Summer 2015

Quantum inspired algorithms for learning and control of stochastic systems

Karthikeyan Rajagopal

Department: Mechanical and Aerospace Engineering

QUANTUM INSPIRED ALGORITHMS FOR LEARNING AND CONTROL OF STOCHASTIC SYSTEMS

by

KARTHIKEYAN RAJAGOPAL

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

AEROSPACE ENGINEERING

2015

Approved by
Dr. S.N. Balakrishnan, Advisor
Dr. Jerome Busemeyer
Dr. Jagannathan Sarangapani
Dr. Robert G. Landers
Dr. Ming Leu

ABSTRACT

Motivated by the limitations of the current reinforcement learning and optimal control techniques, this dissertation proposes quantum theory inspired algorithms for learning and control of both single-agent and multi-agent stochastic systems.

A common problem encountered in traditional reinforcement learning techniques is the exploration-exploitation trade-off. To address the above issue an action selection procedure inspired by a quantum search algorithm called Grover's iteration is developed. This procedure does not require an explicit design parameter to specify the relative frequency of explorative/exploitative actions.

The second part of this dissertation extends the powerful adaptive critic design methodology to solve finite horizon stochastic optimal control problems. To numerically solve the stochastic Hamilton Jacobi Bellman equation, which characterizes the optimal expected cost function, large number of trajectory samples are required. The proposed methodology overcomes the above difficulty by using the path integral control formulation to adaptively sample trajectories of importance.

The third part of this dissertation presents two quantum inspired coordination models to dynamically assign targets to agents operating in a stochastic environment. The first approach uses a quantum decision theory model that explains irrational action choices in human decision making. The second approach uses a quantum game theory model that exploits the quantum mechanical phenomena "entanglement" to increase individual pay-off in multi-player games. The efficiency and scalability of the proposed coordination models are demonstrated through simulations of a large scale multi-agent system.

ACKNOWLEDGMENTS

I express my deep gratitude and respect to my research advisor Dr. S. N. Balakrishnan. He inspired me to become an independent researcher and gave me a wonderful opportunity to work on an unconventional research topic. I also thank him for the guidance and the financial support he provided me over the course of my study.

My sincere thanks to Dr. Jerome Busemeyer, Indiana University for being part of my dissertation committee and for introducing me to the fascinating aspects of the human decision making. I am most grateful to my dissertation advisory committee members: Dr. Robert G. Landers, Dr. Jagannathan Sarangapani, and Dr. Ming Leu for being part of my committee and providing me valuable suggestions on my research work.

I deeply thank my parents for their blessings and support all this time. It was their unconditional love that kept me going and helped me get through some of the agonizing periods in a positive way. Finally, I thank all my friends especially Dr. Magesh Thiruvengadam, Manoj Kumar, Muthukumaran Loganathan for some insightful discussions, not only about my research topic but also about the philosophy of life. They made my stay in Rolla memorable.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

1. INTRODUCTION

1.1. MOTIVATION

Optimal control theory is an established mathematical framework with numerous applications in science and engineering [1-3]. However, very few optimal control problems have analytical solutions. The growth of computers has enabled the use of numerical techniques to solve some of the challenging optimal control problems. It also has fuelled the evolution of intelligent autonomous systems capable of learning and performing tasks in any unstructured or stochastic environment with a high degree of autonomy. The idea of utilizing multiple autonomous systems to complete complex tasks has also garnered immense interest. Although efficient numerical techniques have been discovered, the state space dimension of these autonomous systems can prohibit the use of such numerical techniques. Furthermore, even finding comprehensive optimal control solutions for single agent operating in stochastic environments is still a computationally hard problem. In case of multi-agent systems this is a much harder problem. To reduce the computational complexity, they typically need to operate with limited information. Moreover, a general dynamic model of multi-agent interactions is hard to build. Hence, a new numerically efficient design philosophy is required to address the above issues.

Famous physicist Richard Feynman first envisaged that a quantum computer can efficiently simulate the quantum mechanical effects compared to a classical computer. His idea inspired many researchers and a new branch of computer science called quantum computation was born. The main focus of this research field is to find efficient algorithms that uses the quantum mechanical effects to solve challenging classical problems. Over the years, lot of success has been achieved by using the quantum superposition effect intelligently for building efficient quantum algorithms. Some famous algorithms are: Shor's factoring algorithm [4], Grover's search algorithm [5-6] etc. However, the success of these algorithms entirely depends on the availability of quantum computers.

The mathematics of quantum theory has find applications in other branches of science also. Quantum cognition [7] is a branch of cognition science which applies the mathematical formalism of quantum theory to model various human cognitive phenomena. The governing belief is that humans are highly sensitive to context, order effects and the measurement disturbance. The human cognition models based on classic probability cannot accurately represent these complex phenomena. However, models based on quantum theory are more general and can efficiently represent the above complex phenomena. Furthermore, humans are very efficient in finding reasonable solution in limited time for certain complex problems (Examples: playing a game of chess, driving a car etc.), even with limited computation and information. There has been attempts to build artificial intelligence that can efficiently mimic this human intelligence. Quantum way of processing the available information might help us build better models of human intelligence [15].

Motivated by the above factors this dissertation proposes a quantum inspired action selection mechanism to improve the performance of traditional reinforcement learning techniques. It also proposes a quantum inspired coordination mechanism to reduce the computational complexity encountered in multi-agent dynamic task allocation problems. Another important contribution of this dissertation is to provide a path integral based adaptive critic solution suitable for stochastic optimal controller design.

Since ideas and notations from quantum theory are freely used in this dissertation a brief introduction to the subject is given in the next section.

1.2. QUANTUM MECHANICS

Classical mechanics [8] describes the motions of macroscopic objects; however, as the size of the object becomes sufficiently small, its laws fails to hold good. Quantum mechanics [9-11] was specifically developed to accurately predict the behaviors of microscopic particles. Some of the microscopic phenomena that the classical mechanics cannot account for are:

The wave-particle duality of the matter: The famous double split experiment demonstrated that the fundamental matter can behave both as a wave and as a particle. In a basic version of this experiment, light emitted from a coherent source of light was used to illuminate a plate with two parallel slits. A screen was placed behind the plate to observe the light passing through the slits. Interference patterns were observed on the screen when both the slits were open which indicated that the light behaves as a wave. However, when detectors were placed in the slit, they detected one photon passing through one slit. To account for this puzzling phenomena, in quantum mechanics the state of the microscopic particle is associated with a wave function. The interference patterns can now be explained through the interference of these wave functions. Then the detection of the light particles is explained through the collapse of the wave function during the measurement process. The square of the wave function gives the probability of detecting the light particle at a particular position.

Quantum superposition: It refers to the quantum mechanical property of a particle to exist in all possible states simultaneously. This quantum superposition state is quite different from the concept of mixed states defined in both classical and quantum mechanics. Mixed states are merely a statistical ensemble of pure states. However, superposition states are actual states of the quantum mechanical system that are formed by the superposition of pure states. They can cause observable effects. One example is the interference effect observed in the double slit experiment.

Quantum entanglement: The concept of entanglement in quantum mechanics describes the unintuitive behavior of two quantum particles prepared in a special quantum state. When these quantum particles are separated spatially and their spins are measured, the results obtained indicate that the spins of the particles are anti-correlated. This non-local behavior of entangled particles cannot be explained by classical mechanics.

1.2.1. Postulates of Quantum Mechanics. The basic mathematical framework of quantum mechanics is summarized using these postulates [9]:

Postulate I: The dynamical state of a quantum mechanical system at every time instant is completely described by a state vector of unit norm in Hilbert space. In Dirac's Ket notation it is represented by $|\psi\rangle$. Hence, the possible state vectors of a quantum mechanical system are the elements of a complex Hilbert space. The quantum state can also be completely described by a mathematical object called wave function. For example, the square modulus of the position wave function denoted by $|\psi(x,t)|^2$ can be interpreted as the probability density that the particle is at position x.

Postulate II: The continuous time evolution of a quantum mechanical system is *deterministically* described by the Schrodinger wave equation:

$$i\hbar\frac{\partial}{\partial t}\psi(x,t)=\frac{-\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2}+V(x,t)\psi(x,t) \tag{1}$$

Here, m is the particle's mass, V is its potential energy function and $\hbar$ is the reduced Planck's constant. Moreover, the states of a quantum mechanical system at two different time instants are related by a unitary transformation

Postulate III: Every measurable quantity/observable of a quantum mechanical system is associated with a linear Hermitian operator. The numerical outcome of the measurement process is given by the eigenvalues of these Hermitian operators.

Postulate IV: The state-space of the composite system is the tensor product of the component systems.

1.2.2. An Example to Illustrate Quantum Theory. A simplest quantum mechanical system is a qubit. It is very similar to the classical bit. The difference between a classical bit and a qubit is that, the qubit can be in a state other than 0 or 1. Thus, the state-space of the qubit is a two-dimensional Hilbert space. In Dirac notation, the orthonormal basis of this state space are denoted as $|0\rangle$ and $|1\rangle$. Let the qubit be in one of the superposition state:

$$|\psi(t)\rangle=\alpha|0\rangle+\beta|1\rangle \tag{2}$$

at time t where α and β are complex numbers such that $|\alpha|^2+|\beta|^2=1$. The terms α and β are called probability amplitudes. Suppose a measurement is performed, then the probability of obtaining $|0\rangle$ as outcome is $|\alpha|^2$ and the probability of obtaining $|1\rangle$ as outcome is $|\beta|^2$; also the qubit will collapse from the superposition state to one of the basis states. Let us assume that the qubit given in Eq. (2) is subjected to a time evolution in which the basis states are flipped i.e. 0 to 1 and 1 to 0 then this time evolution can be represented by a unitary transformation given by

$$\left|\psi\left(t+\tau\right)\right\rangle = U\left|\psi\left(t\right)\right\rangle \tag{3}$$

where

$$U = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ \left|\psi\left(t+\tau\right)\right\rangle = \beta\left|0\right\rangle + \alpha\left|1\right\rangle \tag{4}$$

Suppose a measurement is performed after the quantum evolution, then the probability of obtaining $\left|1\right\rangle$ as outcome is $\left|\alpha\right|^2$ and the probability of obtaining $\left|0\right\rangle$ as outcome is $\left|\beta\right|^2$ .Consider two qubits with states $\left|\psi_1\right\rangle$ and $\left|\psi_2\right\rangle$ such that

$$\begin{aligned} \left|\psi_1\right\rangle &= \alpha_1\left|0\right\rangle + \beta_1\left|1\right\rangle \\ \left|\psi_2\right\rangle &= \alpha_2\left|0\right\rangle + \beta_2\left|1\right\rangle \end{aligned} \tag{5}$$

According to *postulate IV*, the composite state of the combined system is given by

$$\left|\psi\right\rangle = \left|\psi_1\right\rangle \otimes \left|\psi_2\right\rangle = \alpha_1\alpha_2\left|00\right\rangle + \alpha_1\beta_2\left|01\right\rangle + \beta_1\alpha_2\left|10\right\rangle + \beta_1\beta_2\left|11\right\rangle \tag{6}$$

Here, $\left|01\right\rangle$ represents that the first qubit is in state $\left|0\right\rangle$ and the second qubit is in state $\left|1\right\rangle$. In similar vein, other notations $\left|00\right\rangle$, $\left|10\right\rangle$ and $\left|11\right\rangle$ also can be interpreted. There are some distinct states in composite quantum systems such that, they cannot be represented using Eq. (6). These states are called entangled states. One such state is given below

$$\left|\psi\right\rangle = \alpha\left|00\right\rangle + \beta\left|11\right\rangle \tag{7}$$

Note that if both coefficients α and β are non-zero, it is not possible to reduce Eq. (7) to any of the states given in Eq.(6). The concept of entangled states was first discussed by Albert Einstein in 1935 [12], as a non-classical behavior of microscopic particles to show that quantum mechanical theory was incomplete. However, both theoretically [13] and experimentally [14] the existence of entanglement have been verified. This concludes the brief introduction to quantum mechanics.

1.2.3. Connection Between Classical Mechanics and Quantum Mechanics. The common theoretical framework that links both classical mechanics and quantum mechanics is the Hamilton-Jacobi theory [16]. Consider a particle moving in an external potential field V from the initial position $q(t_1)=q_1$ to final position $q(t_2)=q_2$. According to classical mechanics the particle will follow a trajectory that minimizes the following functional

$$J=\int_{t_1}^{t_2} L(t,q,\dot{q})\ dt \tag{8}$$

where q is the particle's position, $\dot{q}$ is the particle's velocity and L is the Lagrangian function defined by

$$L(q,\dot{q},t)=\frac{1}{2}\dot{q}^T q-V(q) \tag{9}$$

To solve the above problem using the Hamilton-Jacobi theory the Hamiltonian function is defined as following

$$H=p^T q-L\left(q,\dot{q},t\right) \tag{10}$$

where $p_i=\partial L/\partial\dot{q}_i$ is called the canonical momentum. Let $S=\min_q J$. S is called the Hamilton's principal function. It is also called as the least action function in Lagrangian mechanics. By Hamilton -Jacobi theory we can obtain the Hamilton-Jacobi equation:

$$\frac{\partial S}{\partial t} + H(q, p) = 0 \tag{11}$$

The canonical momentum p can also be written using the principal function as $p = \partial S / \partial q$. For a particle of mass m moving in a one dimensional potential field, the above HJ equation becomes

$$\frac{\partial S}{\partial t} + \frac{\partial S}{\partial q}\dot{q} - L(q, \dot{q}, t) = 0 \tag{12}$$

The solution S obtained from Hamilton-Jacobi equation is non-unique for a given mechanical problem. It is connected with an infinite set of potential trajectories pursued by an ensemble of identical particles. To get a unique solution both the initial position and velocity has to be defined. The Hamilton-Jacobi equation solution naturally provides us with an ensemble description of particle motion and provides a basis for classical statistical mechanics. In classical statistical mechanics the motion of a particle is deterministic but unpredictable because of lack of information. Suppose we know the initial probability distribution $\rho\left(q, t_1\right)$ for position of an ensemble of particles the evolution of this distribution is given by the Liouville's equation

$$\frac{\partial \rho}{\partial t} + \nabla . \left(\rho \frac{\nabla S}{m} \right) = 0 \tag{13}$$

In classical statistical mechanics the HJ equation and density equation are partially coupled we can solve the HJ equation without knowing the probability distribution but solving of the density equation requires knowledge about S. As mentioned earlier, in quantum mechanics the state of the system is completely defined by the wave function $\psi(x, t)$ and the evolution of this wave function is dictated by the Schrodinger equation given in Eq. (1).

Now to bring in the particle interpretation of quantum mechanics (Bohmian Mechanics [16-17]) the following form of the wave function is considered

$$\psi(x,t)=R(x,t)e^{-iS/\hbar} \tag{14}$$

Substituting the above form of wave equation in the Schrodinger equation and splitting them into real and imaginary parts will result in the quantum Hamilton Jacobi equation

$$\frac{\partial S}{\partial t}+\frac{\left(\nabla S\right)^2}{2m}-\frac{\hbar^2}{2m}\frac{\nabla^2 R}{R}+V=0 \tag{15}$$

and the density equation

$$\frac{\partial R^2}{\partial t}+\nabla.\left(\frac{R^2\nabla S}{m}\right)=0, R^2=\rho \tag{16}$$

Comparing the classical HJ equation and the quantum HJ equation we will see there is one extra term $\left(-\frac{\hbar^2}{2m}\frac{\nabla^2 R}{R}\right)$ in the quantum HJ equation which is called quantum potential and that is a major difference between classical and quantum mechanics. The density equation of quantum mechanics is of the same form as classical mechanics but unlike classical mechanics the HJ equation and density equation are fully coupled. Both of them have to be solved simultaneously. Interestingly the quantum potential term in the quantum HJ equation depends on the shape of the density function (i.e. the curvature $\nabla^2 R$). Thus the motion uncertainty in quantum mechanics is not due to lack of information, nevertheless a fundamental attribute of the microscopic particle.

There is a close connection between classical mechanics and deterministic optimal control theory [18]. In a general optimal control problem, given the dynamics of the system,

$$\dot{x}=f(x,u) \tag{17}$$

the objective is to find control u that minimizes the cost function.

$$J(x,t) = \int_{t_1}^{t_2} l(x,u)\ dt \tag{18}$$

Suppose $S(x,t) = \min\ J(x,t)$, then $S(x,t)$ is the solution of the HJB equation

$$\frac{\partial S}{\partial t} + \frac{\partial S}{\partial x} f(x,u) + l(x,u) = 0 \tag{19}$$

The HJB equation given in Eq. (19) becomes the classical HJ equation when $f(x,u) = u$ and $l(x,u) = -L(x,\dot{x})$. Thus the deterministic optimal control theory is simply a generalization of the classical mechanics. Similarly there were various attempts to link the stochastic processes and quantum mechanics [18-20]. However, much progress has not been made yet.

1.3. DISSERTATION OUTLINE

There are 5 Sections in this dissertation including the introductory Section. Section 2 presents a novel Quantum inspired reinforcement learning (QiRL) algorithm. In RL algorithms, agents learn by interacting with the environment. Hence, the agent has two options: either to pick an action based on already gained knowledge or to explore by choosing a random action. Ideally, the agent should try a variety of actions and progressively favor the action that maximizes its reward. Section 2.1 starts with a brief introduction to current action selection mechanisms available in literature to handle the exploration-exploitation trade-off. Then, the details of a fast quantum search algorithm called Grover’s iteration are discussed. The Grover’s iteration effectively uses the superposition property to reduce the search time when dealing with unsorted databases. Next, the concept of QiRL is introduced in Section 2.2. QiRL uses a generalized version of Grover’s iteration to select actions. Several simulation results are presented in Section 2.3 to demonstrate the effectiveness of the proposed algorithm.

In Section 3, an approximate dynamic programming approach using neural networks is proposed for solving a class of finite horizon stochastic optimal control problems. This Section starts with a brief introduction to existing literature on deterministic optimal control approaches. Then, the path integral formulation of the stochastic optimal control theory is discussed. For non-linear deterministic systems, Pontraygin Maximum principle based adaptive critic approaches provide a systematic way to synthesize the optimal control solution. However, in the presence of noise the maximum principle formulation becomes complex and very few methods exist to synthesize the optimal control solution. On the other hand, inclusion of noise in the Hamilton-Jacobi-Bellman (HJB) framework is very straight forward. For certain class of systems, the solution of stochastic HJB equation can be formulated as a path integral problem. The contribution of this dissertation is the development of an adaptive critic approach for synthesizing stochastic optimal controllers using path integrals. The developed adaptive critic algorithm is presented in Section 3.3 and the convergence analysis of this algorithm is performed in Section 3.4.

In Section 4, two quantum inspired coordination models are developed to dynamically assign targets to agents operating in a stochastic environment. Two ideas governed this development: a quantum decision theory model and a quantum game theory model. In Section 4.1, the principles of the above theories are discussed through examples. The quantum decision theory model explains the irrational behavior of humans as a result of entanglement between their actions and beliefs. On the other hand, quantum game theory model assumes that the game players have access to entangled quantum states and then finds quantum strategies that will maximizes the individual pay-off. In Section 4.2, the multi-agent problem considered for this study is presented and in Section 4.3 solution approaches are discussed. These quantum inspired coordination algorithms are scalable and efficient. In addition to the above algorithms, a classical game theory based coordination algorithm is also developed and its details are discussed in Section 4.4. Several simulation studies were conducted to analyze the performance of the developed approaches. In Section 4.6, the results obtained are presented and discussed.

In Section 5, conclusions are drawn.

2. QUANTUM INSPIRED REINFORCEMENT LEARNING

2.1. INTRODUCTION

In many real world automation problems, the software agents need to learn how to perform a task optimally. A general machine learning approach used for this purpose is supervised learning [21]. In supervised learning, agents are trained using examples prepared by an expert supervisor. However, there are many problems in which such external knowledge might not be available. In such cases, the agents need to figure out themselves the optimal way of performing a task. Reinforcement learning (RL) [22-23] is a trial and error approach in which an intelligent agent learns to take optimal actions by interacting with the environment. In RL, the goal of the agent is defined by a reward function. The agent receives a positive reinforcement (reward) whenever its actions result in a favorable outcome. Typically the RL agents start from an initial state and perform a series of action to reach a final state. The agent might receive immediate rewards for every action it performs or a delayed reward that depends only on the terminal state. The RL researchers widely use the Markov decision processes (MDPs) [24] framework to study sequential decision tasks. A MDP model contains a set of possible world states $\left(s_t \in S\right)$, a set of possible actions $\left(a_t \in A\right)$, a real valued reward function $R\left(S,\mathrm{A}\right)$ and a state transition probability descriptor $T\left(S,A\right)$. The solution of an MDP problem is the optimal mapping $\pi^* : S \to A$ between the states and actions that will maximize a long term expected reward. For a given policy π the long term expected reward is given by [22]

$$V^{\pi}\left(s_t\right) = E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^t R\left(s_{t+k}, s_{t+k+1}, a_{t+k}\right) \middle| s_t = s\right) \tag{20}$$

where V^{π} is also called the value function for policy π and $\gamma \in [0,1]$ is the discount factor.

