... & 0 & 0 & b & a \\ 0 & 0 & 0 & 0 & ... & 0 & 0 & a & b \end{pmatrix} \qquad M_2 = \begin{pmatrix} b & 0 & 0 & 0 & ... & 0 & 0 & 0 & a \\ 0 & b & a & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & a & b & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b & ... & 0 & 0 & 0 & 0 \\ & & & & ... & & & & \\ 0 & 0 & 0 & 0 & ... & b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & ... & 0 & b & a & 0 \\ 0 & 0 & 0 & 0 & ... & 0 & a & b & 0 \\ a & 0 & 0 & 0 & ... & 0 & 0 & 0 & b \end{pmatrix}$$

With a and b complex numbers such that $|a|^2 + |b|^2 = 1andab^* + ba^* = 0$ ($a*$ is the conjugate of a).

It has been proven that $m = i\frac{b}{a}$ a where m is the mass of the particle.

Using the equation described before allows us to act on the particles by pairs using the matrix S described as follow:[21]

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & b & a & 0 \\ 0 & a & b & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

It is important to note that the conditions on a and b assures that the matrix S is unitary which is a necessary condition for it to be a quantum gate. The algorithm using the matrix S can be represented through this figure. Indeed, on each iteration, only two adjacent points interact with each other.[21]

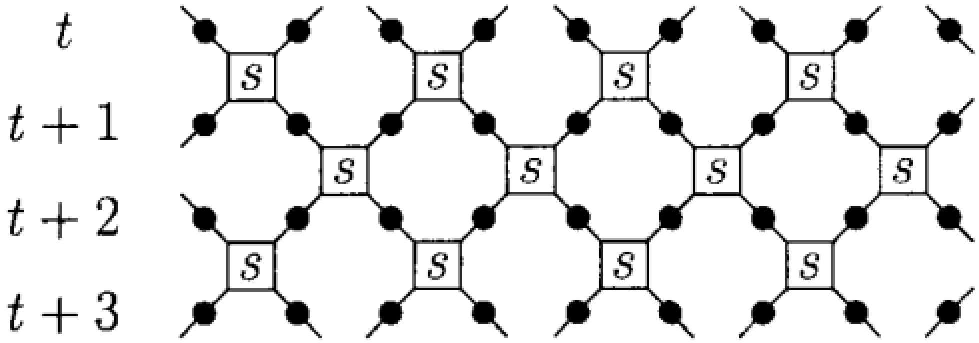


Figure 5.24: Representation of the update of the one-dimensional lattice through time

We now have an algorithm for simulating a free quantum particle in one dimension. The dynamics of this system, which are described by the dispersion relation , are essentially trivial. We shall now add various features to the model to give a system of more physical interest. First, we describe how an external potential can be easily incorporated into the system.

Given an external potential function U(x), the effect of this potential can be brought into the dynamics by acting on each q-bit at each time step with the matrix[21]

$$U_x = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{-\mathrm{i}\epsilon^2 U(x)} \end{pmatrix},$$

5.3.1 Qiskit

Qiskit [kiss-kit] is an open-source SDK for working with quantum computers at the level of pulses, circuits, and application modules.

It provides tools for creating and manipulating quantum programs and running them on prototype quantum devices on IBM Quantum Experience or on simulators on a local computer. It follows the circuit model for universal quantum computation, and can be used for any quantum hardware (currently supports superconducting qubits and trapped ions[4]) that follows this model [22]

Components:

Qiskit is made up of elements that work together to enable quantum computing. The central goal of Qiskit is to build a software stack that makes it easy for anyone to use quantum computers, regardless of their skill level or area of interest; Qiskit allows users to easily design experiments and applications and run them on real quantum computers and/or classical simulators. Qiskit provides the ability to develop quantum software both at the machine code level of OpenQASM, and at abstract levels suitable for end-users without quantum computing expertise. [22]

Installation:

To install Qiskit locally, you will need Python 3.7+. Although it is prefered to use virtual environment with Anaconda and jupyter notebook

5.3.2 IBM cloud quantum computer

In 2016, IBM connected a small quantum computer to the cloud and it allows for simple programs to be built and executed on the cloud.[1] In early 2017, researchers from Rigetti Computing demonstrated the first programmable cloud access using the pyQuil Python library.[2] Many people from academic researchers and professors to schoolkids, have already built programs that run many different quantum algorithms using the program tools. Some consumers hoped to use the fast computing to model financial markets or to build more advanced AI systems. These use methods allow people outside a professional lab or institution to experience and learn more about such a phenomenal technology[23]

5.3.3 Process

The first step is to find a set of basic quantum gates equivalent to the matrix S mentioned earlier.