Dynamic programming approaches are typically employed to solve MDP problems. However, they are computationally infeasible for large-scale problems. Furthermore, dynamic programming algorithms require full specification of the MDPs. However, RL algorithms can be used even when the full specification of the MDP is unavailable. One of the key ideas in both the dynamic programming approaches and RL algorithms is the Bellman optimality equation:

$$V^{*}(s) = \max_{a} \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^{*}(s')\right] \tag{21}$$

Here, $V^{*}(s) = \max_{\pi} V^{\pi}(s)$. The optimality equation reduces the infinite-stage optimization problem to a two-stage optimization problem. Another way of writing Eq. (21) is by using Q-values [25-26].

$$Q^{*}(s,a) = \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma \max_{a'} Q^{*}(s',a')\right] \tag{22}$$

where $V*(s) = \max_{a} Q^{*}(s,a)$. Unlike value functions V which are state dependent, Q-values are dependent on state-action pairs. This form of representation is very helpful for model-free learning. For example, if the transition probability $T(s,a,s')$ of the MDP is unknown, then the agents can directly interact with the environment and record their experience of choosing action a in state s by updating the Q-values. There are different RL techniques available to solve MDP when there is no explicit specification of transition probabilities. Temporal difference learning [22, 27] (TDL) is one such approach. TDL uses an estimate of the future reward to update the Q-values. A version of TDL called Sarsa on-policy is presented here.

$$Q^{i+1}\left(s_t,a_t\right)=Q^i\left(s_t,a_t\right)+\alpha\left(R\left(s_t,a_t,s_{t+1}\right)+\gamma Q^i\left(s_{t+1},a_{t+1}\right)-Q^i\left(s_t,a_t\right)\right) \quad (23)$$

In this technique, the future reward is approximated using $R\left(s_t,a_t,s_{t+1}\right)+\gamma Q^i\left(s_{t+1},a_{t+1}\right)$ and α is the learning rate. During the learning process, the agents need to choose actions based on their current knowledge. However, their current knowledge might not be complete. One of the challenges in reinforcement learning is the trade-off between exploration and exploitation [28]. The agent must try a variety of actions and progressively settle for those that maximizes their reward. Various action selection mechanisms are proposed in literature to resolve the exploration-exploitation trade-off. In [29], the agent maintains a complete but inaccurate model of its environment and optimal actions are chosen based on this model. A biology-inspired model-based RL scheme is proposed in [30] to control the balance between exploration and exploitation. In [31], an idea inspired by the Monte Carlo simulation literature called “probability of correct selection” is used to improve the action selection during exploratory phase. However, the most widely used approaches are

a. $\varepsilon-$greedy policy
b. Softmax approach

In ε -greedy policy [25], optimal actions are selected using acquired knowledge with probability $1-\varepsilon$ and new actions are explored with probability$1-\varepsilon$. However, there is no particular methodology exists for selecting the right value of ε and it is problem dependent. To counter the above issue, an adaptive ε-greedy policy that depends on the temporal-difference error is proposed in [32]. Furthermore, during exploration the ϵ-greedy policy equally weighs all possible actions. Hence, there is no difference between best action, second best action, and worst action and so on. In Softmax approach, a probability distribution is defined over the action set using the Q-values. The probability of selecting an action u is proportional to

$$\frac{e^{\frac{Q^i(s,u)}{T}}}{\sum_a e^{\frac{Q^i(s,a)}{T}}} \tag{24}$$

where T is a positive parameter called temperature. A high value of temperature parameter will make all the actions equally probable. For small value, the action with maximum Q-value is highly probable. In contrast to ε -greedy policy, Softmax approach ranks the actions according to the Q-values. The advantage of both the approaches is that the designer need to tune only one parameter to learn near-optimal control policies [33]. Designing ε and T parameter depends on the complexity of the environment and plays a critical role in convergence of search algorithms and the speed of convergence. In this Section, a reinforcement learning algorithm inspired by the quantum mechanical phenomena is proposed to mitigate the exploration-exploitation trade-off.

Quantum information processing is rapidly emerging field [34]. The basis of this field is the use of quantum mechanical phenomena like superposition and entanglement for processing of data. Exploiting these basic characteristics of quantum systems many quantum algorithms have emerged which can solve certain kind of difficult problems much faster than classical algorithm. For example Grover algorithm [5-6] can search an unsorted database of N entities for a particular data in $\sqrt{N}$ iterations compared to its classical counterpart which will take N iterations. Although these algorithms are primarily designed for quantum computers, they can be still simulated in traditional computers. Dong et.al [35-37] suggested using these quantum algorithms for improving the performance of traditional

RL method and they call this novel methodology as “Quantum Reinforcement Learning”. Inspired by the above idea, a more robust RL algorithm using a generalized version of the Grover algorithm is developed. A brief introduction to quantum theory concepts relevant to our discussion are presented here.

2.1.1. Grover Algorithm. Quantum theory allows the microscopic particles to exist in a superposition state at each moment in time. A definite state is realized only when the state of the particle is measured. This property of quantum particles is effectively utilized in the construction of Grover’s algorithm. Consider a function $f : A \to \{0,1\}$ where A is a set with N elements. Let each element of the set A has an index $x \in [0, N-1]$. However, there is only one element $s \in A$ such that $f(x) = 1$. Our objective is to search for that element. With a classical algorithm, the time that it will take to complete a search is $O(N)$. Finding a solution to the search problem is a hard task. However, recognizing a solution is much easier. Assume we can construct a device called oracle which can *recognize* the solution when it is presented to it. Then by superposition principle, in a quantum search, we can look at all possible solutions simultaneously. Assume the index x can be stored in n qubits and $N = 2^n$. Prepare the quantum mechanical system in the following super positional state

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \qquad (25)$$

Here, $\frac{1}{\sqrt{N}}$ is the amplitude of any index state $|x\rangle$. Then, the probability of selecting any state is given by $\left(\frac{1}{\sqrt{N}}\right)^2 = \frac{1}{N}$. Apply the oracle on $|\psi\rangle$. The oracle O is a unitary operator and it flips the sign of the solution element that we are searching for. The resulting quantum state is given by

$$|\psi_1\rangle = O|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{f(x)}|x\rangle \tag{26}$$

As mentioned earlier, during the measurement process the quantum mechanical system changes from the indefinite state given in Eq. (26) to a definite state. The result of the measurement process might be any one of the states $|x\rangle$, $x \in [0, N\text{ -}1]$. Hence, the sign change cannot be detected by any measurement. To tackle the above issue, Grover proposed an algorithm that maximizes the probability of getting the right answer. He proposed a unitary operator which is now called the Grover diffusion operator:

$$U = 2|\psi\rangle\langle\psi| - I_{N\times N} \tag{27}$$

Suppose x_0 is the index of the searched element, then Eq. (26) can be rewritten as

$$|\psi_1\rangle = \sqrt{\frac{N-1}{N}}|\alpha\rangle - \frac{1}{\sqrt{N}}|x_0\rangle \tag{28}$$

where

$$|\alpha\rangle = \frac{1}{\sqrt{N-1}}\sum_{x=0,x\neq x_0}^{N-1}|x\rangle \tag{29}$$

Then, the application of the Grover diffusion operator results in

$$|\psi_2\rangle = U|\psi_1\rangle = -\sqrt{\frac{N-1}{N}}\left(\frac{4-N}{N}\right)|\alpha\rangle + \frac{1}{\sqrt{N}}\left(\frac{3N-4}{N}\right)|x_0\rangle \tag{30}$$

If we take $\frac{1}{\sqrt{N}} = \sin\left(\frac{\theta_2}{2}\right)$, then

$$\sqrt{\frac{N-1}{N}} = \cos\left(\frac{\theta_2}{2}\right) \tag{31}$$

$$\frac{1}{\sqrt{N}}\left(\frac{3N-4}{N}\right)=\sin\left(\frac{\theta_2}{2}\right)\left(3-4\sin^2\left(\frac{\theta_2}{2}\right)\right)=\sin\left(\frac{3\theta_2}{2}\right) \tag{32}$$

$$-\sqrt{\frac{N-1}{N}}\left(\frac{4-N}{N}\right)=-\cos\left(\frac{\theta_2}{2}\right)\left(3-4\cos^2\left(\frac{\theta_2}{2}\right)\right)=\cos\left(\frac{3\theta_2}{2}\right) \tag{33}$$

Hence, Eq. (30) becomes

$$|\psi_2\rangle=\cos\left(\frac{3\theta_2}{2}\right)|\alpha\rangle+\sin\left(\frac{3\theta_2}{2}\right)|x_0\rangle \tag{34}$$

Furthermore, $|\psi\rangle$ also can be represented as

$$|\psi\rangle=\cos\left(\frac{\theta_2}{2}\right)|\alpha\rangle+\sin\left(\frac{\theta_2}{2}\right)|x_0\rangle \tag{35}$$

Comparing Eq. (34) and Eq. (35), as long as $\frac{3\theta_2}{2}\leq\frac{\pi}{2}$ the probability of measuring $|x_0\rangle$ increases. To further increase the likelihood of detecting $|x_0\rangle$ the oracle and Grover diffusion operator can be applied l times which results in

$$|\psi_2\rangle=(UO)^l|\psi\rangle=\cos\left(\frac{(2l+1)\theta_2}{2}\right)|\alpha\rangle+\sin\left(\frac{(2l+1)\theta_2}{2}\right)|x_0\rangle \tag{36}$$

However, one should be careful that $\frac{(2l+1)\theta_2}{2}$ should not exceed $\frac{\pi}{2}$. In quantum framework, the idea of increasing the probability is equivalent to boosting the amplitude. A generalization of boosting technique applied by Grover was proposed by Brassard and Hoyer [38-39]. Their idea is referred to as amplitude amplification. In their version, the operators O and U_s are defined as

$$O=I_{N\times N}-\left(1-e^{i\phi_1}\right)\left(|x_0\rangle\langle x_0|\right) \tag{37}$$

$$U=\left(1-e^{i\phi_2}\right)|\psi\rangle\langle\psi|-I_{N\times N} \tag{38}$$

Here, ϕ_1 and ϕ_2 are factors that control the amount of amplification. Hence, a generalized unitary operator for amplitude amplification is given by

$$G = \left(\left(\left(1-e^{i\phi_2}\right)|\psi\rangle\langle\psi| - I_{N\times N}\right)\left(I_{N\times N} - \left(1-e^{i\phi_1}\right)\left(|x_0\rangle\langle x_0|\right)\right)\right)^l = (UO)^l \tag{39}$$

Note that Eq. (39) will be equivalent to the Grover's searching algorithm when $\phi_1 = \phi_2 = \pi$ The RL algorithm proposed in this Section uses Eq. (39) for the action selection mechanism.

2.2. REINFORCEMENT LEARNING USING GROVER'S ALGORITHM

Quantum inspired Reinforcement Learning (QiRL) uses all the main concepts of traditional RL. It requires goal-oriented reward function, a Q-values update rule etc. However, action representation, action selection mechanism and policy updates are very different.

In QiRL, events are possible actions that agent could choose. The basic idea is that the current environmental state puts the agent in a superposition state over the set of possible actions. The superposition state is a vector in a m dimensional space spanned by m orthonormal basis vectors denoted $|a\rangle\,, k = 1, \ldots, m$ and each basis vector corresponds to one of the actions. If the current state is s, then the superposition state over actions is

$$|\psi_s\rangle = \sum_{k=1}^{m} \psi_{sk}\,|a_k\rangle \tag{40}$$

In this formula, ψ_{sk} indicates the amplitude of each action. If there is any action not available for state s, then $\psi_{sk} = 0$ for that particular action. The probability of taking action a_k in state s by definition equals to $|\psi_{sk}|^2$; thus amplitudes are related to probability by a nonlinear mapping. In QiRL formulation, initially the agent doesn't have any preferences among different actions; so it could equally weigh them. Assuming the agent is at the state s, one representation of superposition state in first episode might be the following:

$$\left|\psi_s\right\rangle = \sum_{k=1}^{4} \frac{1}{2}\left|a_k\right\rangle \tag{41}$$

This representation indicates that the agent has equally weighted all four possible actions. After a while when it learns more about the environment, this representation might be changed to:

$$\left|\psi_s\right\rangle = 0.0294\left|a_1\right\rangle + 0.4401\left|a_2\right\rangle + 0.0970\left|a_3\right\rangle + 0.8922\left|a_4\right\rangle \tag{42}$$

which demonstrates that the agent weighs less (close to zero) a_1 and a_3 and considers a_4 with a high probability of 80% and a_2 with probability of 19%. We will describe in details how these probabilities are updated in next section but the key new idea is the learning rule for modifying the amplitudes ψ_{sk}.

2.2.1. Amplitude Amplification. In this step, agent learns to adjust transition probabilities in each state using updating algorithms. In other words, the agent modifies its estimation of action-reward map in each state. In traditional RL techniques like TDL, only the Q-values are updated. However, in QiRL, the TD updating is followed by another updating rule, namely the Grover algorithm which amplifies actions' amplitude based on their Q-values. To increase the amplitude of action $\left|a_k\right\rangle$, using Eq. (37), the oracle is defined as

$$O = I_{N\times N} - \left(1 - e^{i\phi_1}\right)\left(\left|a_k\right\rangle\left\langle a_k\right|\right) \tag{43}$$

Then, using Eq. (39) we get

$$\left|\psi_s^i\right\rangle = (UO)^l \left|\psi_{\sup}\right\rangle \tag{44}$$

where

$$\left|\psi_{\sup}\right\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\left|a_k\right\rangle \tag{45}$$

Here, $\left|\psi_s^i\right\rangle$ is the superposition state over actions for state s at the end of iteration i. The novel idea of QiRL is to relate the parameters ϕ_1, ϕ_2 and l of the Grover algorithm to Q-values. There are two ways by which the above objective can be implemented:

1) By fixing l and varying ϕ_1 and ϕ_2
2) By fixing ϕ_1 & ϕ_2 and varying l

Dong et. al. [36], proposed a version of QiRL in which they fixed $\phi_1 = \phi_2 = \pi$ and varied l. The algorithm proposed here follows the first approach i.e. l is assigned the value of 1 and the parameters ϕ_1 and ϕ_2 are varied. For example, consider a problem in which an agent has four possible actions. If the first action is chosen for amplitude amplification, then its probability $\left|\psi_{s1}\right|^2$ computed using Eq. (44) is shown in Figure 2.1. Further, the probability of choosing any one of other three actions becomes $\frac{1-\left|\psi_{s1}\right|^2}{3}$. It can be observed that Eq. (44) exhibits a very nonlinear behavior. To simplify the amplification process, a parameterization of the following form is suggested

$$\begin{aligned} \phi_1 &= \pi(\phi + 3.7) \\ \phi_2 &= 0.15\phi_1 \end{aligned} \tag{46}$$

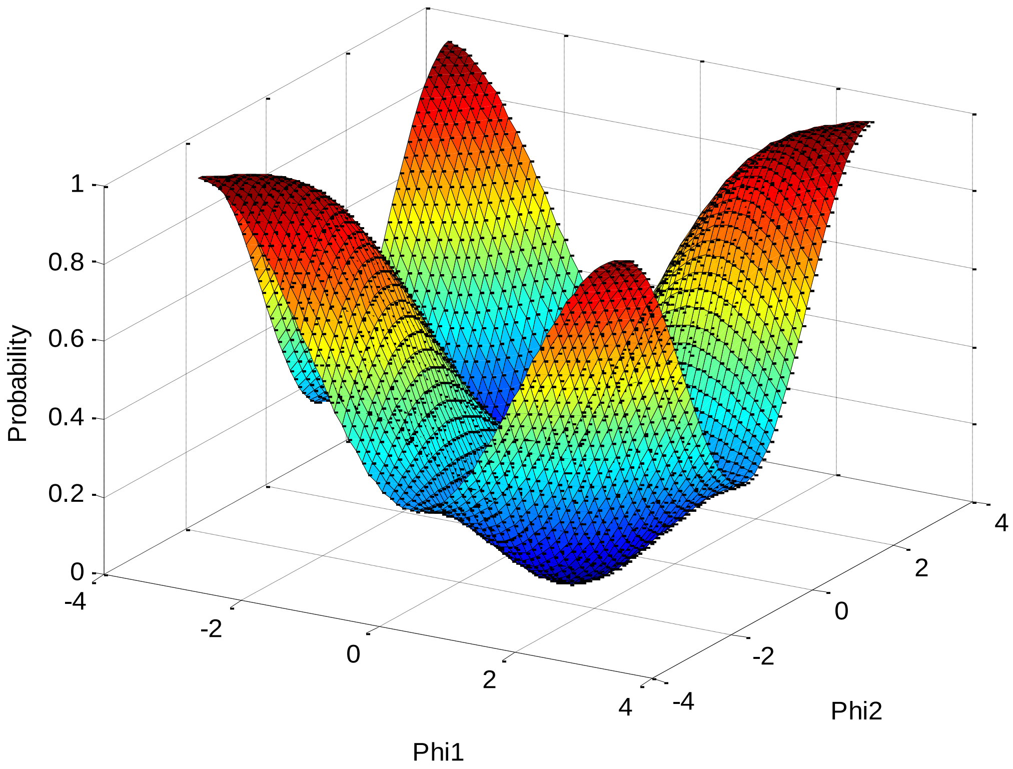


Figure 2.1 Probability variation with respect to changes in ϕ_1 and ϕ_2 (Agent has four possible actions)

where $\phi \in [0.3,1]$. Figure 2.2 shows the $|\psi_{s1}|^2$ variation with respect to ϕ. The parameter ϕ indicates the degree of goodness of the chosen action. Higher the value of ϕ, higher the amplification. For $\phi = 0.3$, all actions are equally weighed.

Based on the above discussions, a RL algorithm using amplitude amplification is outlined next. At each state, probability distribution for action selection is created as follows:

Step 1: Find the action $a_{\max}(s)$ corresponding to maximum Q-value in the current state s and select it for amplitude amplification.

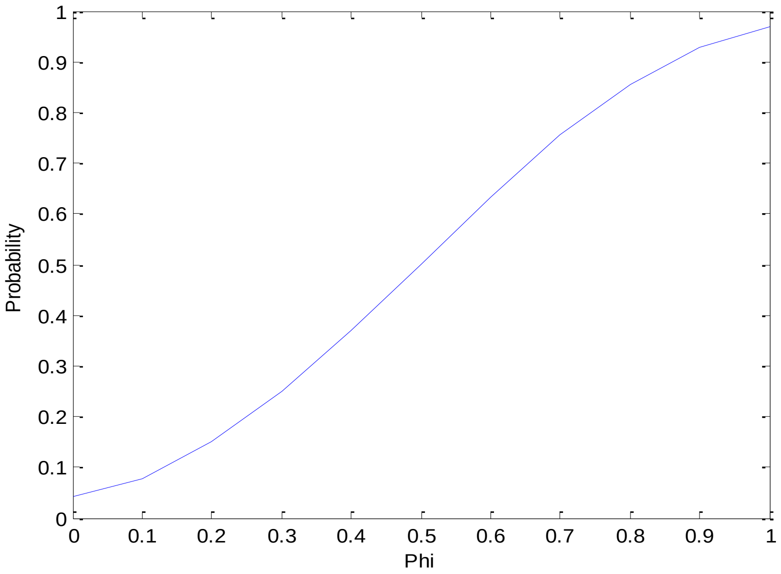


Figure 2.2 Probability variation with respect to parameter ϕ (Agent has four possible actions)

Step 2: Find the next possible states ns from the current state s and determine the following ratio

$$\phi = \frac{\max_{a} Q\left(ns\left(a_{\max}\right), a\right)}{\max_{ns} \max_{a} Q\left(ns, a\right)} \qquad (47)$$
$$if \ \max_{ns} \max_{a} Q(ns, a) = 0 \ or \ \phi < 0 \ then \ \phi = 0.3$$

Here, $ns\left(a_{\max}\right)$ represents the future state that will result if the agent chooses $a_{\max}$ in the current state. This step provides crucial information about how good a particular action is in the given state and provides a measure for assigning correct weightage for the selected action. Note the basic idea behind Eq. (47) is that if the agent follows a greedy policy then it will choose $a_{\max}$. Hence, the Q-values of $ns\left(a_{\max}\right)$ is compared with the Q-values of the all the possible future states to determine the degree of goodness of action $a_{\max}$.

Step 3: Determine $\phi_1 = \pi(\phi + 3.7)$ and $\phi_2 = 0.15\phi_1$. This set of equations will change depending on the number of actions available to the agent.