Then, we built the quantum circuit out of the S gate, the $R_\phi with\phi = \epsilon^2 U(x) isthe$ position in the lattice) and the CNOT as a copy-gate. Indeed, CNOT can copy the state of a qubit when the control bit is the one to copy, and the target is $|0>$. This helped to implement the S gate twice to the same qubit.

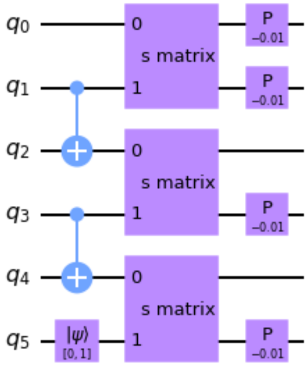


Figure 5.25: Quantum Circuit representing the S matrix

[22]

This step was repeated n times for the results to be maintained. [22]

Finally, we chose different positions for the particle and started the simulation.

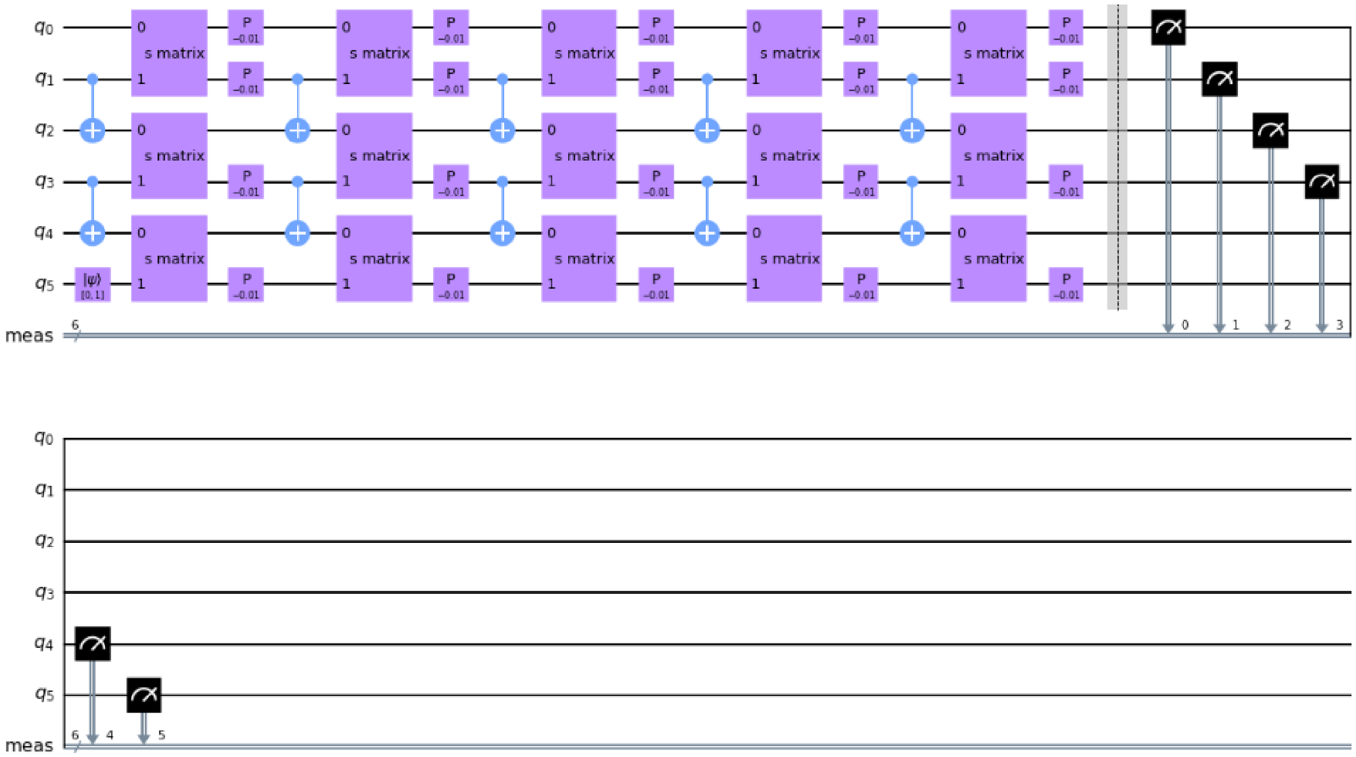


Figure 5.26: Schrodinger 's equation simulation after 4 iterations

5.3.4 Results and discussion

We choose lattice of 6 qubits and initialise the position at the last qubit[22] [24]