Step 4: Determine the probability amplitude using Eq. (44) and the square of their magnitude will give the probability distribution. This probability distribution will be used for action selection.

Step 5: Update the Q-values using Eq. (23).

2.3. SIMULATION RESULTS

To evaluate the proposed QiRL algorithm, a typical grid world example with two agents is considered; one is predator and the other is prey. The aim of the task is to find a policy which will let the predator to find the prey with minimum punishment. Two specific cases were considered

In one, the size of the grid world is 20 by 20 and it has obstacles in and around it. The prey is fixed in all time steps but the predator has four possible actions to consider on each time step: four directions (up, down, right and left). This grid world environment is shown in Figure 2.3. For simulations, an episode begins when the predator moves from starting point and ends when it catches the prey or reaches the maximum number of steps. This termination criterion indicates the time agent could explore the grid world and depends on the grid world's size. If we set a smaller number of steps, then the agent has less time to observe action/state to updates the Q-values. Thus, this criterion is important for the very first episodes which agent explores more and isn't biased to a particular policy yet. Agents don't have previous knowledge about the environment and must experience it to find the relationship between the inputs and outputs of the system and update their estimation. To explain experimental set-up in detail, we consider these two scenarios separately, although the first scenario is special case of the second.

2.3.1. Case A: One Predator and Fixed Prey. In this 20 by 20 grid world, each agent can start from different positions in the environment. The predator receives a reward of 100 when it finds the prey, for all other steps it is punished by a reward of -1. The amount of the reward (punishment) is deterministic and independent of the distance between the goal and the agent. The maximum number of steps is 8000; which means that if the predator couldn't capture the prey in 8000 number of steps the episode will be terminated and new episode begins. The discount factor, γ is 0.99. The learning rate for Q-values, α is 0.04.

In this particular problem, the agent has four actions; hence Eq. (46) is used for amplitude amplification. In Figure 2.3, S1 and S2 are two possible starting points for predator and G indicates prey position. In this scenario each cell in the grid world relates to a particular state in the environment. In other words, state definition is based on X and Y coordination. For instance, in Figure 2.3, S1 is in state (cell) 22, S2 is in state 39 and G is in state 379. Overall we have 400 states including vertical and horizontal boundaries (e.g. 1 to 21), obstacles (e.g. 260 or 301) and available positions for agents (e.g. 146). Figures 2.4 and 2.5 show the learning history of the RL agent for starting states S1 and S2 respectively. It can be observed that the number of steps the agent took to reach the final goal state G progressively decreases irrespective of the starting state.

Another scenario that was considered was that there is uncertainty in the agent's movement. It was assumed that if the agent executes a particular action say going up, then there is only 80% chance that the agent will go up; the remaining 20% the agent will move.

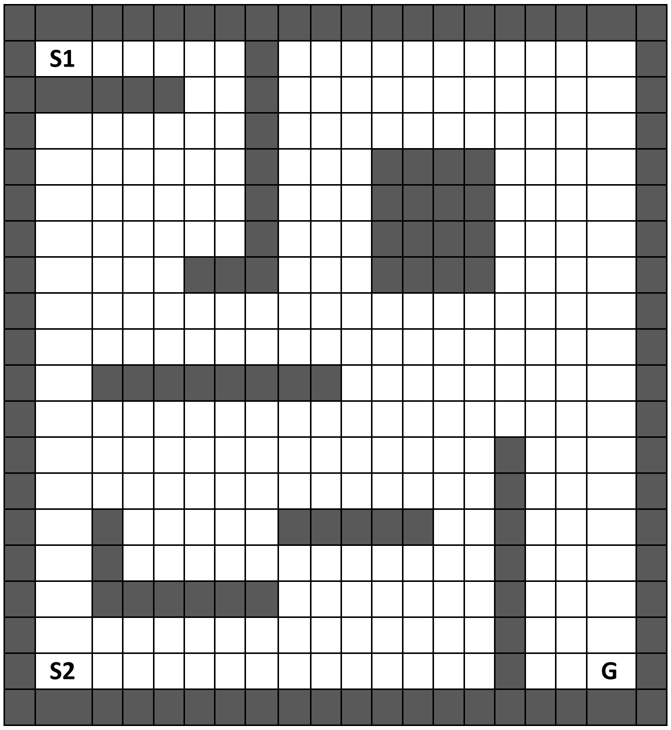
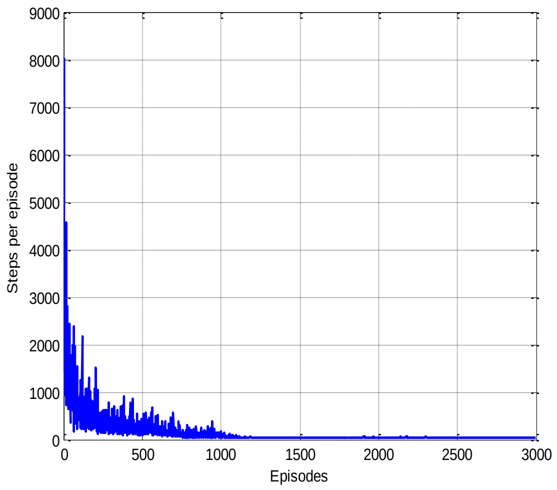


Figure 2.3 Task environment



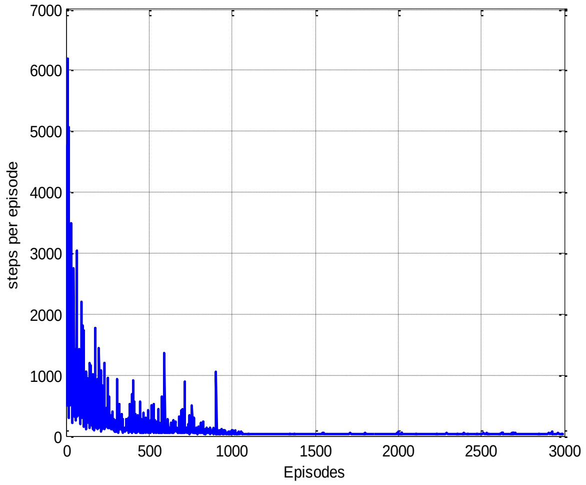
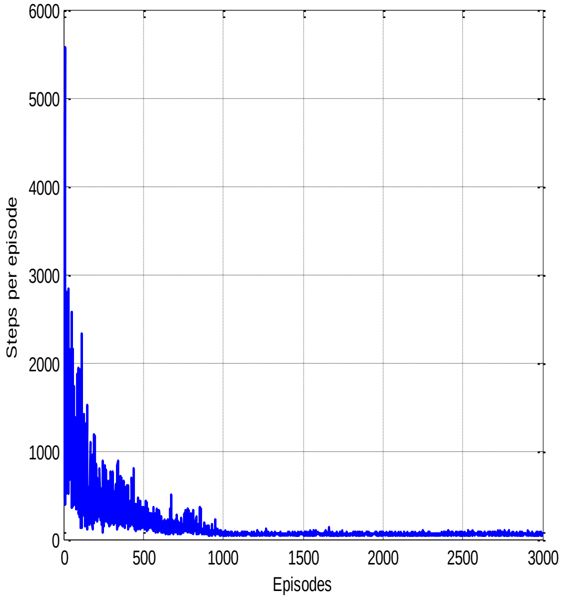


Figure 2.4 Predator learning history for starting state S1

Figure 2.5 Predator learning history for starting state S2

in some other direction. Figures 2.6 and 2.7 illustrate the results obtained. Not much change in the performance was observed. To compare the performance of the QiRL algorithm with traditional RL algorithm a complex problem is simulated in the next section.



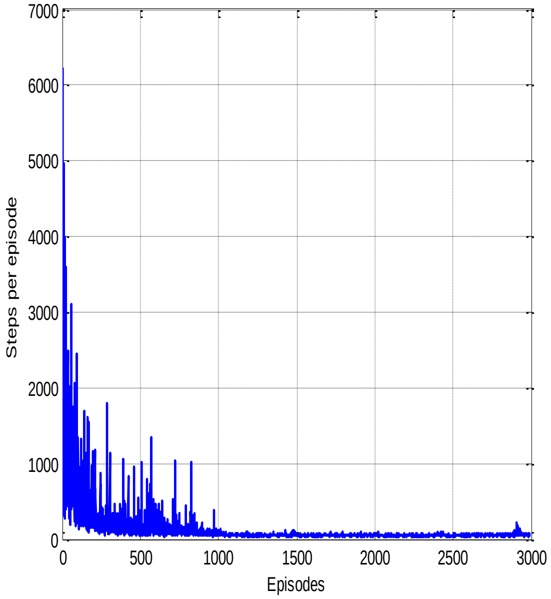


Figure 2.6 Predator learning history with noise for starting state S1

Figure 2.7 Predator learning history with noise for starting state S2

2.3.2. Case B: Two Predator and One Prey. In this experiment a scenario involving two intelligent predators and one randomly moving prey is considered. The size of the grid world is 10 by 10 and there is no obstacle in the environment.

The state of any predator is defined by its relative distance from the prey. Both the predators and the prey will have five actions to choose from**.** They are left, right, up, down and stay put. One of the predators uses QRL for action selection and the other predator uses Softmax method for action selection .The prey moves randomly and all its actions are weighted equally. In case of ‘Softmax method’ the temperature parameter was set at a constant value of 0.9 for determining the probability distribution.

Experimental setup: The experiment has two phases, the training phase and the testing phase. In the training phase both the agents are trained separately to catch the randomly moving prey. During the training the starting position of the predator and the prey are randomly chosen. Each agent is trained for 20000 episodes. The following reward/punishment strategy was used for simulation

a) A punishment of –1 for any action that does not result in predator catching the prey.

b) A reward of 100 for any action that result in predator catching the prey.

The number of steps both the agent takes to catch the prey for each of the training episode is shown in Figures 2.8 and 2.9. It can be observed that initially both the agents take more than 100 steps to catch the prey.

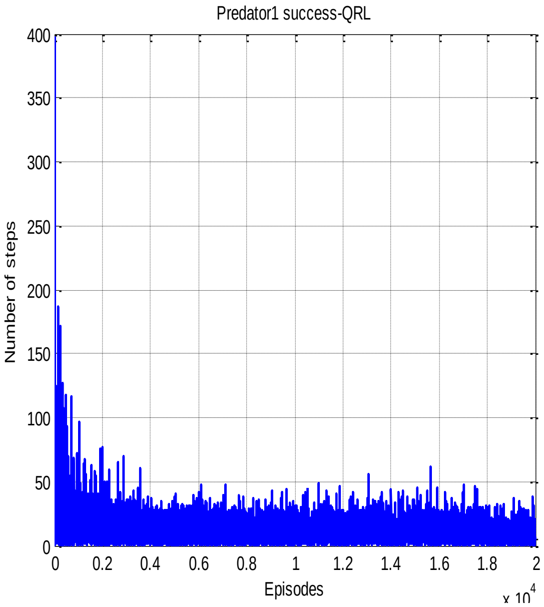


Figure 2.8 Number of steps history for QRL agent alone winning episodes

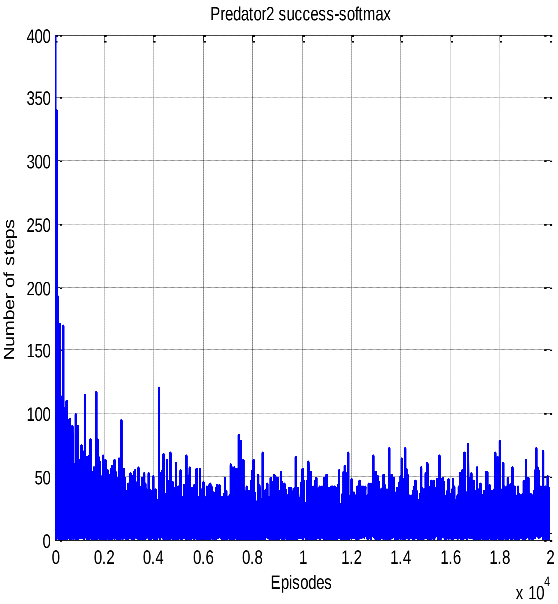


Figure 2.9 Number of steps history for Softmax agent alone winning episodes

As the training progresses the number of steps decreases and both the agents are able to catch the prey in less than 50 steps. In the final stages of the training the QRL agent is able to catch the prey in a minimum of one step and a maximum of 45 steps. The Softmax agent has a slightly higher variance and it takes a minimum of one step and a maximum of 70 steps for catching the prey in the final training stages.

Testing Phase: In the testing phase both the predator agents will start from the same initial grid position and the prey will start from some random position. The predator's initial grid position was fixed at upper left corner of the grid world. The predators do not know about each other and hence can occupy the same grid position any time. Both the QRL and Softmax agent will now use greedy policy for action selection. To compare the performance of QRL algorithm and Softmax algorithm, the Q-values in the intermediate stages of training were used for evaluation. The Q-values considered for evaluation are

a) Initial stage – Q-values at the end of 100th training episode.

b) Intermediate stage – Q-values at the end of 1000th episode.

c) Final stage – Q-values at the end of 20000th episode.

Results: For each case 50000 trails were carried out and the results are presented below. The winning statistics of both agents for the three different test cases is tabulated in Table 2.1. Figures 2.10 and 2.11 show the number of steps the agents takes to catch the prey when Q-values at the end of 100th training episode are used for action selection. Figures 2.12 and 2.13 show the number of steps the agents takes to catch the prey when final Q-values are used for action selection.

The number of episodes the 'Softmax agent' alone wins during the initial stages is quite high compared to 'QRL agent' (Table 2.1). When intermediate and final Q-values are used the QRL agent is able to completely outperform the Softmax agent. The reason for above behavior of QRL agent is during the initial stages the QRL action selection policy favors more exploration compared to Softmax agent. Because of the additional exploration, the Q-values as determined by the QRL agent is complete i.e the rewards the agent will receive from the environment is more accurate. The Softmax action selection policy provides limited exploration during the initial stages and as the Q-values become relatively bigger than the temperature parameter it behaves like a greedy policy. This tendency of Softmax agents limit its exploration space and effectively limit the accuracy of the determined Q-values. During the initial stages the average number of steps required for both the agents is quite high (Figures 2.8 and 2.9). As the training proceeded the number of the steps decreased as shown in Table 2.1 and Figures 2.10 to 2.13. For all the three test cases the average number of steps taken by the QRL agent is less than the Softmax agent.

Table 2.1. Winning statistics of QRL and Softmax agent

Test cases	**QRL alone winning**	**Softmax alone winning**	**Both agents winning**	**Average number of steps**	
				QRL	**Softmax**
Initial stage	14722	31795	3482	32.76	41.23
Intermediate stage	28463	11129	10407	18.27	19.73
Final stage	33511	15494	994	10.53	10.71

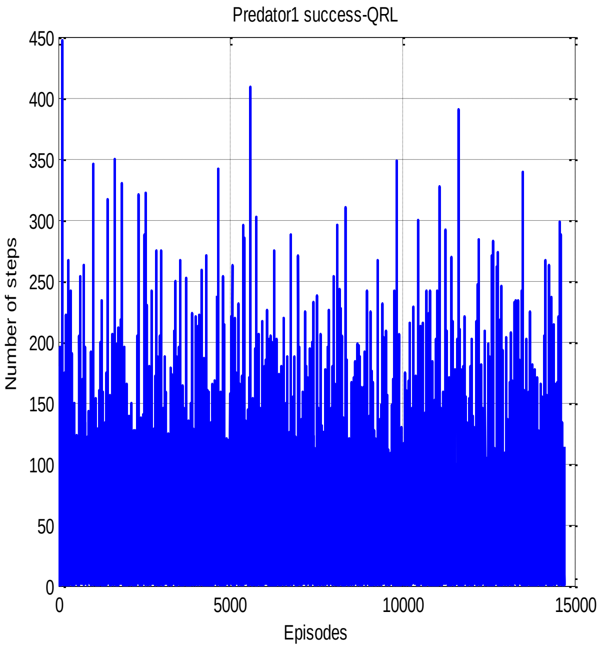


Figure 2.10 Number of steps history for QRL agent winning episodes (initial stage)

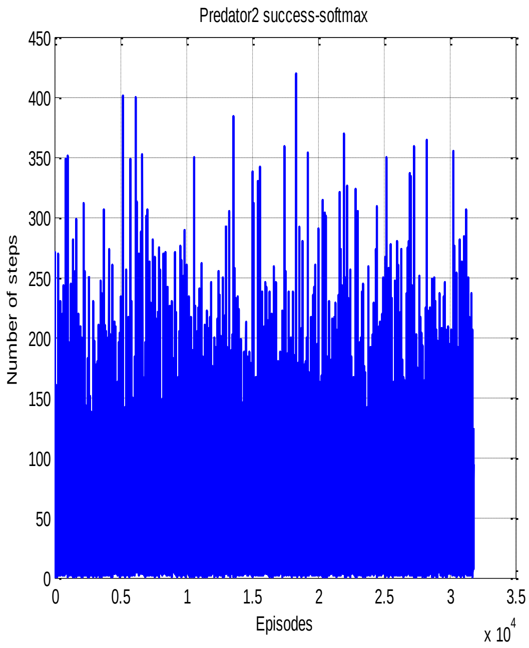


Figure 2.11 Number of steps history for Softmax agent winning episodes (initial stage)

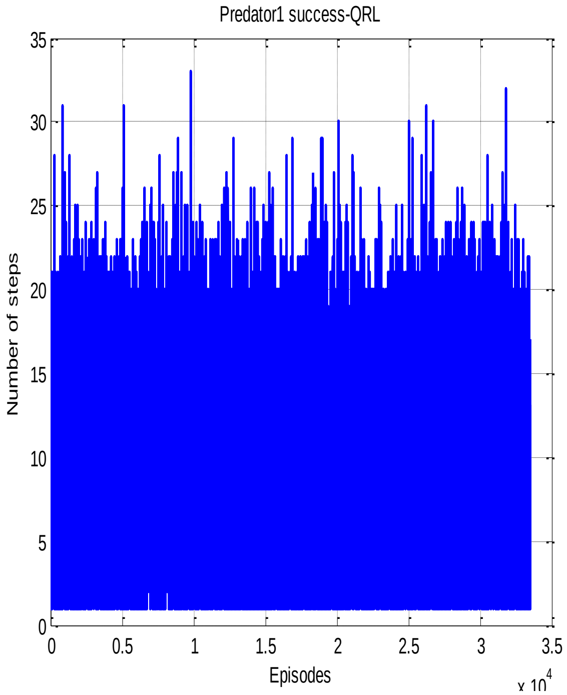


Figure 2.12 Number of steps history for QRL agent winning episodes (final stage)

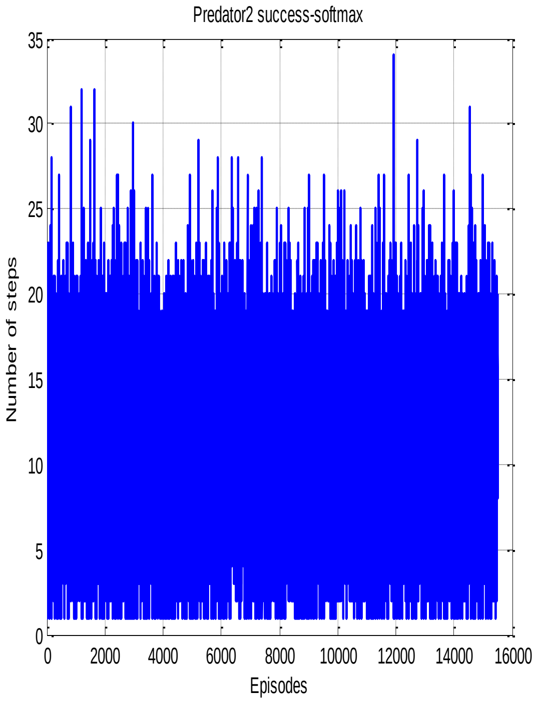


Figure 2.13 Number of steps history for Softmax agent winning episodes (final stage)

2.4. CONCLUSIONS

A new quantum theory based reinforcement learning is proposed. In this novel method, the action selection mechanism uses unitary transformation to assign a probability distribution to the available actions. The proposed approach does not use any design parameter to explicitly specify the relative frequency of explorative-exploitive actions. However, the quantum inspired algorithm considers all possible future rewards to determine the degree of goodness of the selected action. Its performance was evaluated in a complex scenario involving two intelligent predators and one randomly moving prey wherein it totally outperformed the Softmax algorithm. In Softmax algorithm, adjusting temperature to balance the exploration-exploitation trade-off is a very crucial factor. However, in QiRL, this adjustment implicitly happens through the evolution of Q-values.