Particle initialised at $|1>$ and other positions are $|0>$,phase=-0.01

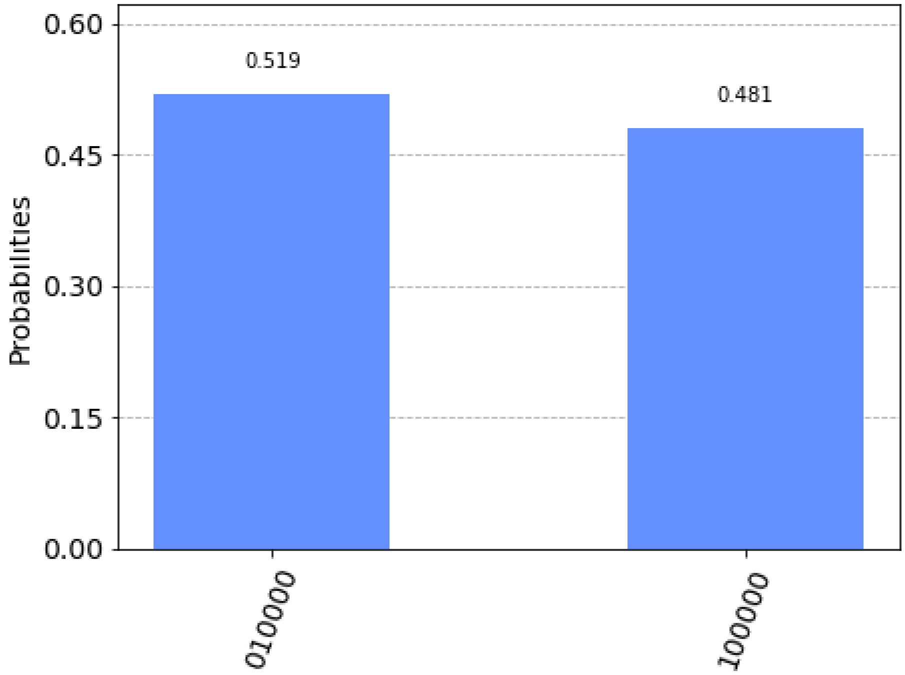


Figure 5.27: After 5 iterations

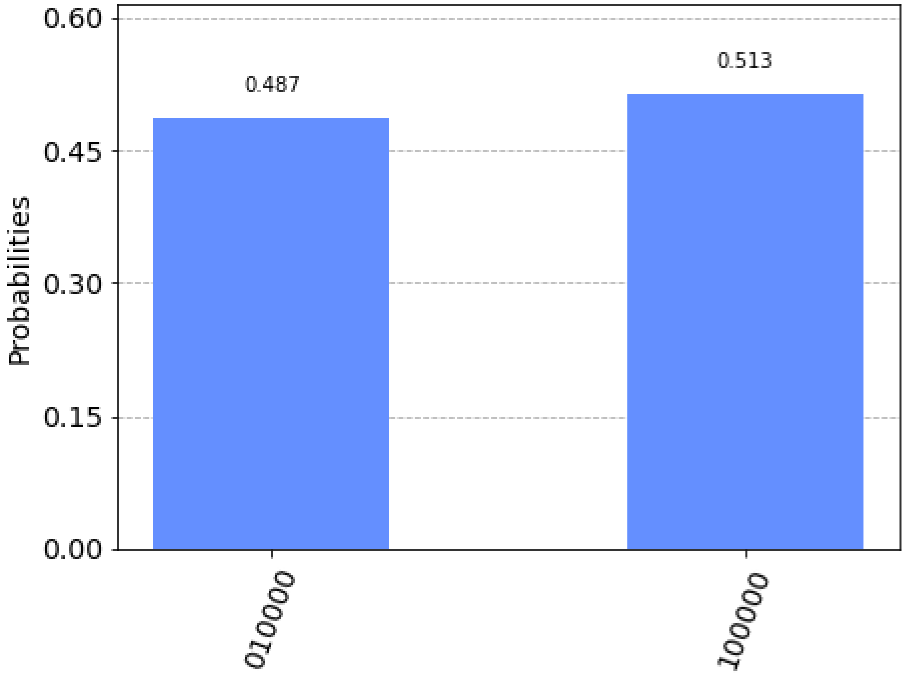


Figure 5.28: After 50 iterations

Discussion

We see that for small phase the particle qubit jumps only to the nearest qubits (adjacent lattice point) even with incresing the number of iterations

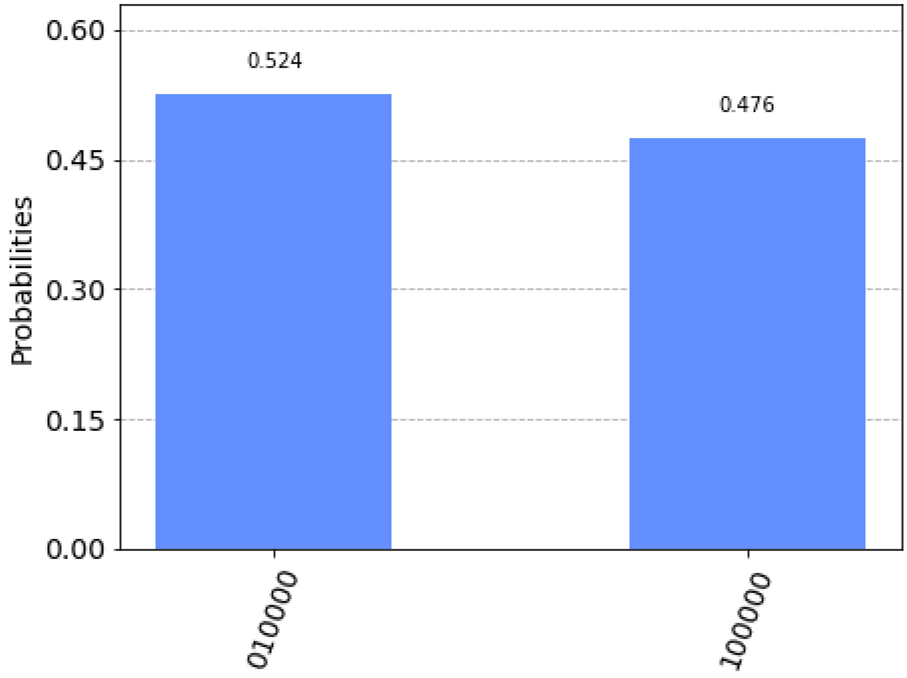


Figure 5.29: After 150 iterations

with a half half probability (superposition of the initial position and adjacent qubit)