3. STOCHASTIC OPTIMAL CONTROL USING PATH INTEGRALS

3.1. INTRODUCTION

Many real world systems are nonlinear, dynamical in nature and may require decision-making in uncertain environment. These uncertainties can affect the system behavior in unexpected ways and cause instability. The decision-maker using only the current state information, need to select from a set of possible actions an optimal action such that the dynamical system will evolve as planned. Typically these dynamical systems in uncertain environment are characterized by a set of stochastic differential equations (SDE). Stochastic optimal control theory [2-3, 40] provides the framework to arrive at optimal decisions for systems modeled by SDEs. The two most commonly used approaches for solving the stochastic optimal control problems are the Pontryagin's maximum principle and the Bellman's dynamic programming.

For deterministic optimal control problems, the Pontryagin's maximum principle [3] leads to a set of first order differential equations (state equation and adjoint equation) as necessary conditions for computing the optimal pair (state trajectory and control input). If the final time is fixed, then the optimal control problem is reduced to a two point boundary value problem. The boundary conditions are specified by the state vector at the initial time and the adjoint vector at the final time. However, in case of stochastic optimal control, stochastic maximum principle leads to a backward stochastic differential equation (SDE) as adjoint equation with final time boundary constraint [40]. The above set of SDEs is difficult to solve.

Another approach called Bellman's dynamic programming establishes relationships among a family of optimal control problems with different initial states and time via a partial differential equation called the Hamilton-Jacobi-Bellman (HJB) equation. The solution of this partial differential equation is the optimal cost function. This nonlinear PDE is of first order in deterministic cases and *second* order in stochastic cases. For nonlinear dynamical systems and even for some linear systems getting analytical solutions for the controller using any of the above two approaches is practically impossible. Hence, one has to resort to numerical techniques for solving the optimal control problem. However, the associated computational cost is very high for higher dimensional systems.

Approximate dynamic programming [41-44] provides a way out of the above bottlenecks by using reinforcement learning for identifying the optimal value function and the optimal controller. In this approach, the value function is incrementally updated as new information becomes available. A typical reinforcement learning architecture consists of an actor that provides optimal action (control input) information and a critic that criticizes the actions taken by the actor. Generally, neural networks are used to approximate both the control function and the cost function. The training of the respective networks takes place alternatively. The advantage of the above approach, called the 'adaptive critic' is that information regarding optimal trajectory or control action need not be known beforehand. Many variants of adaptive critic designs (ACDs) are now available [45]. The most common adaptive critic architectures are Heuristic Dynamic Programming (HDP) and Dual Heuristic Programming (DHP). In the HDP design, the critic network maps the control and state function to the optimal cost function and the action network maps the state function to the optimal control function. In the DHP design, the critic network directly maps the gradient of the cost function to the system states and thus provides better performance. To alleviate the computational burden of two network ACDs, single network adaptive critic

(SNAC) architectures [46] have been proposed. In SNAC, the dual network acts both as an actor and critic. All the above mentioned approaches require a model of the system in some form or other to train their critic networks. The success of ACDs can be measured by the number of current literature available in this topic and its simplicity of implementation has been exploited in a wide variety of applications [47-49].

The potential of ACDs in solving stochastic optimal control problems has not been fully explored. In the presence of Weiner noise, the HJB equation characterizing optimal value functions becomes second order and embeds information about process noise variance. In this paper, a continuous time single network adaptive critic architecture is proposed which uses the above information while solving the stochastic optimal control problem.

Recently, Kappen [50] has proposed the idea of using logarithmic transformations of the cost function to convert the nonlinear stochastic HJB equation into a linear HJB equation. The above linear transformation is possible for control affine non-linear dynamical systems with quadratic control cost function and arbitrary state cost function. The Feynman-Kac Lemma connects the solution of the above linear PDE to a forward diffusion process. Hence, the transformed cost function can be computed as an expected value under a forward diffusion process. For numerical computation this expected value is represented in terms of path integrals. The above formulation of stochastic optimal control theory is called the path integral control. One of the major drawbacks of the path integral control framework is the requirement of large number of trajectory samples to accurately estimate the optimal controller. Hence, generating comprehensive solutions for the entire operating domain of the system will be computationally expensive. However, adaptive critic formulations can effectively handle the curse of dimensionality problems. In this paper, a continuous time adaptive critic algorithm that effectively uses the path integral framework is proposed.

3.2. PROBLEM FORMULATION

Consider a system described by a stochastic differential equation of the following form

$$dx = f(x)dt + B\left(udt + \sigma d\xi\right) \tag{48}$$

where $x \in \Omega$ is the state vector of dimension n, $\Omega \subset \mathbb{R}^n$ is a compact subset, $u \in \mathbb{R}^m$ is the control vector of dimension m, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $B \in \mathbb{R}^{n\times m}$ represent known system dynamics, $\xi \in \mathbb{R}^m$ is the Weiner process with mean 0 and variance t and $\sigma \in \mathbb{R}^{m\times m}$ is the noise variance of the dynamics. The objective of the control problem is to find the optimal control $u^*\left(x,t\right),\ t \le t_f$ that minimizes the following cost function:

$$C\left(x,t\right) = \left\langle \phi\left(x\left(t_f\right)\right) + \int_t^{t_f} \left(Q\left(x\left(\theta\right)\right) + \frac{1}{2} u\left(\theta\right)^T Ru\left(\theta\right) \right) d\theta \right\rangle_{x,t} \tag{49}$$

where $\phi\left(.\right)$ is the terminal cost at the final time t_f, $Q\left(.\right)$ is the instantaneous state cost and $R \in \mathbb{R}^{m\times m} > 0$ is the control weighting matrix. The optimal cost-to-go is defined by

$$J\left(x,t\right) = \min_u C(x,t) \tag{50}$$

From stochastic optimal control theory [40], the optimal cost-to-go function is the solution of the stochastic HJB equation

$$
\begin{aligned}
-\frac{\partial J}{\partial t} &= \min_u\left(\frac{1}{2}u^T Ru + Q\left(x\right) + \left(f(x) + Bu\right)^T \frac{\partial J}{\partial x} + \frac{1}{2}\text{trace}\left(B\sigma\sigma^T B^T \frac{\partial^2 J}{\partial x^2}\right)\right) \\
&= \min_u\left(\frac{1}{2}u^T Ru + Q\left(x\right) + L_u J\right)
\end{aligned} \quad (51)
$$

with the boundary condition $J\left(x, t_f\right) = \phi\left(x\left(t_f\right)\right)$. Here, L_u is the infinitesimal generator of the stochastic process specified in Eq. (48); It is defined by

$$
L_u = \left(f\left(x\right) + Bu\right)^T \frac{\partial}{\partial x} + \frac{1}{2}\text{trace}\left(B\sigma\sigma^T B^T\right)\frac{\partial^2}{\partial x^2} \quad (52)
$$

Since the cost function is quadratic in u, minimization of Eq. (51) yields

$$
u^*\left(x, t\right) = -R^{-1}B^T \frac{\partial J}{\partial x} \quad (53)
$$

Then, Eq. (51) becomes

$$
-\frac{\partial J}{\partial t} = \frac{1}{2}\frac{\partial J}{\partial x}^T BR^{-1}B^T \frac{\partial J}{\partial x} + Q\left(x\right) + L_{u^*} J \quad (54)
$$

The stochastic HJB equation given in Eq. (54) is nonlinear and does not lend itself to analytical solutions. Hence, numerical techniques are needed to find solutions. In this paper, a continuous time adaptive critic learning scheme is proposed to solve the stochastic HJB equation.

3.3. ADAPTIVE CRITIC SCHEME FOR STOCHASTIC SYSTEMS

For stochastic systems, the recursive expression for cost function involves the expectation operator as shown below:

$$\begin{aligned} J\left(x,t\right) &= \left\langle \int_{t}^{t+\tau} \left(Q\left(x\left(\theta\right)\right) + \frac{1}{2}u^{*}\left(\theta\right)^{T} Ru^{*}\left(\theta\right) \right) d\theta \right\rangle_{x} + \left\langle J\left(x\left(t+\tau\right),t+\tau\right)\right\rangle_{x} \\ &= \left\langle g\left(x\left(t \to t+\tau\right),u^{*}\right)\right\rangle_{x} + \left\langle J\left(x\left(t+\tau\right),t+\tau\right)\right\rangle_{x} \end{aligned} \tag{55}$$

Here, $g\left(x\left(t \to t+\tau\right),u^{*}\right)$ is the path cost and $\tau > 0$. Note that the computation of $J\left(x,t\right)$ requires a set of sample paths. In adaptive dynamic programming literature, noise effect is typically ignored. However, to accurately estimate the cost function, sufficient number of sample trajectories is needed. For multidimensional problems, this might result in high computational cost. One way of reducing the computational cost is to use importance sampling(reference). Path integral control approach associates with each sample path a probability value. The optimal cost function can then be expressed as a weighted sum of individual path costs. The adaptive critic approach proposed in this paper uses the path integral formulation to adaptively sample trajectories of importance. The theory behind the path integral control approach is explained in the next subsection.

3.3.1. Path Integral Formulation. Assume that the noise variance of the stochastic system given in Eq. (48) can be related to a constant parameter λ by the relation

$$\sigma\sigma^{T} = \lambda R^{-1} \tag{56}$$

Then the path integral formulation [50-52] allows us to convert the nonlinear stochastic HJB equation into a linear form by using a logarithmic transformation $J\left(x,t\right) = -\lambda \log \psi\left(x,t\right)$ which results in

$$\frac{\lambda}{\psi}\frac{\partial\psi}{\partial t}=\left(\frac{\lambda}{\psi}\right)^{2}\frac{1}{2}\frac{\partial\psi}{\partial x}^{T}BR^{-1}B\frac{\partial\psi}{\partial x}+Q\left(x\right)$$
$$-\left(f(x)+BR^{-1}B^{T}\left(\frac{\lambda}{\psi}\right)\frac{\partial\psi}{\partial x}\right)^{T}\left(\frac{\lambda}{\psi}\frac{\partial\psi}{\partial x}\right) \tag{57}$$
$$+\frac{1}{2}\text{trace}\left(-\frac{\lambda}{\psi}\left(-\frac{1}{\psi}\frac{\partial\psi}{\partial x}\left(\frac{\partial\psi}{\partial x}\right)^{T}+\frac{\partial^{2}\psi}{\partial x^{2}}\right)B\sigma\sigma^{T}B^{T}\right)$$

The linear stochastic HJB is given by

$$\frac{\partial\psi}{\partial t}=\left(\frac{Q\left(x\right)}{\lambda}-f(x)^{T}\frac{\partial}{\partial x}-\frac{1}{2}\text{trace}\left(B\sigma\sigma^{T}B^{T}\frac{\partial^{2}}{\partial x^{2}}\right)\right)\psi \tag{58}$$

with the boundary constraint $\psi(x,t_f)=\exp\left(-\phi\left(x\left(t_f\right)\right)/\lambda\right)$. The above linear PDE is called the Chapman-Kolmorgov backward equation. The Feynman-Kac Lemma [84] connects the solution of the above PDE to a forward diffusion process. This solution can be computed as an expected value using the Feynman-Kac formula:

$$\psi(x,t)=\left\langle \exp\left(-\int_{t}^{t+\tau}\frac{Q\left(\bar{y}\left(\theta\right)\right)}{\lambda}d\theta\right)\psi\left(\bar{y}\left(t+\tau\right),t+\tau\right)\right\rangle_{x} \tag{59}$$

The expectation value is the sum over all possible sample paths originating from x at time t and propagated until time reaches $t+\tau$. Propagation is performed by using an uncontrolled forward diffusion process given below:

$$d\bar{y}=f(\bar{y})dt+B\sigma d\xi \tag{60}$$

with $\bar{y}\left(t\right)=x$.For writing Eq. (59) as a path integral, the time interval $\left[t,\ t+\tau\right]$ is split into N intervals of equal length Δt with $t=t_1<t_2<t_3........<t_{N+1}=t+\tau$. The

corresponding states at these time intervals are represented by $x_1 = x,\ x_2,\ ....,\ x_{N+1} = \bar{y}\left(t+\tau\right)$. The resulting path integral equation is given by

$$\begin{aligned}&\psi(x,t)\\&=\lim_{\Delta t\to 0}\int dx_2\int dx_3.....\int dx_{N+1}\prod_{k=2}^{N+1}\rho(x_k,t_k\,|\,x_{k-1},t_{k-1})\exp\left(-\frac{Q(x_{k-1})}{\lambda}\Delta t\right)\psi(x_{N+1},t_{N+1})\end{aligned} \tag{61}$$

where $\rho(x_k,t_k\,|\,x_{k-1},t_{k-1})$ is the transition probability of the uncontrolled dynamics given in Eq. (60) to propagate from $\left(x_k,t_k\right)$ to $\left(x_{k-1},t_{k-1}\right)$. In most systems, the dimension of the control vector is usually less than that of the state vector. Since it is assumed that both noise and the control input acts on the same subspace, the states that are not directly actuated will behave deterministically. Thus, the transition probability will depend only on the directly actuated states. Hence, the state vector is partitioned into directly actuated and non-directly actuated states. Equation (60) is rewritten in the following form:

$$\begin{bmatrix} d\bar{y}^{na} \\ d\bar{y}^{a} \end{bmatrix} = \begin{bmatrix} f^{na}\left(\bar{y}\right) \\ f^{a}\left(\bar{y}\right) \end{bmatrix} + \begin{bmatrix} 0_{(n-m)\times m} \\ B^{a} \end{bmatrix}\sigma d\xi \tag{62}$$

where $\bar{y} = \left[\bar{y}^{na} \quad \bar{y}^{a}\right]^T$ with $\bar{y}^{na} \in \mathbb{R}^{(n-m)\times 1},\ \bar{y}^{a} \in \mathbb{R}^{m\times 1}$, $f\left(\bar{y}\right) = \left[f^{na}\left(\bar{y}\right) \quad f^{a}\left(\bar{y}\right)\right]^T$ with $f^{na} \in \mathbb{R}^{(n-m)\times 1}$, $f^{a} \in \mathbb{R}^{m\times 1}$ and $B = \left[0_{(n-m)\times m} \quad \left(B^a\right)^T\right]^T$ with $B^a \in \mathbb{R}^{m\times m}$. The transition probability of the forward diffusion process is given by

$$\begin{aligned}&\rho(x_k,t_k\,|\,x_{k-1},t_{k-1}) \propto \rho(x_k^a,t_k\,|\,x_{k-1},t_{k-1})\\&\rho(x_k^a,t_k\,|\,x_{k-1},t_{k-1})\\&\quad=\frac{1}{\sqrt{\det\left(\Delta t 2\pi B^a\sigma\sigma^T\left(B^a\right)^T\right)}}\exp\left(-\frac{\Delta t}{2}\Sigma\left(t_k\right)^T\left(B^a\sigma\sigma^T\left(B^a\right)^T\right)^{-1}\Sigma\left(t_k\right)\right)\end{aligned} \tag{63}$$

$$\Sigma\left(t_k\right)=\frac{x_k^a-x_{k-1}^a}{\Delta t}-f^a\left(x_{k-1}\right) \tag{64}$$

By using Eq. (63) and the assumption given in Eq.(56), Eq. (61) is rewritten in the following path integral form as

$$\psi(x,t)$$
$$=\frac{1}{\sqrt{\left(\det\left(\Delta t 2\pi B^a\sigma\sigma^T\left(B^a\right)^T\right)\right)^N}}\lim_{\Delta t\to 0}\int dx_2^a\int dx_3^a.....\int dx_{N+1}^a\exp\left(-\frac{S_{path}^{\Delta t}}{\lambda}\right)\psi(x_{N+1},t_{N+1}) \tag{65}$$
$$S_{path}^{\Delta t}=\Delta t\sum_{k=2}^{N+1}\left(\frac{1}{2}\Sigma\left(t_k\right)^T\left(B^aR^{-1}\left(B^a\right)^T\right)^{-1}\Sigma\left(t_k\right)+Q(x_{k-1})\right)$$

Optimal control $u^*(x,t)$ is computed by the relation

$$u^*(x,t)=\lambda R^{-1}B^T\frac{1}{\psi\left(x,t\right)}\frac{\partial\psi}{\partial x} \tag{66}$$

From Eq. (65), the probability of a sample path contributing to the computation of the optimum cost-to-go function is given by

$$P\left(x_1,x_2,....x_{N+1}/x_1,t_1\right)=\frac{1}{\sqrt{\left(\det\left(\Delta t 2\pi B^a\sigma\sigma^T\left(B^a\right)^T\right)\right)^N}}\frac{\exp\left(-\frac{S_{path}^{\Delta t}}{\lambda}\right)\psi\left(x_{N+1},t_{N+1}\right)}{\psi\left(x_1,t_1\right)} \tag{67}$$

The proposed adaptive critic scheme uses Eq. (67) to adaptively sample the trajectories and is described next.

3.3.2. Adaptive Critic Algorithm. Let $J^i\left(x,\ t\right)$ be the cost function estimate at the end of iteration $i \geq 0$. Then, the iterative procedure is mathematically represented by the following set of equations:

i) Generate trajectories using the forward diffusion process given in Eq. (60). Sample the generated trajectories according to the following probability distribution:

$$P^i\left(x,....x_{N+1}/x,t\right) = \frac{1}{\sqrt{\left(\det\left(\Delta t 2\pi B^a \sigma\sigma^T \left(B^a\right)^T\right)\right)^N}} \frac{\exp\left(-\frac{S_{path}^{\Delta t}}{\lambda}\right)\psi^i\left(x_{N+1},t+\tau\right)}{\psi^i\left(x,t\right)} \tag{68}$$

where $\psi^i\left(x,t\right) = \exp\left(-J^i\left(x,t\right)/\lambda\right)$. Let $\bar{y}\left(t\right)$ represent one of the sampled trajectories.

ii) Compute $\psi^{i+1}\left(x,t\right)$ using the following relation

$$\begin{aligned}&\hat{\psi}^{i+1}(x,t)\\ &= \frac{1}{\sqrt{\left(\det\left(\Delta t 2\pi B^a \sigma\sigma^T \left(B^a\right)^T\right)\right)^N}} \lim_{\Delta t \to 0} \int dx_2^a \int dx_3^a .....\int dx_{N+1}^a \exp\left(-\frac{S_{path}^{\Delta t}}{\lambda}\right)\psi^i(x_{N+1},t_{N+1})\end{aligned} \tag{69}$$

iii) Compute u^{i+1} from $\hat{\psi}^{i+1}(x,t)$ using the optimal control relation given in Eq. (66)

iv) Compute the cost function $J^{i+1}\left(x,\ t\right)$ using the following relation:

$$\begin{aligned}J^{i+1}\left(x,t\right) &= \left\langle \int_t^{t+\tau}\left(Q\left(\bar{y}\left(\theta\right)\right) + \frac{1}{2}u^{i+1}\left(\theta\right)^T R u^{i+1}\left(\theta\right)\right)d\theta + J^i\left(x_{N+1},t+\tau\right)\right\rangle_x \\ &= \left\langle g\left(\bar{y}\left(t \to t+\tau\right),u^{i+1}\left(t \to t+\tau\right)\right) + J^i\left(x_{N+1},t+\tau\right)\right\rangle_x\end{aligned} \tag{70}$$

$$J^{i+1}\left(x,t_f\right)=J^i\left(x,t_f\right) \tag{71}$$

For the next iteration, the paths are sampled using the updated cost function $J^{i+1}\left(x,t\right)$. Convergence analysis of this iterative procedure is performed in the next section.

3.4. CONVERGENCE ANALYSIS OF THE ADAPTIVE CRITIC SCHEME

3.4.1. Relation Between $J^{i+1}\left(x,t\right)$ and $J^i\left(x,t\right)$. In this section, a partial differential equation characterizing the relation between $J^{i+1}\left(x,t\right)$ and $J^i\left(x,t\right)$ is derived.