Particle initialised at |1> and other positions are |0> ,phase=-5

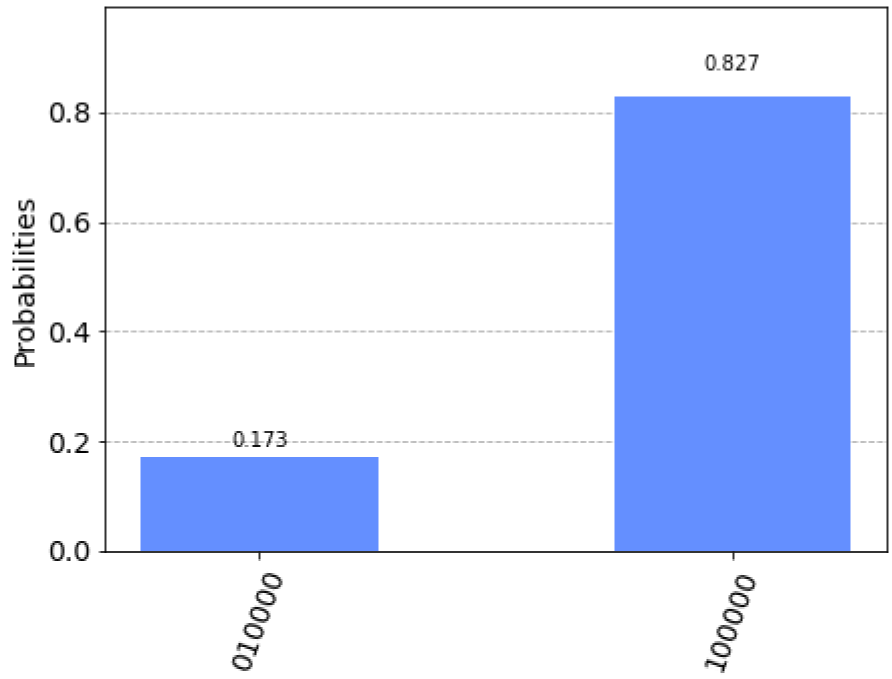


Figure 5.30: After 5 iterations

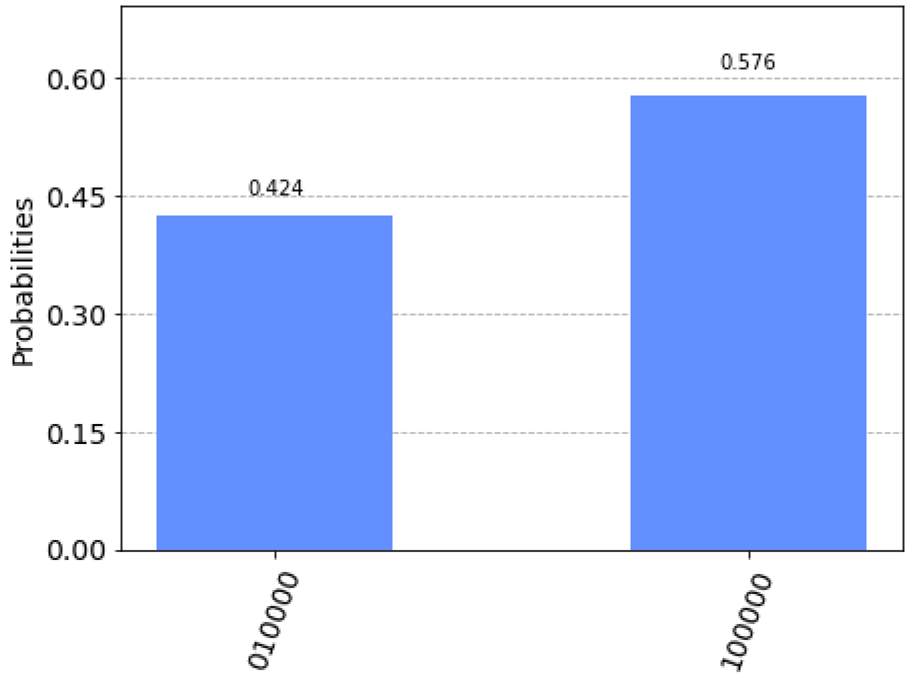


Figure 5.31: After 50 iterations

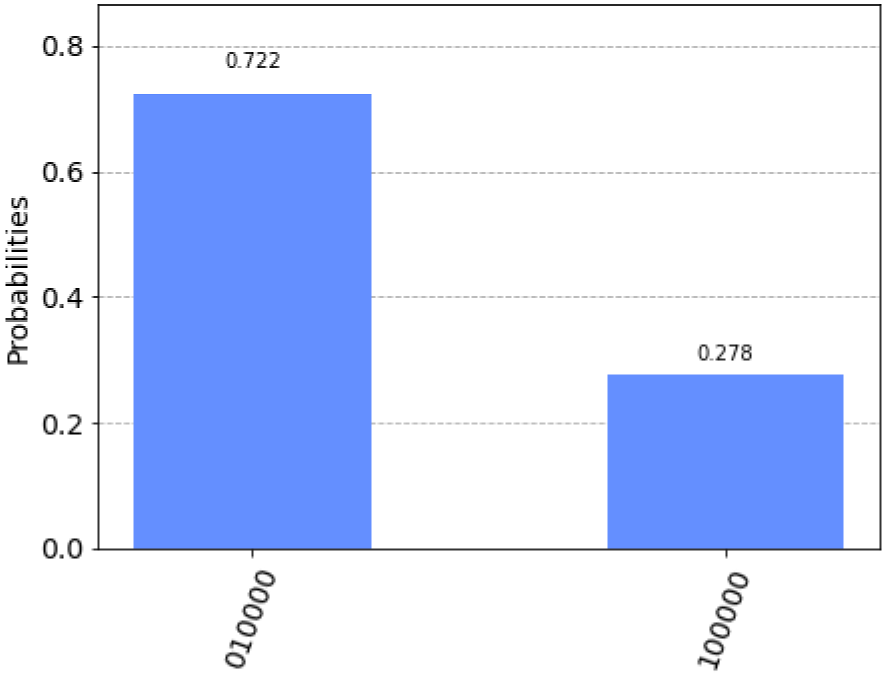


Figure 5.32: After 150 iterations

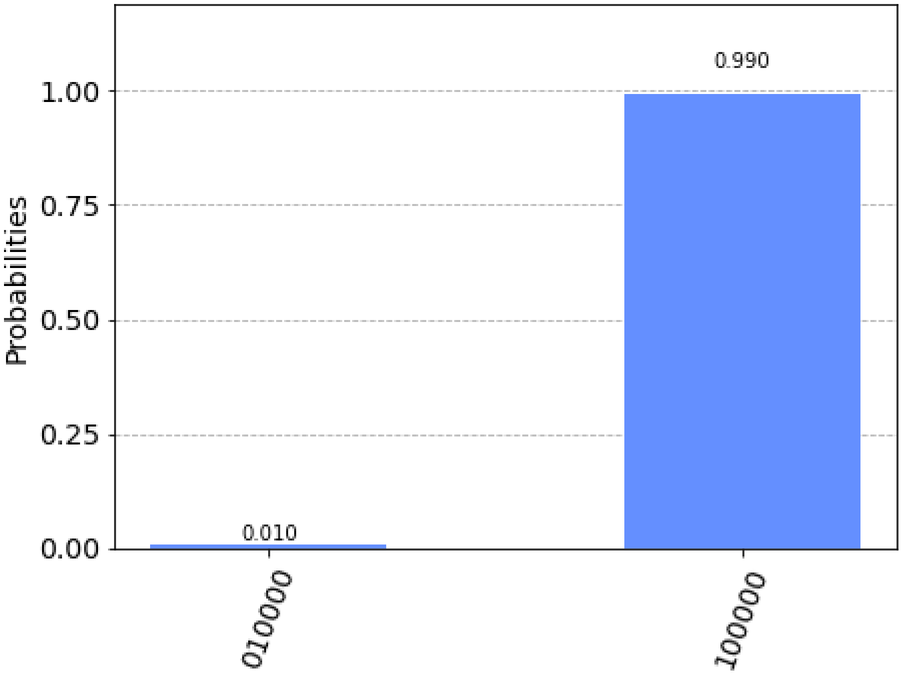


Figure 5.33: After 200 iterations

Discussion

A big value of external potential phase effects the probability every time P gate is introduced which is shown above