Theorem 1: Assume an arbitrary function $J^i\left(x,t\right):\mathbb{R}^n\times\mathbb{R}\rightarrow\mathbb{R}$ with continuous J^i, $\frac{\partial J^i}{\partial t}$, $\frac{\partial J^i}{\partial x}$ and $\frac{\partial^2 J^i}{\partial x^2}$ and it satisfies the condition

$$\left\|J^i\right\|+\left\|\frac{\partial J^i}{\partial t}\right\|+\left\|x\right\|\left\|\frac{\partial J^i}{\partial x}\right\|+\left\|x\right\|^2\left\|\frac{\partial^2 J^i}{\partial x^2}\right\|<\gamma\left(1+\left\|x\right\|^2\right) \tag{72}$$

where γ is a suitable constant. Then, the cost function $\left(J^{i+1}\right)$ computed using Eqs. (68) to (71) satisfies the following condition:

$$\frac{\partial J^{i+1}}{\partial t}+L_{u^{i+1}}J^{i+1}+Q\left(x\right)+\frac{1}{2}\left(u^{i+1}\right)^T Ru^{i+1}=0 \tag{73}$$

Proof: Equation (70) can be rewritten as

$$\begin{aligned}&J^{i+1}\left(x,t\right)-J^i\left(x,t\right)\\&=\left\langle\int_t^{t+\tau}\left(Q\left(\bar{y}\left(\theta\right)\right)+\frac{1}{2}u^{i+1}\left(\theta\right)^T Ru^{i+1}\left(\theta\right)\right)d\theta+J^i\left(x_{N+1},t+\tau\right)-J^i\left(x,t\right)\right\rangle_x\end{aligned} \tag{74}$$

By applying Ito’s integration [84] formula to $J^i\left(x_{N+1},t+\tau\right)$ along the trajectory $\overline{y}\left(\theta\right)$ we get

$$\begin{aligned} & J^i\left(x_{N+1},t+\tau\right)-J^i\left(x,t\right) \\ & =\int_t^{t+\tau}\frac{\partial J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial t}d\theta+\int_t^{t+\tau}\left(\frac{\partial J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial\overline{y}}\right)^T d\overline{y}+\frac{1}{2}\int_t^{t+\tau}\left(d\overline{y}\right)^T\frac{\partial^2 J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial\overline{y}^2}d\overline{y} \end{aligned} \tag{75}$$

Note that the path $\overline{y}\left(\theta\right)$ generated during the sampling process is equivalent to propagating the following stochastic process

$$d\overline{y}=f(\overline{y})dt+Bu^{i+1}+B\sigma d\xi \tag{76}$$

Substituting Eq. (76) in Eq. (75) results in

$$\begin{aligned} & J^i\left(x_{N+1},t+\tau\right)-J^i\left(x,t\right) \\ & \quad =\int_t^{t+\tau}\frac{\partial J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial t}d\theta+\int_t^{t+\tau}\left(\left(f\left(\overline{y}\right)+bu^{i+1}\right)dt+b\sigma d\xi\right)^T\frac{\partial J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial x} \\ & \qquad +\frac{1}{2}\int_t^{t+\tau}\text{trace}\left(b\sigma\sigma^T b^T\right)\frac{\partial^2 J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial x^2}dt \end{aligned} \tag{77}$$

Taking expectation on both sides leads to

$$\begin{aligned} & \left\langle J^i\left(x_{N+1},t+\tau\right)-J^i\left(x,t\right)\right\rangle_{x,t} \\ & \quad =\left\langle\int_t^{t+\tau}\frac{\partial J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial t}d\theta+\int_t^{t+\tau}\left(f\left(\overline{y}\right)+bu^{i+1}\right)^T\frac{\partial J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial\overline{y}}dt\right\rangle_{x,t} \\ & \qquad +\left\langle\frac{1}{2}\int_t^{t+\tau}\text{trace}\left(b\sigma\sigma^T b^T\right)\frac{\partial^2 J^i\left(\overline{y}\left(\theta\right),\theta\right)}{\partial\overline{y}^2}dt\right\rangle_{x,t} \end{aligned} \tag{78}$$

Let $J_d^{i+1}\left(x,t\right)=J^{i+1}\left(x,t\right)-J^i\left(x,t\right)$. Hence, Eq. (74) becomes

$$J_d^{i+1}(x,t) = \left\langle \int_t^{t+\tau} \begin{pmatrix} Q(\bar{y}(\theta)) + \frac{1}{2}u^{i+1}(\theta)^T Ru^{i+1}(\theta) + \left(f(\bar{y}) + bu^{i+1}\right)^T \frac{\partial J^i(\bar{y}(\theta),\theta)}{\partial \bar{y}} \\ + \frac{\partial J^i(\bar{y}(\theta),\theta)}{\partial t} + \frac{1}{2}\mathrm{trace}\left(b\sigma\sigma^T b^T\right) \frac{\partial^2 J^i(\bar{y}(\theta),\theta)}{\partial \bar{y}^2} \end{pmatrix} d\theta \right\rangle \tag{79}$$

By using the definition of Eq. (52), Eq. (79) is simplified as

$$\begin{aligned} &J_d^{i+1}(x,t) \\ &= \left\langle \int_t^{t+\tau} \left(Q(\bar{y}(\theta)) + \frac{1}{2}u^{i+1}(\theta)^T Ru^{i+1}(\theta) + \frac{\partial J^i(\bar{y}(\theta),\theta)}{\partial t} + L_{u^{i+1}} J^i(\bar{y}(\theta),\theta) \right) d\theta \right\rangle \\ &= \left\langle \int_t^{t+\tau} \kappa(\bar{y}(\theta),\theta) d\theta \right\rangle \end{aligned} \tag{80}$$

Equation (80) is claimed as the solution of the following PDE

$$\frac{\partial J_d^{i+1}}{\partial t} + L_{u^{i+1}} J_d^{i+1} = -\kappa(x,t) \tag{81}$$

with boundary condition $J_d^{i+1}(x_{N+1}, t+\tau) = 0$. To prove the above claim, expressing $dJ_d^{i+1}(\bar{y}(\theta),\theta)$ in terms of Ito's lemma leads to

$$\begin{aligned} dJ_d^{i+1}(\bar{y}(\theta),\theta) &= \frac{\partial J_d^{i+1}}{\partial t} d\theta + L_{u^{i+1}} J_d^{i+1} d\theta + (b\sigma d\xi)^T \left(\frac{\partial J_d^{i+1}}{\partial \bar{y}} \right) \\ &= -\kappa(\bar{y}(\theta),\theta) d\theta + (b\sigma d\xi)^T \left(\frac{\partial J_d^{i+1}}{\partial \bar{y}} \right) \end{aligned} \tag{82}$$

Integrating Eq. (82) from $\theta = t$ to $\theta = t+\tau$ and taking expectation results in

$$\left\langle J_d^{i+1}(x_{N+1}, t+\tau) \right\rangle - J_d^{i+1}(x,t) = -\left\langle \int_t^{t+\tau} \kappa(\bar{y}(\theta),\theta) d\theta \right\rangle \tag{83}$$

Since $J_d^{i+1}\left(x_{N+1},t+\tau\right)=0$, $J_d^{i+1}\left(x,t\right)=\left\langle \int_t^{t+\tau}\kappa\left(\bar{y}\left(\theta\right),\theta\right)d\theta\right\rangle$. Thus, Eq. (80) can be construed as the solution of the PDE defined in Eq. (81). Expanding the terms of the PDE results in

$$\frac{\partial J^{i+1}}{\partial t}+L_{u^{i+1}}J^{i+1}+Q\left(x\right)+\frac{1}{2}\left(u^{i+1}\right)^T Ru^{i+1}=0 \tag{84}$$

with the boundary condition $J^{i+1}(x,t_f)=\phi\left(x\left(t_f\right)\right)$. Hence, Theorem 1 is proved. The above equation describes the relationship between $J^{i+1}\left(x,t\right)$ and $J^i\left(x,t\right)$. Note that u^{i+1} depends on J^i

3.4.2. Scalar Diffusion Problem. The convergence analysis of the adaptive critic scheme described in the previous section is performed on a scalar diffusion problem. The governing equations of the diffusion process are given by

$$dx=udt+\sigma d\xi \tag{85}$$

The objective of the controller design is to minimize the following objective function

$$c(x,t)=\left\langle Q_f x\left(t_f\right)^2+\int_t^{t_f}\left(\frac{1}{2}Ru^2\right)dt\right\rangle_x \tag{86}$$

For the adaptive critic scheme, the optimum cost function at iteration i is approximated as follows

$$J^i\left(x,t\right)=a^i(t)+T^i\left(t\right)x^2 \tag{87}$$

where $a^i(t) > 0$ and $T^i(t) > 0$. The iteration process can be started with any $T^0(t) \geq 0$. For the purpose of convergence analysis $N = 1,\ \Delta t = \tau$ and $x_{N+1} = y$.

The state transition probability for the uncontrolled dynamics of the stochastic system defined in Eq. (85) is given by

$$\rho\left(x \to y / x, t\right) = \frac{1}{\sqrt{2\pi\sigma^2\tau}} \exp\left(-\frac{\left(y-x\right)^2}{2\sigma^2\tau}\right) \tag{88}$$

The transformed cost function $\psi^{i+1}\left(x,t\right)$ can now be computed as

$$\begin{aligned} \psi^{i+1}\left(x,t\right) &= \int dy \rho\left(y, t+\tau / x, t\right) \psi^i\left(y, t+\tau\right) \\ &= \int \frac{dy}{\sqrt{2\pi\sigma^2\tau}} \exp\left(-\frac{\left(y-x\right)^2}{2\sigma^2\tau}\right) \exp\left(\frac{-a^i(t+\tau) - T^i\left(t+\tau\right) y^2}{R\sigma^2}\right) \end{aligned} \tag{89}$$

Evaluation of Eq. (89) results in

$$\psi^{i+1}\left(x,t\right) = \frac{1}{\sqrt{R\sigma^2 + 2\sigma^2\tau T^i\left(t+\tau\right)}} \exp\left(-\frac{x^2}{\sigma^2}\left(\frac{T^i\left(t+\tau\right)}{R + 2\tau T^i\left(t+\tau\right)}\right) - \frac{a^i\left(t+\tau\right)}{R\sigma^2}\right) \tag{90}$$

The control u^{i+1} can now be computed as

$$u^{i+1} = -R^{-1} \frac{-\lambda}{\psi^{i+1}\left(x,t\right)} \frac{\partial \psi^{i+1}}{\partial x} \tag{91}$$

Evaluation of Eq. (91) leads to

$$u^{i+1} = -\frac{x 2 T^i\left(t+\tau\right)}{R + \tau 2 T^i\left(t+\tau\right)} = -F^{i+1} x \tag{92}$$

To interpret the above control expression consider the deterministic discrete dynamics of the stochastic system defined in Eq. (85)

$$x_{s+1} = x_s + \tau u_s \tag{93}$$

with $s = 0,1,2....F$. The discretized cost function is defined as

$$c^d(x_s, s) = Q_f x_F^2 + \sum_{j=s}^{s} \frac{\tau}{2} R u_j^2 = T_{ds} x_s^2,\ T_{dF} = Q_f \tag{94}$$

The dynamics of the cost function parameter T_{ds} is described by the discrete Riccati equation

$$T_{ds} = 2T_{d(s+1)} - \frac{4\tau^2 T_{d(s+1)}^2}{R\tau + 2\tau^2 T_{d(s+1)}} = \frac{2RT_{d(s+1)}}{R + 2\tau T_{d(s+1)}} \tag{95}$$

Comparing Eq. (113) and Eq. (122), the expressions for F^{i+1} and T_{ds} are very similar. Thus the sampling process (Eqs.(88) to (90)) results in a controller expression that depends on the discrete Riccati equation solution.

To compute $T^{i+1}(t)$ substitute $J^{i+1}(x,t) = a^{i+1}(t) + T^{i+1}x^2$ in Eq. (84) that results in

$$\dot{T}^{i+1}(t)x^2 + \dot{a}^{i+1}(t) + \sigma^2 T^{i+1}(t) + 2T^{i+1}(t)xu^{i+1} + \frac{1}{2}Ru_{i+1}^2 = 0 \tag{96}$$

Let the parameter $a^{i+1}(t)$ be updated such that

$$a^{i+1}(t) = \int_t^T \frac{1}{2}\sigma^2 T^{i+1}(\theta)\,d\theta \tag{97}$$

Substituting Eq. (97) and Eq. (113) in Eq. (96) results in

$$\dot{T}^{i+1}(t)x^2 - 2T^{i+1}(t)x^2F^{i+1} + \frac{1}{2}Rx^2\left(F^{i+1}\right)^2 = 0 \tag{98}$$

$$\dot{T}^{i+1}(t) = 2T^{i+1}(t)F^{i+1} - \frac{1}{2}R\left(F^{i+1}\right)^2 \tag{99}$$

Equation (99) is the continuous time Riccati equation with the boundary condition $T\left(t_f\right) = Q_f$. For computer implementation $\dot{T}^{i+1}(t)$ is approximated as

$$\dot{T}^{i+1}(t) = \frac{T^{i+1}(t+\tau) - T^{i+1}(t)}{\tau} \tag{100}$$

By using Eq. (99) and Eq. (126) a backward difference equation satisfying the boundary condition $T^{i+1}\left(t_f\right) = Q_f$ can be derived as

$$T^{i+1}(t) = \frac{1}{2}\frac{1}{\left(2\tau F^{i+1}+1\right)}\left(2T^{i+1}(t+\tau) + R\tau\left(F^{i+1}\right)^2\right) \tag{101}$$

The adaptive critic iterative procedure for the scalar diffusion problem can be summarized as follows

i) Select $T^0(t) \geq 0$

ii) Compute $F^{i+1}(t)$ using

$$F^{i+1} = \frac{2T^i(t+\tau)}{R + \tau 2T^i(t+\tau)} \tag{102}$$

iii) Compute $T^{i+1}(t)$ using

$$T^{i+1}(t) = \frac{1}{2}\frac{1}{\left(2\tau F^{i+1}+1\right)}\left(2T^{i+1}(t+\tau) + R\tau\left(F^{i+1}\right)^2\right),\ T^{i+1}\left(t_f\right) = Q_f \tag{103}$$

iv) Repeat steps (ii) and (iii) until convergence is achieved.

The convergence of the above iterative procedure to the optimal solution when started with an initial stabilizing controller $F^0(t)$ is shown in [53-54].

For numerical illustration of the above iterative procedure the cost function parameters are assumed as $R=1$, $Q_f=2.5$, $t_f=5s$ and $T(t)=Q_f$. Figure 3.1 compares the solution of the iterative procedure with that of the Riccati equation. It can be observed that the adaptive critic algorithm converged to the optimal solution by the end of 6th iteration.

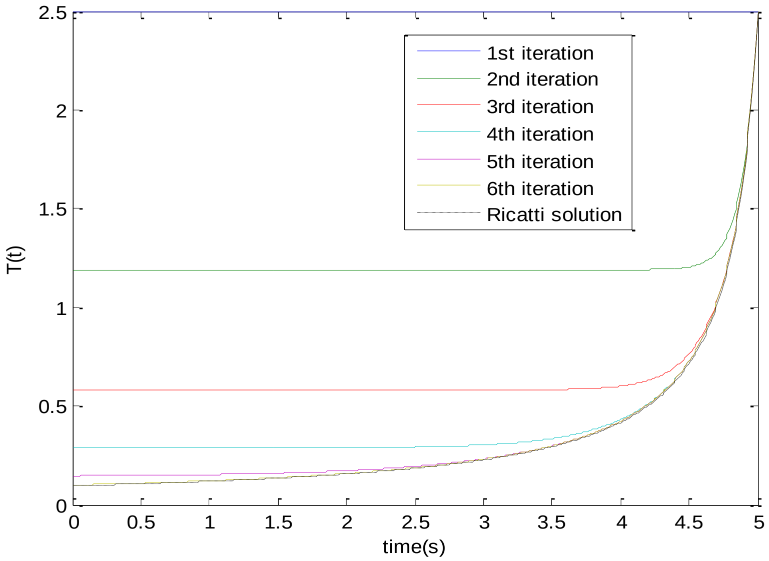


Figure 3.1. Comparison of the adaptive critic and Riccati equation solutions (Scalar case)

3.4.3. Vector Diffusion Problem. The governing equations of a class of linear systems are given by

$$dx=\left(Ax+u\right)dt+\sigma d\xi \tag{104}$$

The objective of the controller design is to minimize the following objective function

$$c(x,t) = \left\langle \left(x\left(t_f\right)\right)^T Q_f x\left(t_f\right) + \int_t^{t_f} \left(\frac{1}{2}u^T R u + x^T Q x\right) dt \right\rangle_x \tag{105}$$

For the adaptive critic scheme, the optimum cost function at iteration i is approximated as follows

$$J^i\left(x,t\right) = a^i(t) + x^T T^i\left(t\right) x \tag{106}$$

where $a^i(t) > 0$, $T^i\left(t\right) \in \mathbb{R}^{n\times n}$ is a positive symmetric matrix. The iteration process can be started with any $T^0(t) \geq 0$. For the purpose of convergence analysis $N = 1$, $\Delta t = \tau$ and $x_{N+1} = y$.

The state transition probability for the uncontrolled dynamics of the stochastic system is given by

$$\rho\left(x \to y / x,t\right) = \frac{1}{\sqrt{\det\left(\tau 2\pi\sigma\sigma^T\right)}} \exp\left(-\frac{\tau}{2}\left(\Sigma\left(t\right)\right)^T \left(\sigma\sigma^T\right)^{-1}\left(\Sigma\left(t\right)\right)\right) \tag{107}$$

where $\Sigma\left(t\right) = \left(\frac{y-x}{\tau} - Ax\right)$. The transformed cost function $\psi^{i+1}\left(x,t\right)$ can now be computed as

$$\begin{aligned}
&\psi^{i+1}\left(x,t\right) \\
&= \int dy \rho\left(y, t+\tau / x,t\right)\psi^i\left(y,t+\tau\right) \\
&= \int \frac{dy}{\sqrt{\det\left(\tau 2\pi\sigma\sigma^T\right)}} \exp\left(-\frac{\tau}{2}\left(\Sigma\left(t\right)\right)^T\left(\sigma\sigma^T\right)^{-1}\Sigma\left(t\right)\right) \exp\left(\frac{-a^i(t+\tau) - y^T T^i\left(t+\tau\right) y}{\lambda}\right)
\end{aligned} \tag{108}$$

After some algebraic manipulation Eq. (108) is written in the following form

$$\psi^{i+1}(x,t) = \frac{\phi}{\sqrt{\det\left(\tau 2\pi\sigma\sigma^T\right)}} \int dy \; \exp\left(-\frac{1}{2} y^T C_1 y + C_2^T y\right) \tag{109}$$

with

$$\begin{aligned} S_2 &= \frac{x}{\tau} + Ax \\ \phi &= \exp\left(-\frac{a^i(t+\tau)}{\lambda} - \frac{\tau}{2\lambda} S_2^T R S_2 - \frac{\tau}{\lambda} x^T Q x\right) \\ C_1 &= \left(\frac{R}{\lambda\tau} + \frac{2T^i(t+\tau)}{\lambda}\right) \\ C_2^T &= \frac{S_2^T R}{\lambda} \end{aligned} \tag{110}$$

Evaluation of Eq. (109) results in

$$\psi^{i+1}(x,t) = \frac{\phi}{\sqrt{\det\left(\tau 2\pi\sigma\sigma^T\right)}} \sqrt{\frac{(2\pi)^n}{|C_1|}} \exp\left\{\frac{1}{2} C_2^T C_1^{-1} C_2\right\} \tag{111}$$

The control u^{i+1} is computed as

$$u^{i+1} = -R^{-1} \frac{-\lambda}{\psi^{i+1}(x,t)} \frac{\partial \psi^{i+1}}{\partial x} \tag{112}$$

Evaluation of Eq. (112) leads to

$$u^{i+1} = -\frac{x 2T^i(t+\tau)}{R + \tau 2T^i(t+\tau)} = -F^{i+1} x \tag{113}$$

$$\begin{aligned} u^{i+1} &= R^{-1}\left[\tau\left\{-S_2^T R\frac{\partial S_2}{\partial x} + S_2^T C_3\frac{\partial S_2}{\partial x} - 2Q\right\}\right]x \\ &= R^{-1}\left[\frac{1}{\tau}\left\{\left(I_n+\tau A\right)^T\left(C_3-R\right)\left(I_n+\tau A\right)\right\} - 2Q\tau\right]x \\ &= -F^{i+1}x \end{aligned} \tag{114}$$

with

$$\begin{aligned} C_3 &= R\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1}R - R \\ F^{i+1} &= -R^{-1}\left[\frac{1}{\tau}\left\{\left(I_n+\tau A\right)^T\left(C_3-R\right)\left(I_n+\tau A\right)\right\} - 2Q\tau\right] \end{aligned} \tag{115}$$

The expression for control given in Eq. (114) can be further simplified as follows:

$$\begin{aligned} C_3 - R &= R\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1}R - R \\ &= R\left\{\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1} - R^{-1}\right\}R \end{aligned} \tag{116}$$

Further by using the matrix inversion lemma for $\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1}$ it can be shown that

$$C_3 - R = -R\left\{\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1}2\tau T^i\left(t+\tau\right)\right\} \tag{117}$$

Applying the matrix inversion lemma for $\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1}$ once again results in

$$C_3 - R = \left\{-I + 2\tau T^i\left(t+\tau\right)\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1}\right\}2\tau T^i\left(t+\tau\right) \tag{118}$$

Hence

$$\begin{aligned} F^{i+1} = &\frac{R^{-1}}{\tau}\left(I_n+\tau A\right)^T\left\{I_n - 2\tau T^i\left(t+\tau\right)\left(R+2\tau T^i\left(t+\tau\right)\right)^{-1}\right\}2\tau T^i\left(t+\tau\right)\left(I_n+\tau A\right) \\ &+ 2R^{-1}Q\tau \end{aligned} \tag{119}$$

To interpret the above control expression, consider the deterministic discrete dynamics of the stochastic system defined in Eq. (104),

$$\begin{aligned} x_{s+1} &= \left(I_n + \tau A\right) x_s + \tau u_s \\ &= A_d x_s + \tau u_s \end{aligned} \tag{120}$$

with $s = 0,1,2....F$. The discretized cost function is defined as

$$c^d(x_s, s) = x_F^T Q_f x_F + \sum_{j=s}^{s} \left(\frac{\tau}{2} u_j^T R u_j + \tau x^T Q x \right) = x_s^T T_{ds} x_s, \; T_{dF} = Q_f \tag{121}$$

The dynamics of the cost function parameter T_{ds} is described by the discrete Riccati equation

$$T_{ds} = 2Q\tau + 2A_d^T T_{d(s+1)} A_d - 4\tau A_d^T T_{d(s+1)} \left(R + 2\tau T_{d(s+1)}\right)^{-1} T_{d(s+1)} A_d \tag{122}$$

Comparing Eq. (119) and Eq. (122), the expressions for F^{i+1} and T_{ds} are very similar. Thus the sampling process Eqs. (107) to (111) results in a controller expression that depends on the discrete Riccati equation solution.

To compute $T^{i+1}(t)$ substitute $J^{i+1}(x,t) = a^{i+1}(t) + x^T T^{i+1} x$ and Eq. (114) in Eq. (84)that results in

$$\begin{aligned} &x^T \dot{T}^{i+1}(t) x + \dot{a}^{i+1}(t) + x^T \left(A - F^{i+1}\right)^T T^{i+1}(t) x + x^T T^{i+1}(t)\left(A - F^{i+1}\right) x \\ &\qquad + \operatorname{trace}\left(\sigma\sigma^T\right) T^{i+1}(t) + x^T Q x + x^T \frac{1}{2}\left(F^{i+1}\right)^T R F^{i+1} x = 0 \end{aligned} \tag{123}$$

Let the parameter $a^{i+1}(t)$ be updated such that

$$a^{i+1}(t)=\int_{t}^{T}\frac{1}{2}\operatorname{trace}\left(\sigma\sigma^{T}\right)T^{i+1}(\theta)\,d\theta \tag{124}$$

Substituting Eq. (124) in Eq. (123) results in

$$\dot{T}^{i+1}(t)+\left(A-F^{i+1}\right)^{T}T^{i+1}(t)+T^{i+1}(t)\left(A-F^{i+1}\right)+Q+\frac{1}{2}\left(F^{i+1}\right)^{T}RF^{i+1}=0 \tag{125}$$

Equation (125) is just the continuous time Riccati equation with the boundary condition $T\left(t_{f}\right)=Q_{f}$. For computer implementation $\dot{T}^{i+1}(t)$ is approximated as

$$\dot{T}^{i+1}(t)=\frac{T^{i+1}(t+\tau)-T^{i+1}(t)}{\tau} \tag{126}$$

Substituting Eq. (126) in Eq. (125) results in

$$\begin{aligned}&\left(-I_{n}+\tau\left(A-F^{i+1}\right)^{T}\right)T^{i+1}(t)+\tau T^{i+1}(t)\left(A-F^{i+1}\right)\\&\qquad=-\left(T^{i+1}(t+\tau)+Q\tau+\frac{\tau}{2}\left(F^{i+1}\right)^{T}RF^{i+1}\right)\end{aligned} \tag{127}$$

Equation (127) is rewritten in a compact form as

$$A_{syl}T^{i+1}(t)+T^{i+1}(t)B_{syl}=-C_{syl} \tag{128}$$

with

$$\begin{aligned}A_{syl}&=\left(-I_{n}+\tau\left(A-F^{i+1}\right)^{T}\right)\\B_{syl}&=\tau\left(A-F^{i+1}\right)\\C_{syl}&=\left(T^{i+1}(t+\tau)+Q\tau+\frac{\tau}{2}\left(F^{i+1}\right)^{T}RF^{i+1}\right)\end{aligned} \tag{129}$$

Equation (128) is a Sylvester equation and a unique solution $T^{i+1}(t)$ exists as long as A_{syl} and $-B_{syl}$ do not have any common eigenvalues. The adaptive critic iterative procedure can be summarized as follows:

v) Select a $T^0(t) \geq 0$

vi) Compute $F^{i+1}\left(t\right)$ using

$$\begin{aligned} F^{i+1} &= R^{-1}\left[\Delta_1\Delta_2 + 2Q\tau\right] \\ \Delta_1 &= \frac{1}{\tau}\left(I_n + \tau A\right)^T \left\{I_n - 2\tau T^i\left(t+\tau\right)\left(R + 2\tau T^i\left(t+\tau\right)\right)^{-1}\right\} \\ \Delta_2 &= 2\tau T^i\left(t+\tau\right)\left(I_n + \tau A\right) \end{aligned} \tag{130}$$

vii) Compute $T^{i+1}\left(t\right)$ using

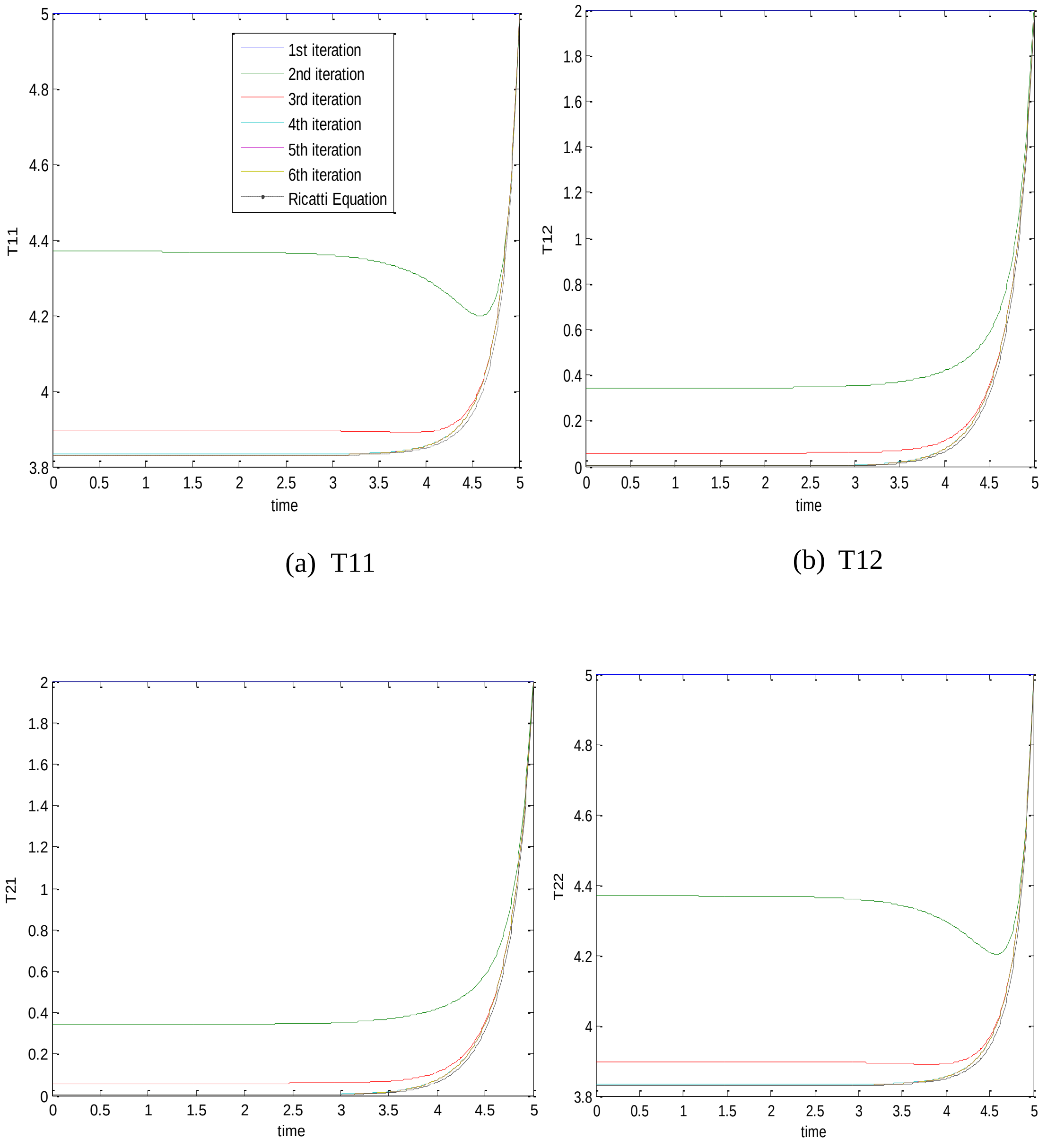
$$A_{syl}T^{i+1}\left(t\right) + T^{i+1}\left(t\right)B_{syl} = -C_{syl} \tag{131}$$

viii) Repeat steps (ii) and (iii) until convergence is achieved.

The convergence of the above iterative procedure to the optimal solution when started with an initial stabilizing controller $F^0\left(t\right)$ is shown in [53-54]. For numerical illustration of the above iterative procedure the cost function parameters are assumed as

$$A = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix},\ \text{Q=}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, R = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix},\ T(t) = \begin{bmatrix} T11 & T12 \\ T21 & T22 \end{bmatrix} \tag{132}$$

Figure 3.2 compares the solution of the iterative procedure with that of the Riccati equation. It can be observed that the adaptive critic algorithm converged to the optimal solution by the end of 6th iteration.



(a) T11

(b) T12

(b) T21

(c) T22

Figure 3.2. Comparison of the adaptive critic and Riccati equation solutions (Vector case)

3.5. IMPLEMENTATION USING NEURAL NETWORKS

For nonlinear systems, adaptive critic schemes are typically implemented using neural network models. In this Section, the cost function $J^i\left(x,t\right)$ is approximated using a multi-layer neural network model. Since the optimum cost function depends both on the current state and the time-to-go; it is approximated as

$$J^i\left(x,t\right) \approx NN^i\left(x,t_f - t\right) \tag{133}$$

The analytical computation of the $\hat{\psi}\left(x,t\right)$ using the path integral expression given in Eq. (69) is very difficult. However, numerical Monte Carlo techniques can be employed to evaluate the path integral. In this paper, the Metropolis-Hastings sampling scheme [55-56] is employed to sample trajectories as per the probability distribution given in Eq. (68). Details of the sampling scheme are given in Table 1. In numerical implementation, the path cost $g\left(\bar{y}\left(t \to t+\tau\right), u^{i+1}\left(t \to t+\tau\right)\right)$ is approximated by $S_{path}^{\Delta t}$. The neural network training procedure is given as follows:

Step 1: Generate M samples of the state vector $x \in \Omega$ randomly. For each sample of state vector choose randomly a time step $t \in \left[t_i, t_f\right]$. Then, execute the following steps.

Step 2: Generate the initial trajectory $\bar{y} = \left(x_1, x_2, ..., x_{N+1}\right)$ by propagating the state vector $x_1 = x$ from time $t_1 = t$ upto time $t_{N+1} = t + \tau$ using the following deterministic dynamics:

$$x_{k+1} = x_k + f(x_k)\Delta t + Bu_k \ \Delta t \tag{134}$$

Table 3.1. Metropolis-Hastings sampling scheme

```
    $\bar{y} = \left(x_1, x_2, ......., x_{N+1}\right)$
for $n = 1 : N_2$
    Define a Gaussian distribution centered on the directly actuated states $\left(x_1^a, x_2^a, ......., x_{N+1}^a\right)$ of the initial trajectory with variance equal to $\sigma$ .
    **Local update:**
    $\bar{y}^{'}\left(1\right) = x_1$
    for $j = 2 : N+1$
        Draw a sample $z_j^a$ from the proposal distribution.
        Find $z_j^{na}$ from the state dynamics equation given in Eq. (134).
        Compute $p = \exp\left(\frac{1}{\lambda} S_{path}^{\Delta t}\left(x_1, x_2, ......., x_{N+1}\right) - \frac{1}{\lambda} S_{path}^{\Delta t}\left(x_1, x_2, ..z_j...., x_{N+1}\right)\right)$
        If ( $p \geq 1$)
            $x_j = z_j$
        else
            $x_j = z_j$ with probability $\left(1 - p\right)$
        endif
        $\bar{y}^{'}\left(j\right) = x_j$
    end
```

Table 3.1. Metropolis-Hastings sampling scheme (cont.)

```
    Global update:
    Compute $p_1 = \exp\left(\frac{1}{\lambda} S_{path}^{\Delta t}\left(\bar{y}\right) - \frac{1}{\lambda} S_{path}^{\Delta t}\left(\bar{y}^{'}\right)\right)$
    If ( $p_1 \geq 1$ )
    $\bar{y} = \bar{y}^{'}$
    else
    $\bar{y} = \bar{y}^{'}$ with probability $\left(1 - p_1\right)$
    endif
    $\left(x_1^n, x_2^n, ......., x_{N+1}^n\right) = \bar{y}$
    $n = n + 1$
end
```

$$\text{where } u_k = -R^{-1} B^T \left(\frac{\partial NN}{\partial x}\right)_{x_k} \text{ with } k = 1,\ 2,\ .....N\,.$$

Step 3: Generate N_2 sample trajectories using the Metropolis-Hastings sampling scheme explained in Table 1.

Step 4: Compute $J^i\left(x_{N+1}^n, t_f - \left(t + \tau\right)\right)$ using the critic network.

Step 5: Compute $S_{path}^{\Delta t}\left(x_1^n, x_2^n, ..., x_{N+1}^n\right)$ for all the sample paths.

Step 6: Compute the average cost

$$\hat{J}\left(x, t\right) = \left\langle S_{path}^{\Delta t}\left(x_1^n, x_2^n, ..., x_{N+1}^n\right) + J\left(x_{N+1}^n,\ t_f - \left(t + \tau\right)\right)\right\rangle_n, n = 1,\ 2,\ 3, ...N_2 \qquad (135)$$

Step 7: Repeat steps (2) to (6) for the next sample.

Step 8: Train the critic network to minimize the following error

$$E = \frac{1}{M} \sum_{M} \left(J^{i} \left(x,t \right) - J^{i+1} \left(x,t \right) \right)^{2} \tag{136}$$

Step 9: Repeat steps (i) to (viii) until the error E reaches the desired level.

Remark 1: The proposed adaptive critic scheme belongs to the class of reinforcement learning schemes called value iteration scheme. The initial weights of the neural network can be chosen randomly. However, convergence of the proposed adaptive critic to the optimal solution depends heavily on the sampling procedure. To minimize the search space, it is recommended to start the training process with a stabilizing controller.

3.6. SIMULATION RESULTS

The proposed controller design methodology was applied to two bench mark problems.

3.6.1. Case A: Scalar Example.

The first problem considered is a diffusion problem for which analytical solution for optimal controller exists. The governing equation of the diffusion problem is given by

$$dx = udt + \sigma d\xi \tag{137}$$

where $\sigma = 0.1$. Objective of the controller design is to minimize the following cost function

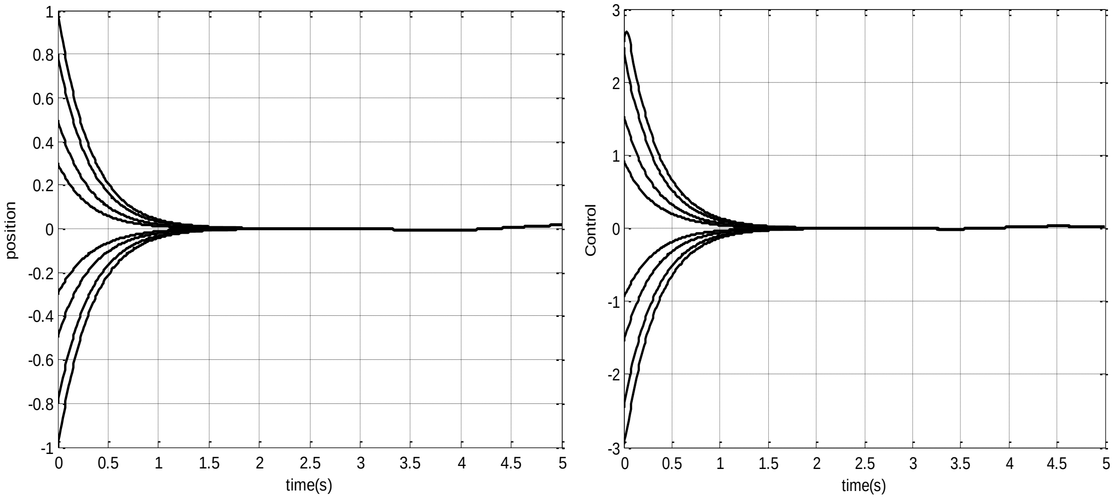
$$c(x,t) = \left\langle Q_f x\left(t_f\right)^2 + \int_{t}^{t_f} \left(\frac{1}{2} Ru^2 + Px^2 \right) dt \right\rangle_x \tag{138}$$

In this study, the cost function parameters are selected as $Q_f = 5$, $P = 5$, $R = 1$ and $t_f = 5$s . The cost function is approximated using a single hidden layer neural network with twelve neurons. The neurons are constructed using tansigmoid basis functions. The training of the critic network was done using “Matlab R2014a” neural network toolbox. The first iteration of the critic network training was carried out with an arbitrarily chosen stabilizing controller $u_{init} = -20x$. The critic network was trained for a range where $x \in [-1,1]$. Initially, the performance of the adaptive critic controller is evaluated with zero noise input. Figure 3.3 shows the performance of the adaptive critic controller for different initial conditions.

The analytical optimal control solution for this scalar case is given by

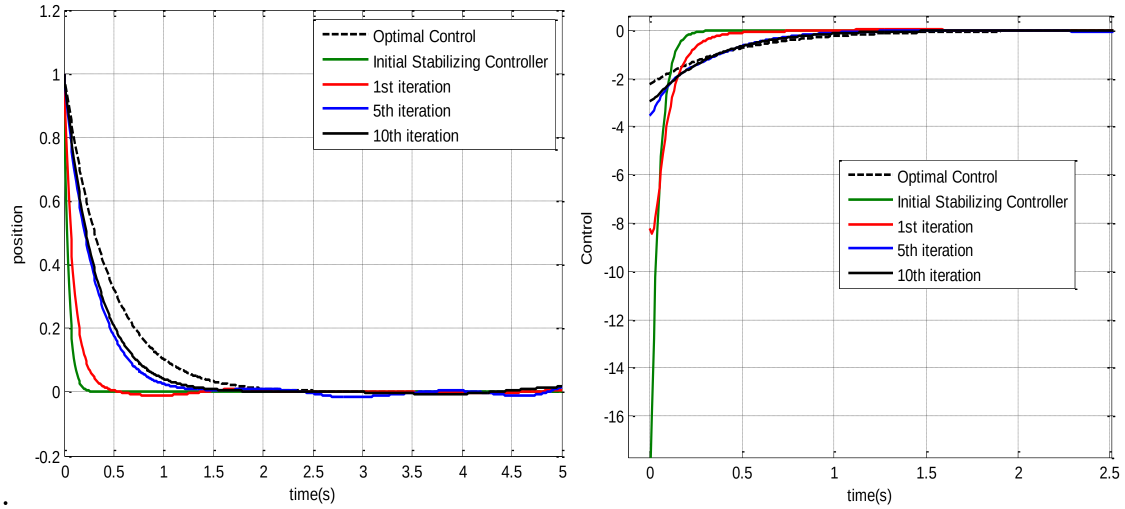
$$u_{opt} = -\sqrt{\frac{P}{R}}\frac{\gamma^2 - \dfrac{\sqrt{PR}-Q_f}{\sqrt{PR}+Q_f}}{\gamma^2 - \dfrac{\sqrt{PR}-Q_f}{\sqrt{PR}+Q_f}}x \tag{139}$$

where $\gamma = e^{\sqrt{P/R}(t_f - t)}$. Figure 3.4 shows how the training process iteratively improves the adaptive critic controller performance. It can be observed that as the number of iteration steps increases the adaptive critic solution tends toward the analytical solution Figures 3.5 and 3.6 show the performance of the adaptive critic controller in the presence of noise. It can be observed that even in the presence of noise the performance of the adaptive critic controller is very similar to that of the optimal control solution.