Particle initialised at |1> and other positions are |0> ,phase=-0.01 but with different a and b values (not equal modules)

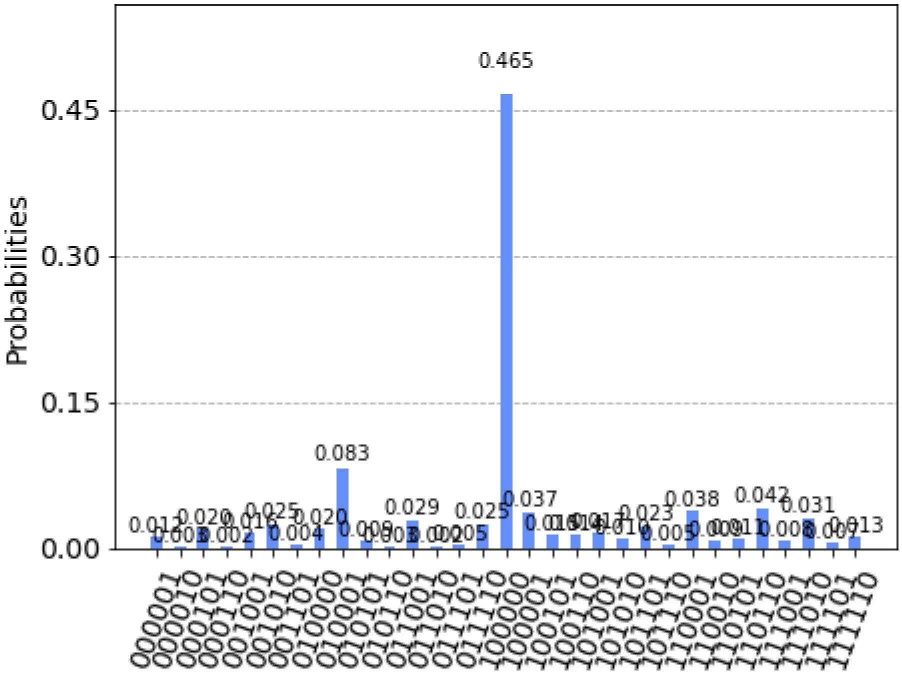


Figure 5.34: After 5 iterations

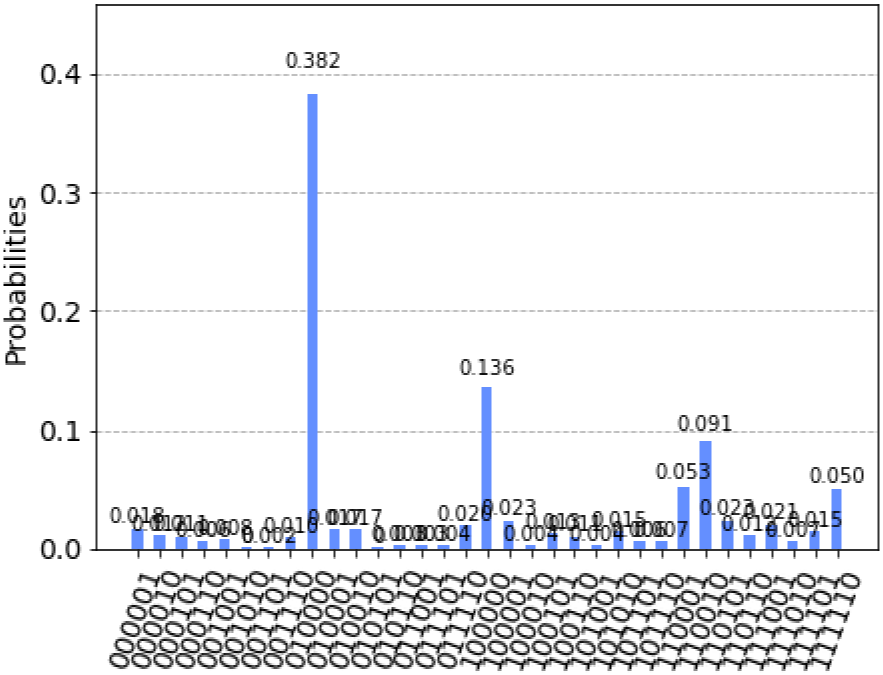


Figure 5.35: After 50 iterations

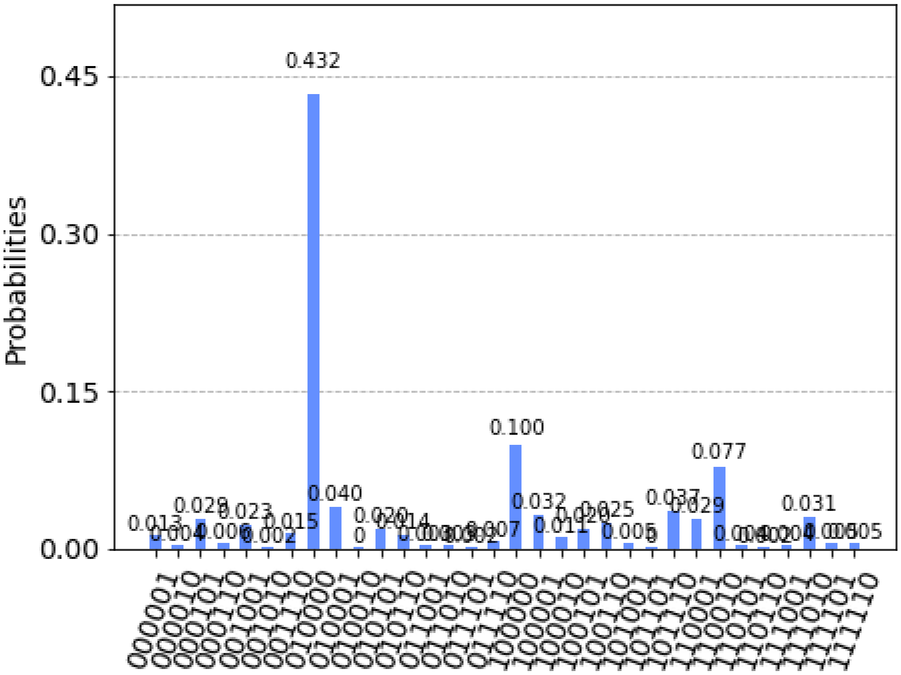


Figure 5.36: After 150 iterations

Discussion

Different a and b value means different wave function behavior which implies different density function ,with iteration's increasing the probability density will collapse into one point which is shown above

5.4 Conclusion

In this chapter we applied the Quantum behavior to a well known optimization algorithm "Particle Swarm" . The scope of this approach was to increase efficiency of PV module. Photovoltaic cell characteristic is non-linear whose output power varies as irradiance and temperature varies.A DC-DC Boost converter which steps up the voltage from the output of solar panel which is 16-21v to a voltage required to power the device. Then a DC-DC Boost convertor is Designed to increase the output voltage, suitable inductor and capacitance values were found. A DC-DC Boost converter modelling is done to check stability of the circuit.

The efficiency of the PV module is increased by using a Quantum particle swarm optimization(QPSO) assisted MPPT algorithm. Quantum Particle Swarm Optimization technique is implemented and increase in output power is demonstrated . This method is tested for varying atmospheric conditions. QPSO method was found to track MPP faster and accurately even under varying atmospheric conditions.

In addition QPSO is generally have better advantages of the PSO and eliminate the disadvantages either ,QPSO is faster ,more accurate and most importantly Robust under the varying conditions ,adjusting the expansion factor will attain a far better results