(a) State history (b) Control history

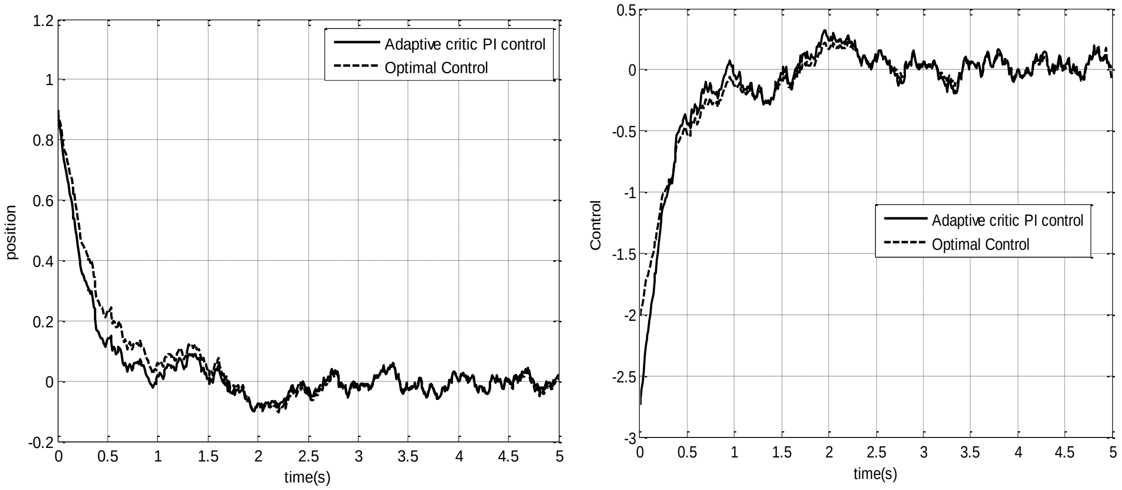
Figure 3.3. PI Adaptive critic controller performance for various initial conditions (without noise)



.

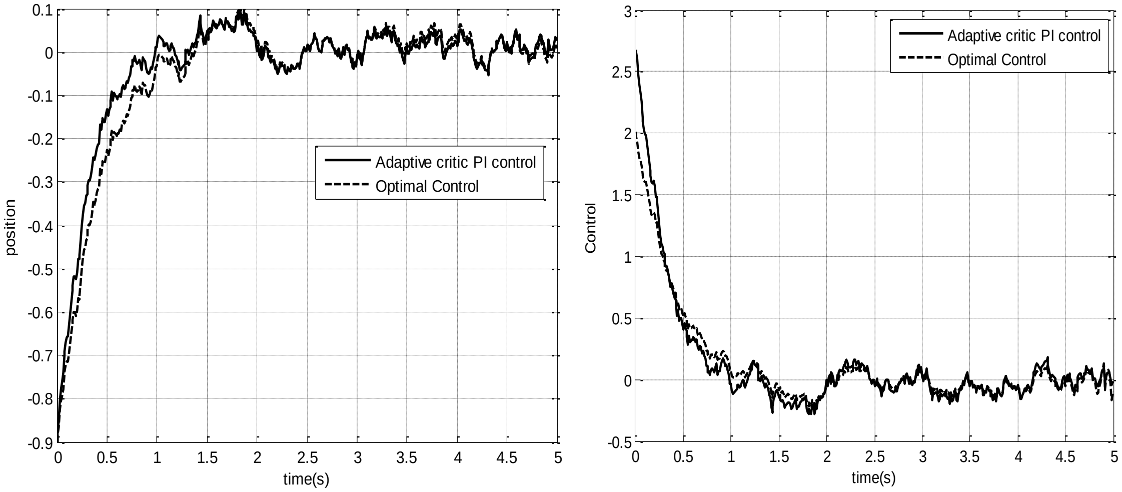
(a) State history (b) Control history

Figure 3.4. Adaptive critic controller performance at different iteration steps



(a) State history (b) Control history

Figure 3.5 PI Adaptive critic controller performance with noise $\left(x\left(t_i\right)=0.9\right)$



(a) State history (b) Control history

Figure 3.6 PI Adaptive critic controller performance with noise $\left(x\left(t_i\right)=-0.9\right)$

3.6.2. Case B: Nonlinear Vector Example. The proposed PI adaptive critic controller is now applied to a difficult Vanderpol oscillator problem. Since the path integral formulation requires that both the control and the noise act in the same subspace it is assumed that noise is present only in velocity state evolution. The governing SDEs of the system are given by

$$\begin{aligned} dx_1 &= x_2 dt \\ dx_2 &= \left(-x_1 + \varepsilon(1-x_1^2)x_2 + u\right)dt + \sigma d\xi \end{aligned} \tag{140}$$

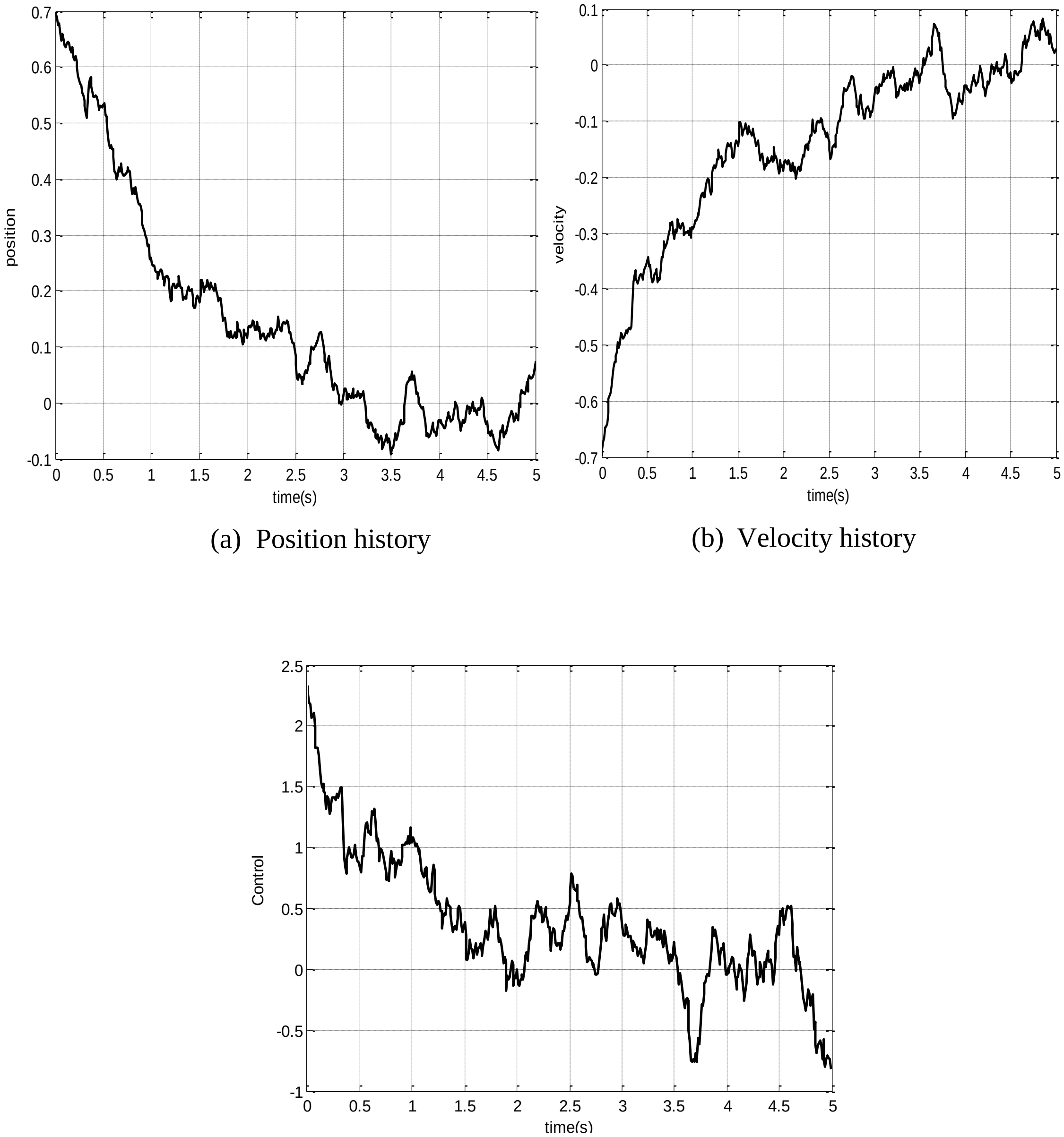
where $\varepsilon = 0.9$ and $\sigma = 0.1$. The objective is to minimize the following cost function:

$$c(x,t) = \left\langle x(t_f)^T Q_f x(t_f) + \int_t^{t_f} \left(x^T P x + \frac{1}{2} u^T R u \right) d\theta \right\rangle \tag{141}$$

The cost function parameters are

$$Q_f = 10;\ P = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix};\ R = 1 \tag{142}$$

The cost function in this case was approximated using a single hidden layer neural network with 20 neurons. The neurons are constructed using tansigmoid basis functions as in the scalar example. The weights and bias of the neural network were initialized to zero i.e. no initial stabilizing controller was used. Figures 3.7 and 3.8 show the performance of the PI adaptive critic controller for different initial conditions. It can be observed that in both the cases the adaptive critic controller was able to stabilize the oscillator.

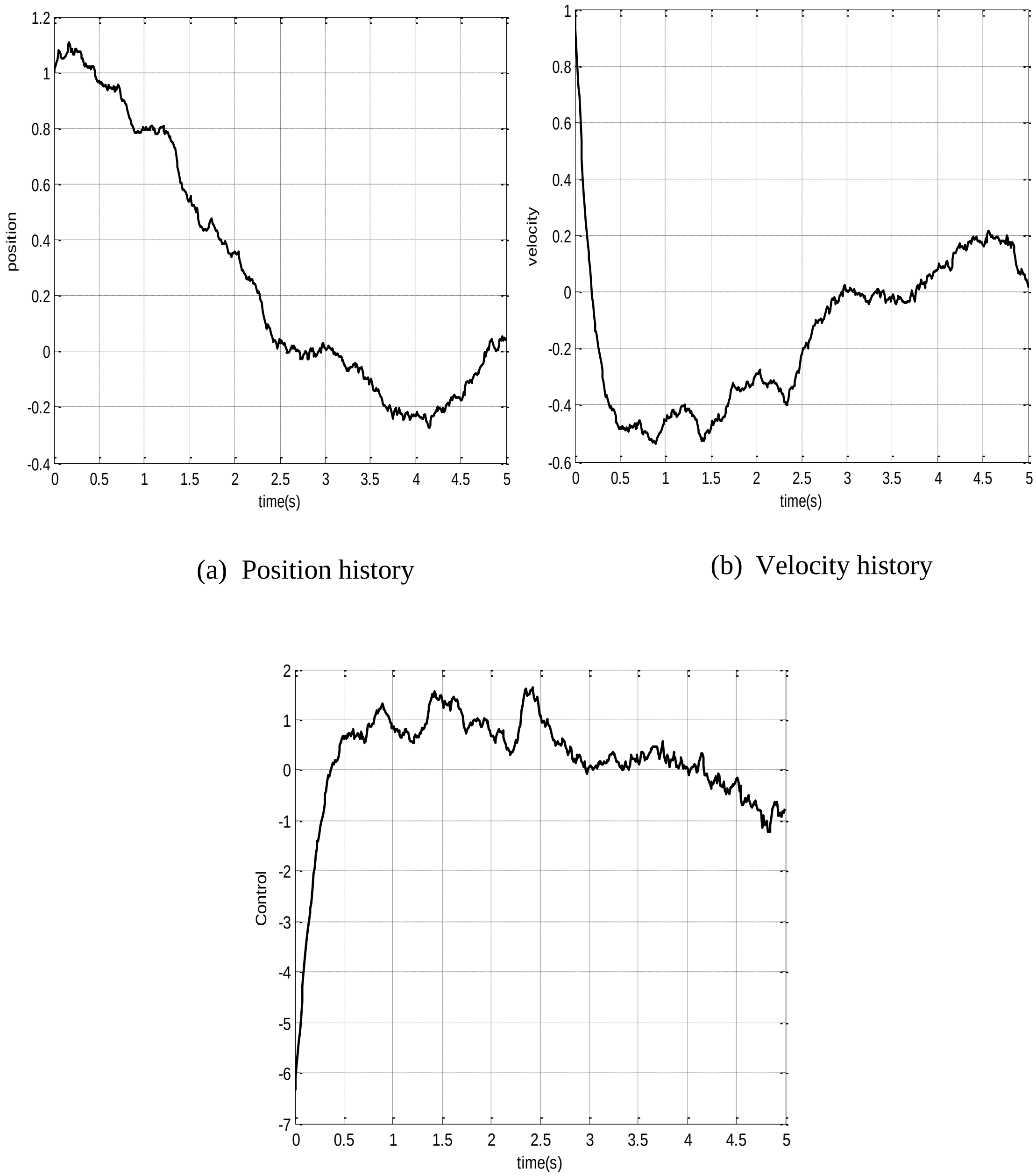


(a) Position history

(b) Velocity history

(c) Control history

Figure 3.7 Adaptive critic controller for Vanderpol oscillator problem with noise $x_1(t_i) = 0.7,\ x_2(t_i) = -0.7$



(a) Position history

(b) Velocity history

(c) Control history

Figure 3.8 Adaptive critic controller for Vanderpol oscillator problem with noise $x_1(t_i)=1,\ x_2(t_i)=1$

3.7. CONCLUSIONS

A novel adaptive critic framework based stochastic optimal controller design methodology using path integral was proposed in this study. This design paradigm combines the recently developed path integral control approach with the powerful adaptive critic design methodology and provides a robust iterative algorithm for solving stochastic optimal control problems. The novelty of the proposed adaptive critic algorithm is in using the stochastic model for state propagation and *directly* solving for the second order stochastic Hamilton-Jacobi-Bellman equation. The adaptive critic controller was tested on a scalar diffusion problem for which analytical solution already exists. The resulting performance matches the analytical solution very closely. The methodology was also applied on a difficult Vanderpol oscillator problem. The adaptive critic algorithm was able to come up with stabilizing solution for all the test cases. Since no constraining assumptions were made in its development, the path integral based adaptive critic controller is widely applicable.

4. QUANTUM INSPIRED COORDINATION MECHANISMS

4.1. INTRODUCTION

The control of Multi-agent systems is a fast developing research area. In [57] (Tomlin), an air traffic management problem is considered. Future aircrafts will have full autonomous capability and they should be able to choose their own optimal flight paths. However, when multiple aircrafts are involved a robust negotiation mechanism is required to synthesize conflict-free trajectories. Tomlin et.al [57] proposed an approach using hybrid control theory to address the above issue. Ren [58], proposed a decentralized scheme using virtual structure approach for spacecraft formation flying. Behavior based approaches are used in [59] for multi-robot teams formation control. A potential based approach was proposed in [60] for distributed cooperative control of multiple vehicle formations using structural potential functions.

Most of the control approaches available in the literature does not consider the effect of stochastic noise on the dynamics of the agents. In real world problems, the assumption that the agent’s dynamics are deterministic is rarely valid. One approach to find the optimal action when there is uncertainty in agent’s dynamics is to model the problem as a Markov decision process (MDP). The MDPs satisfy the Markov property, i.e. an agent’s transition to a new state depends only on the current state and the action choice of the agent. The application of MDP model for single agent decision-making problems is a well-studied problem. Typically dynamic programming [1] and reinforcement learning [22] approaches are employed to synthesize optimal policies for a MDP problem. The application of MDP framework for multi-agent systems is a relatively new field. Boutilier [61] showed how various single-agent decision-making mechanisms can be readily extended to multi-agent settings. However, he assumed that any new information is readily available to all agents. The extension of MDP framework to cases where the agent has incomplete information about the environment is called partially observable MDPs (POMDPs). Bernstein [62] used the decentralized framework to solve multi-agent problems where the individual agents make decisions based only on local observations and called it as decentralized partially observable Markov decision process (Dec-POMDP). Furthermore, they showed that the decentralized MDP problems are computationally complex then a centralized MDP. Guestrin et. al. [63] developed a multi-agent planning algorithm that uses system dynamics and factorized linear value function approximations to reduce the computational complexity of the multi-agent planning algorithm.

4.1.1. Game Theory. Game theory [64-66] examines situations where a player's reward depends both on his decision and the behavior of other players. The mathematical tools of game theory have found applications in a wide variety of fields. It has been applied to economics, biological sciences, social sciences etc. Some of the most commonly used game theory terminologies are introduced in this section:

Payoff function: In game theory, the payoff function assigns to each player a reward depending on his strategy and the strategy of other players.

Nash Equilibrium: The Nash equilibrium is a solution concept used in game theory to define a playing situation in which none of the players will benefit by unilaterally changing their strategies.

Best Response: The best response is the strategy that a player should play to achieve a desired outcome, given the strategies of other players.

Learning in games: Game theory as such cannot be applied for dynamic situations. Furthermore, it assumes that the players are rational. Hence, it is typically used to perform equilibrium analysis in situations where multiple intelligent agents interact. However to accommodate the non-rational behavior of agents, equilibrium concepts like Nash equilibrium can be thought of as a long-run outcome of a non-equilibrium dynamic process that models learning or adaptation of the agents. There are different learning models available in literature. They can be generally classified as individual level model or aggregate level model.

Fictitious play: The fictitious play learning model was introduced as an iterative solution procedure to find equilibrium solutions of discrete zero-sum games [67]. It was later extended as a learning model in multi-player games by Fudenberg and Levine [68]. The fictitious play is a belief based approach in which players form beliefs about the behavior of other players and act rationally with respect to these beliefs. A standard model of fictitious play is presented here. Consider a $N-$ players game. Let A_i denote the action set of the player i and $\bar{a}_i(t) = a_j \in A_i$ represent the action played by the player i at time t. Further, the empirical frequency of player i upto time t is given by

$$q_i^t(a_s) = \sum_{n=1}^{t} I(\bar{a}_i(n) == a_s) \tag{143}$$

Here, $I(A_i)$ is the indicator function and $q_i^t(a_s)$ denotes the count that how many times the player i has played action a_s upto time t. Now each player will select a best response with respect to the joint empirical frequency distribution. One of the assumptions typically used in fictitious play is that the agents make a simultaneous and independent action selection. Let $a_{-i} \in A_{-i}$ represent the action selected by any player other than agent i, then the best response of the player i is given by

$$a_i = \max_{A_i} E_{a_{-i}} \left(f\left(a_i, a_{-i}\right)\right) \tag{144}$$

where $f\left(s_i, s_j\right)$ is the pay-off function.

4.1.2. Evolutionary Game Theory. Evolutionary game theory also provides a mathematical framework to account for irrational behavior of players. Hence, it can be used to describe the time evolution of player's strategies. Evolutionary game theory (EGT) originated through the works of mathematical biologist John Maynard smith [69-70]. He adapted the methods from traditional game theory to explain the natural selection process among biological species. In a similar vein, EGT studies the interaction among different population of players and how the players might change the strategy they follow at the end of any interaction. The dynamic evolution of player's strategies is described using differential equations. A central concept in evolutionary game theory is the notion of evolutionarily stable strategies (ESS).

Evolutionarily stable strategies: ESS is a strategy which, if adopted by all the players of a population, then the natural selection process itself will not allow any competing alternative strategies to invade. Thus, ESS can be interpreted as an equilibrium strategy of the natural selection processes. The differential equations that describes how populations playing specific strategies evolve are known as the replicator dynamics.

Consider a two-player game that has a set of pure strategies $S = \left\{s_1,\ s_2,\ s_3,\ .....s_n\right\}$. Let, $f\left(s_i, s_j\right)$ denote the pay-off function and s_i is the strategy played by player '1' and s_j is the strategy played by player '2'. The proportion of players playing strategy s_i at time t is denoted by $P_i(t)$. The evolution of player's strategies over a period of time is described by the following differential equation:

$$\dot{P}_i(t) = \alpha\beta P_i(t)\left(f_i\left(t\right) - \overline{f}\left(t\right)\right) \tag{145}$$

$$f_i\left(t\right) = \sum_{j=1}^{n} P_j(t) f\left(s_i, s_j\right) \tag{146}$$

$$\overline{f}\left(t\right) = \sum_{j=1}^{n} P_j(t) f_i\left(t\right) \tag{147}$$

where α is the learning rate, $f_i(t)$ is the expected utility of player $P_i(t)$ at time t and $\overline{f}(t)$ is the expected utility for the entire population at time t. In this Section, a game theory based approach is proposed for dynamic target assignment of multiple agents moving in a stochastic environment.

4.1.3. Quantum Game Theory. There have been attempts to recast the classical game theory using quantum formalism [71]. This new field called the quantum game theory finds ways of using quantum phenomena to maximize a player's utility. Meyer [72] analyzed a coin tossing game and demonstrated that by utilizing quantum superposition a player could win with certainty against a classical player. Eisert et. al. [73] proposed a generalized quantization technique for converting a 2-player-2-strategy classical game into quantum game. Further, they showed that the dilemma of the classical prisoner's dilemma game can be resolved by entangling the states of the two players. The concept of quantizing has also been extended to multi-player classical games [74].