The results of the quantum simulation of quantum particle on IBM cloud computer were logic and have a meaningful interpretation of The effect of external potential and the collapse of the density probability function of wave function into one point after many iterations

Chapter 6

Future work

The approaches used through out this work has without doubt clarified that Quantum thinking is needed to solve world wide problems ,especially Energy.Quantum behaved method has proven it's efficiency but not the best yet ,nonetheless we only used the delta potential variant which can leave to apply the other two variants as an option .Also research is still at it's beginning to transform all the famous MPPT methods to a quantum behaved .

For Quantum Computing research is still going on with hardware problems and simulation of real dynamic algorithms if Quantum Differentiation can be achieved and implemented we can directly attack the incremental conductance with it ,Quantum machine learning can also be a huge impact on the energy domain with penny-lane 's contributions and many others .Theoretical results will comes to life

Chapter 7

General Conclusion

The ultimate goal of this research is to contribute to the optimization of systems photovoltaic, The goal of the maximum power point tracking algorithm (MPPT) is to optimize the operating point of the photovoltaic system to the MPP point of the I-V curve where the module produces its maximum power.However, reaching this objective is a demanding task because the MPP on the P-V characteristic curve is unstable in due to the continuous variation of solar radiation and temperature.

In this Thesis a BUCK-BOOST converter was used in the optimization of a photovoltaic system. This power converter is designed to operate in continuous conduction current. The buck-boost topology is used for several reasons, in particular because it has superior characteristics regarding the performance of the generator MPP photovoltaic, and it allows the continuation of the MPP at any time, whatever the PV panel temperature, solar radiance and connected load.

After the simulation result we conclude that the quantum behaved particle swarm optimization gives high accuracy and good robustness to the varying atmosphere from temperature and partial shading ,also it is more faster and precise than the classical Particle Swarm Optimization.

Adjusting only the expansion factor through the iteration number and adapting it to Monte Carlo stochastic position measurement will make sure to obtain better results

Regarding the simulation on quantum computer ,we only simulated a one free particle with different functions and due to lack of research until this right moment.The results were logical and probabilities eventually collapses to a single position

Bibliography

[1] M. A. Nielsen and I. Chuang, “Quantum computation and quantum information,” 2002.

[2] O. Arthur *et al.*, “An introduction to quantum computing algorithms,” 2022.

[3] R. K. Brylinski and G. Chen, *Mathematics of quantum computation.* CRC Press, 2002.

[4] W. Scherer, *Mathematics of quantum computing.* Springer, 2019.

[5] C. dillemegani, “Quantum hardware components, interfaces challenges.”

[6] M. M. Rahman, M. A. Islam, A. Z. Karim, and A. H. Ronee, “Effects of natural dust on the performance of pv panels in bangladesh,” *International Journal of Modern Education and Computer Science*, vol. 4, no. 10, p. 26, 2012.

[7] M. J. Adinoyi and S. A. Said, “Effect of dust accumulation on the power outputs of solar photovoltaic modules,” *Renewable energy*, vol. 60, pp. 633–636, 2013.

[8] M. Chegaar, A. Hamzaoui, A. Namoda, P. Petit, M. Aillerie, and A. Herguth, “Effect of illumination intensity on solar cells parameters,” *Energy Procedia*, vol. 36, pp. 722–729, 2013.

[9] P. Murty, *Power systems analysis.* butterworth-heinemann, 2017.

[10] F. A. Lindholm, J. G. Fossum, and E. L. Burgess, “Application of the superposition principle to solar-cell analysis,” *IEEE transactions on electron devices*, vol. 26, no. 3, pp. 165–171, 1979.

[11] C. W. Priananda, R. Sulistyowati, and R. Sulistyowati, “Analisis dan simulasi metode hill climbing untuk maximum power point tracker (mppt) pada photovoltaic statis,” *Jurusan Teknik Elektro, Institut Teknologi Adhi Tama Surabaya, Surabaya*, 2015.

[12] H. Abbes, H. Abid, K. Loukil, A. Toumi, and M. Abid, “Etude comparative de cinq algorithmes de commande mppt pour un système photovoltaïque,” *Journal of Renewable Energies*, vol. 17, no. 3, pp. 435–445, 2014.

[13] L. L. Jiang, D. L. Maskell, and J. C. Patra, “A novel ant colony optimization-based maximum power point tracking for photovoltaic systems under partially shaded conditions,” *Energy and Buildings*, vol. 58, pp. 227–236, 2013.

[14] M. Miyatake, M. Veerachary, F. Toriumi, N. Fujii, and H. Ko, “Maximum power point tracking of multiple photovoltaic arrays: A pso approach,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 367–380, 2011.

[15] A. soufyane Benyoucef, A. Chouder, K. Kara, S. Silvestre, *et al.*, “Artificial bee colony based algorithm for maximum power point tracking (mppt) for pv systems operating under partial shaded conditions,” *Applied Soft Computing*, vol. 32, pp. 38–48, 2015.

[16] J. Ahmed and Z. Salam, “A maximum power point tracking (mppt) for pv system using cuckoo search with partial shading capability,” *Applied energy*, vol. 119, pp. 118–130, 2014.

[17] S. Binitha, S. S. Sathya, *et al.*, “A survey of bio inspired optimization algorithms,” *International journal of soft computing and engineering*, vol. 2, no. 2, pp. 137–151, 2012.

[18] H. Patidar, G. K. Mahanti, and R. Muralidharan, “Synthesis of non-uniformly spaced linear array of unequal length parallel dipole antennas for impedance matching using qpso,” *International Journal of Microwave and Optical Technology*, vol. 12, no. 3, pp. 172–181, 2017.

[19] Y. Peng, Y. Xiang, and Y. Zhong, “Quantum-behaved particle swarm optimization algorithm with lévy mutated global best position,” in *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 529–534, IEEE, 2013.

[20] X.-S. Yang and X. He, “Nature-inspired optimization algorithms in engineering: overview and applications,” *Nature-inspired computation in engineering*, pp. 1–20, 2016.

[21] B. M. Boghosian and W. Taylor IV, “Simulating quantum mechanics on a quantum computer,” *Physica D: Nonlinear Phenomena*, vol. 120, no. 1-2, pp. 30–42, 1998.

[22] “Qiskit.” `https://qiskit.org`.

[23] “Ibm quantum computing.” `https://quantum-computing.ibm.com`.

[24] “Ibm cloud quantum computer.” `https://cloud.ibm.com/quantum`.