To demonstrate the use of quantum theory in classical games an example problem is presented here. The concept of entanglement in quantum mechanics describes the unintuitive behavior of two quantum particles prepared in a special quantum state. When these quantum particles are separated spatially and their spins are measured, the results obtained indicate that the spins of the particles are anti-correlated. Hence, it is possible to predict the state of both quantum particles by just knowing the state of one particle. Another way to interpret this phenomena is that the behavior of one particle influences the behavior of the other particle. Quantum game theory often exploits the concept of entangled states to increase the space of possible strategies and maximize the utility of quantum players. Consider the classical 2- person prisoner's dilemma game. In this game each player has 1 move and they have to choose among two pure strategies: confess and defect. The payoff matrix of this game is given in Table 4.1.

Table 4.1. Pay-off matrix in prisoner's dilemma game

	Bob: Confess	Bob: Defect
Alice: Confess	(3,3)	(0,5)
Alice: Defect	(5,0)	(1,1)

The payoff matrix is such that there is a conflict between the Nash Equilibrium solution and Pareto optimal outcome. The Nash equilibrium solution (Defect, Defect) is not a good one for players, however if both players have chosen (confess, confess) then both would have got a higher payoff of (3, 3). In the absence of communication, there is a dilemma among players in choosing the best action. Eisert et. al. [73] quantized the classical prisoner‘s dilemma game and showed that if entanglement is introduced between the player's actions, then the dilemma can be resolved. In an entangled state, the space of joint action strategies is reduced and the player's actions are non-classically correlated.

In quantum formulation, the classical strategies confess (C) and defect (D) are represented as basis vectors $\left|C\right\rangle$ and $\left|D\right\rangle$ of the two dimensional Hilbert space. The state of the game at any stage is described by a vector in the tensor product space which is spanned by the following basis vectors: $\left|CC\right\rangle$, $\left|CD\right\rangle$, $\left|DC\right\rangle$ and $\left|DD\right\rangle$. Here, the first entry refer to player 1's state and the second entry refers to player 2's state. The space of strategies available to both the players are represented by a set of 2×2 unitary matrices:

$$\hat{U}\left(\theta,\phi\right)=\begin{bmatrix} e^{i\phi}\cos\left(\theta/2\right) & \sin\left(\theta/2\right) \\ -\sin\left(\theta/2\right) & e^{-i\phi}\cos\left(\theta/2\right) \end{bmatrix} \tag{148}$$

where $0\le\theta\le\pi$ and $0\le\phi\le\pi/2$. Then, the classical strategies confess and defect are represented respectively by the following unitary matrices:

$$\hat{C}=\hat{U}\left(0,0\right)=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{149}$$

$$\hat{D}=\hat{U}\left(\pi,0\right)=\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{150}$$

The initial state of the game is given by

$$\left|\psi_0\right\rangle=\hat{\Gamma}\left|CC\right\rangle \tag{151}$$

with

$$\hat{\Gamma}=\exp\left\{i\gamma\left(\hat{D}\otimes\hat{D}\right)/2\right\} \tag{152}$$

Here, $\hat{\Gamma}$ is a unitary operator that is known to both players and $\gamma\in\left[0,\pi/2\right]$ is called the entanglement factor. Suppose $\hat{U}_1$ and $\hat{U}_2$ are the strategies of the players 1 and 2 respectively then the final state of the game is given by

$$\left|\psi_f\right\rangle = \hat{\Gamma}^{\dagger}\left(\hat{U}_1 \otimes \hat{U}_2\right)\hat{\Gamma}\left|CC\right\rangle \tag{153}$$

The unitary operator $\hat{\Gamma}^{\dagger}$ represents the measurement device. If both the players are assumed to play rationally, then from the classical prisoner's dilemma game the best strategy for both the players is $\hat{D}$. The initial and final state of the game for different values of the entanglement factor is given below:

$$\gamma = 0 \Rightarrow \begin{cases} \left|\psi_0\right\rangle = \left|CC\right\rangle \\ \left|\psi_f\right\rangle = \left|DD\right\rangle \end{cases} \tag{154}$$

$$\gamma = \frac{\pi}{2} \Rightarrow \begin{cases} \left|\psi_0\right\rangle = \frac{1}{\sqrt{2}}\left(\left|CC\right\rangle + i\left|DD\right\rangle\right) \\ \left|\psi_f\right\rangle = \left|DD\right\rangle \end{cases} \tag{155}$$

Hence, for both the separable game $\left(\gamma = 0\right)$ and the maximally entangled game $\left(\gamma = \frac{\pi}{2}\right)$ the final state of the game is similar to the classical version. However, if the players switch to a quantum strategy given below:

$$\hat{Q} = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix} \tag{156}$$

Then, the following final state results in the maximally entangled case

$$\gamma = \frac{\pi}{2} \Rightarrow \begin{cases} \left|\psi_0\right\rangle = \frac{1}{\sqrt{2}}\left(\left|CC\right\rangle + i\left|DD\right\rangle\right) \\ \left|\psi_f\right\rangle = \left|CC\right\rangle \end{cases} \tag{157}$$

Thus, in the quantum version of the prisoner's dilemma game the state $\left|DD\right\rangle$ is no longer the Nash equilibrium point. However, a new Nash equilibrium point $\left|CC\right\rangle$ appears and both the Nash equilibrium and the pareto-optimal solution coincides. It is important to note that the realization of quantized prisoner’s dilemma requires hardware that behaves quantum mechanically.

4.1.4. Quantum Decision Theory. The mathematics of quantum theory has found applications in other branches of science like cognition science. Quantum decision theory (QDT) [75, 82] is a branch of cognition science which employs the mathematical formalism of quantum theory to explain various features of human decision making. The governing belief is that humans are highly sensitive to context, sequential effects and the measurement disturbance. The human cognition models based on classic probability find it more and more difficult to accurately represent an accumulating array of complex phenomena. However, models based on quantum theory are more general and can efficiently represent the above complex phenomena. The process of quantum decision making is very different from the classical decision making process. In traditional theory of decision making, the decisions are based on a utility function. The optimal decision corresponds to the maximal expected utility. However, in QDT, the concept of an optimal decision is replaced by a probabilistic decision. Thus, QDT is emerging as an alternative approach to explain observed irrational behaviors and choices in human decision-making [75-80].

The results of the quantum version of the prisoner’s dilemma game can also be explained using QDT. Martinez [80] proposed a connection between quantum decision theory and quantum games by introducing a Hamiltonian of interaction. His version of the prisoner’s dilemma game is discussed here. In the EWL model, the deviation from the classical result is explained using the interaction between strategies of the two players. It is assumed that both the players are given an entangled state and they have to choose from a set of strategies (represented using unitary transformations) to play the game. However in QDT, the interaction is assumed to be between the player's action space and his belief space. Further, the strategic space of both the players is restricted to the subset of classical strategies i.e.

$$\hat{U}(\theta,0)=\begin{bmatrix}\cos(\theta/2) & \sin(\theta/2)\\ -\sin(\theta/2) & \cos(\theta/2)\end{bmatrix} \tag{158}$$

The decision-making process is assumed to take place in two stages. During the first stage, the state of the game evolves according to the rational considerations. Assume that the player 1 is intending to play defect i.e. $\hat{U}_1=\hat{D}$. However, he does not know the strategy of player 2. Then, the disentangled state of the game is represented by

$$|\psi_1\rangle=\left(\hat{D}\otimes U_2\right)|CC\rangle \ =\begin{bmatrix}0\\ 0\\ -\cos\left(\frac{\theta_2}{2}\right)\\ \sin\left(\frac{\theta_2}{2}\right)\end{bmatrix} \tag{159}$$

During the second stage, the state of the game evolves due to irrational considerations and this is represented by

$$|\psi_2\rangle = \exp\left\{i\gamma\left(\hat{D}\otimes\hat{D}\right)/2\right\}|\psi_1\rangle = \begin{bmatrix} i\sin\left(\frac{\gamma}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ -i\sin\left(\frac{\gamma}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ \cos\left(\frac{\gamma}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ \cos\left(\frac{\gamma}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \end{bmatrix} \tag{160}$$

Here, the entanglement factor γ models the belief of player 1 about the state of the game. From Eq. (160), the probability that the player 1 will choose confess is given by

$$\Pr^1\left(C\right) = \sin^2\left(\frac{\gamma}{2}\right) \tag{161}$$

According to Eq. (161), if $\gamma = 0$, player 1 will act rationally and choose the dominant strategy irrespective of the player 2's strategy. However, in the maximally entangled state $\left(\gamma = \pi/2\right)$ player 1 completely deviates from the rational behavior and chooses to confess. Thus, in Martinez's model of quantum prisoner's dilemma game the deviation from Nash equilibrium strategies is characterized as irrational behavior of the players. Suppose player 2 acts rationally, then $U_2 = \hat{D}$. Hence, the composite state of the game (See Eq. (160)) becomes

$$|\psi_2\rangle = i\sin\left(\frac{\gamma}{2}\right)|CC\rangle + \cos\left(\frac{\gamma}{2}\right)|DD\rangle \tag{162}$$

However, Eq. (162) is only in the mind of the player 1. Similarly, player 2 will also have a composite state of the game in its mind. Thus, QDT model of the prisoner's dilemma explains how humans deviate from normative behavior (Nash strategies). In the multi-agent task assignment problems situations may arise that require agents to deviate from rational behaviors (behaviors that will maximize their individual rewards) in order to achieve a desired collective behavior.

4.2. PROBLEM DEFINITION

The objective here is to optimally coordinate a group of agents to reach specific target points in a stochastic environment as shown in Figure 4.1.

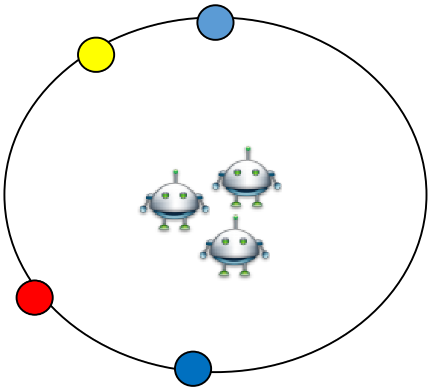


Figure 4.1. Multiple agents moving in a stochastic environment to reach unique targets

Let the dynamics of each agent be governed by the following stochastic differential equation:

$$dz_a = u_a dt + \sigma d\xi,\ a = 1, 2, 3, ..., n \tag{163}$$

where $z_a = [x_a \quad y_a]^T$ is the position of the agent with $x_a, y_a \in \mathbb{R}$, $u_a \in \mathbb{R}^2$ is the control input, a denotes the agent label and n is the number of agents. The term $d\xi \in \mathbb{R}^2$ represents the Weiner process with zero mean and variance dt and $\sigma \in \mathbb{R}^{2\times 2}$ is the variance of the noise process. The controller for each agent is designed using finite horizon stochastic optimal control theory with the objective of minimizing the following cost function:

$$C\left(z_a,t\right)=\left\langle \phi\left(z_a\left(T\right)\right)+\int_t^T\left(\frac{1}{2}u\left(\theta\right)^T Ru\left(\theta\right)\right)d\theta\right\rangle \tag{164}$$

Different controller design methodologies can be utilized to synthesize a variety of stabilizing controllers. In Eq. (164), T is the final time, $R\in\mathbb{R}^{2\times2}$ is a control weighting matrix and $\phi\left(x_a\left(T\right)\right)$ is the end cost. Suppose the objective here is that the agent should reach a target μ_j at final time T then, it is represented as

$$\phi\left(z_a\left(T\right)\right)=\frac{\alpha}{2}\left\|z_a\left(T\right)-\mu_j\right\|^2 \tag{165}$$

where $\alpha\in\mathbb{R}>0$ is a scalar constant. Let u_j^* be the optimal control, then from finite-horizon stochastic optimal control theory the equation for optimal controller is given by

$$u_j^*=\frac{\mu_j-z_a}{T-t+R/\alpha} \tag{166}$$

However, our original objective is to coordinate multiple agents to reach specific target points. Let m be the number of targets. We will assume that $n=m$ and each target is assigned to at least one agent. Then, the overall objective is to minimize the following cost function:

$$V(z,t)=\sum_{a=1}^{n}C\left(z_a,t\right) \tag{167}$$

where $z=\left[z_1^T \quad z_2^T\ldots \quad z_n^T\right]^T$.

4.3. SOLUTION APPROACHES

To solve the above problem consider the case in which there is only one agent and it has the choice of reaching any one of the target points, then this objective can be expressed in terms of the following end cost function

$$\phi\left(z_a\left(T\right)\right)=-\lambda\log\left(\sum_{j=1}^{n}\exp\left(-\frac{\alpha}{2\lambda}\left\|z_a\left(T\right)-\mu_j\right\|^2\right)\right) \tag{168}$$

Here, $\lambda\in\mathbb{R}>0$ is a scalar constant and it is defined as follows

$$\lambda R^{-1}=\sigma\sigma^T \tag{169}$$

Then, the expression for optimal controller can be expressed in terms of single-agent, single target optimal controllers

$$u^*=\sum_{j=1}^{m}p_a\left(\mu_j/z_a,t\right)u_j^* \tag{170}$$

where

$$p_a\left(\mu_j/z_a,t\right)=\frac{\exp\left(-\frac{\left\|z_a-\mu_j\right\|^2}{2\sigma^2\left(T-t+R/\alpha\right)}\right)}{\sum_{j=1}^{n}\exp\left(-\frac{\left\|z_a-\mu_j\right\|^2}{2\sigma^2\left(T-t+R/\alpha\right)}\right)} \tag{171}$$

The expression in Eq. (170) relates the single-agent, single-target optimal controllers with single-agent, multiple-target optimal controllers. The solution approaches proposed for the original problem (multiple agent, multiple targets) is inspired by the expression given in Eq. (170).

Three different approaches are discussed to address the multi-agent, multi-target problem. The first approach uses the fictitious play learning model as a negotiation mechanism to drive the agents to unique targets. This approach is inspired by the game-theoretical approach suggested in [83] to solve a static target assignment problem. The second and third approaches are developed using the entanglement phenomena introduced in section 1.

4.4. GAME THEORY BASED COORDINATION MECHANISM

In this approach, it is assumed that the agents are playing a game in which, they have to choose a target that will maximize their individual utility. The utility functions will depend both on the assignment profile and the time. Typically in game theory only static utility functions are used. To utilize the game theory approach in the dynamic target assignment problem, it is assumed that at every time instant, a static game is played and the agent has to choose one of the targets. Initially, no agent is aware of the strategies played by other agents. As time progresses, they learn the strategies of other agents and choose an action that will maximize an expected utility function. The learning process is modelled using the fictitious play approach. In fictitious play, each agent models the behavior of every other agent by keeping track of their actions at every time instant. An empirical probability distribution is then derived using the above information and it is used to compute the expected utility function.

To use the game theory approach, utility functions are defined for each action of the agent. Furthermore, best action derived using the above utility functions should result in the desired collective behavior i.e. each agent reaches a unique target at the final time

T with minimum the cost (see Eq.(167)). Taking into consideration the above concerns the following approach is proposed:

At every time instant t, an assignment probability vector that depends on the agent's target choice is defined for all the targets

$$p\left(s,t;\mu_j\right)=\left[p_{1\mu_j}\left(s,t\right) \quad p_{2\mu_j}\left(s,t\right)\ldots \quad p_{n\mu_j}\left(s,t\right)\right] \tag{172}$$

Here, $p_{1\mu_1}$ is the probability that agent 1 will be assigned target μ_1, $p_{2\mu_1}$ is the probability that agent 2 will be assigned target μ_1 and so on. Let, $s=\left[s_1 \quad s_2 \quad .... \quad s_n\right]$ is the target assignment profile i.e. s_1 is the target chosen by agent 1, s_2 is the target chosen by agent 2 and so on. Then, the utility values $p_{a\mu_1}$ for any agent a is obtained as follows

```
if $s_a = \mu_j$
$p_{a\mu_j} = p_a\left(\mu_j / z_a, t\right)$
else                                                    (173)
$p_{a\mu_j} = 0$
end
```

The utility vector depends on the probability vector $p_a\left(\mu_j / z_a, t\right)$ given in Eq. (171). In this way, the agent with the highest probability of reaching target μ_j gets assigned to it. By using the utility vector defined in Eq. (172), a target utility function is derived as follows

$$U_{\mu_j}\left(s,t\right)=V_{\mu_j}\delta\left(\beta-n_{\mu_j}\left(s\right)\right)\left(\sum_{a=1}^{n}p_{a\mu_j}\right) \tag{174}$$

Here, $V_{\mu_j} > 0$ is the target value and δ is the Dirac delta function. Further, $n_{\mu_j}(s)$ is the number of vehicles with $p_{a\mu_j} > 0$ and β is the desired number of targets that need to be assigned to target μ_j .The utility, any agent k will receive for choosing a target μ_j is defined in the following way

$$\begin{aligned} U_k(s,t) &= U_{\mu_j}(s,t) - U_{\mu_j}(s/s_k,t) \\ &= V_{\mu_j}\delta\Big(\beta - n_{\mu_j}(s)\Big)\left(\sum_{a=1}^{n} p_{a\mu_j}\right) - V_{\mu_j}\delta\Big(\beta - n_{T_j}(s/s_k)\Big)\left(\sum_{a=1}^{n} p_{a\mu_j}\right) \\ &= V_{\mu_j}\left(\delta\Big(\beta - n_{\mu_j}(s)\Big)p_{a\mu_j} - \delta\Big(\beta - n_{\mu_j}(s/s_k)\Big)p_{l\mu_j}\right) \end{aligned} \quad (175)$$

where l is the agent other than agent k that chose target μ_j and s/s_k denotes the assignment profile in which agent k is not assigned any target.

As mentioned earlier, the fictitious play approach requires that each agent should keep track of the actions selected by other agents at every time instant. This aids in creating an empirical probability distribution (See Eq. (143)) about the action selection behavior of other agents. Then, each agent chooses an action that maximizes the following expected utility function:

$$s_a = \arg\max_{\mu_j} \mathop{\mathrm{E}}_{s/s_a}\Big(U_k\big(\mu_j, s/s_a, t\big)\Big) \quad (176)$$

Note that, in the above equation the expectation value is computed using the empirical probability distribution. The expectation is taken over all possible target assignment profiles and this makes the above step computationally time consuming. Then a target probability is defined using the following relations:

$$p_a^N\big(\mu_j / z_a, t\big) = 1;\ p_a^N\big(\mu_{-j} / z_a, t\big) = 0 \quad (177)$$

Then, the controller value is computed using Eq. (170)

$$u = \sum_{j=1}^{m} p_a^N\left(\mu_j / z_a, t\right) u_j^* \tag{178}$$

4.5. ENTANGLEMENT BASED COORDINATION MECHANISM

4.5.1. Approach 1.

The proposed approach is inspired from the Martinez's prisoner's dilemma QDT model. In this approach, each agent has m composite state representations of its intentions with each one corresponding to the intention of choosing a particular target.

$$\begin{aligned}
\left|\psi_a^1\right\rangle &= i\sqrt{p_a^N\left(\mu_1 / z_a, t\right)}\left|\mu_1, \sim\mu_1\right\rangle + \sqrt{1 - p_a^N\left(\mu_1 / z_a, t\right)}\left|\sim\mu_1, \mu_1\right\rangle \\
\left|\psi_a^2\right\rangle &= i\sqrt{p_a^N\left(\mu_2 / z_a, t\right)}\left|\mu_2, \sim\mu_2\right\rangle + \sqrt{1 - p_a^N\left(\mu_2 / z_a, t\right)}\left|\sim\mu_2, \mu_2\right\rangle \\
&. \\
\left|\psi_a^m\right\rangle &= i\sqrt{p_a^N\left(\mu_m / z_a, t\right)}\left|\mu_m, \sim\mu_m\right\rangle + \sqrt{1 - p_a^N\left(\mu_m / z_a, t\right)}\left|\sim\mu_m, \mu_m\right\rangle
\end{aligned} \tag{179}$$

Here, $\sim\mu_j$ represents choosing a target other than μ_j. To rewrite the above set of equations in a form similar to that given in Eq. (162), define $p_a^N\left(\mu_j / z_a, t\right) = \sin^2\left(\frac{\gamma_a^{\mu_j}(t)}{2}\right)$. Here, the parameter $\gamma_a^{\mu_j}(t)$ aids agent a in modelling the collective behavior of other agents. It is assumed that the agent's decision-making process evolves in two stages.

i) In the first stage, the agents act in a self-interested way and assign a probability distribution to their intentions. One way of assigning the probability distribution is to utilize the single-agent, multi-target probability distribution presented in Eq. (171).
ii) In the second stage, each agent communicates its intention probability distributions to other agents. The communication overhead can be eliminated if each agent can deduce

the intention probability distribution of other agents on its own (for example: use the position information of other agents). Then, every agent updates its entanglement model given in Eq. (179) and uses this to compute the controller. The step-by-step procedure for the second stage is given below:

a) At every time instant t assign each agent a unique target. This assignment depends on $p_a\left(\mu_j / z_a, t\right)$. For example, target μ_1 is assigned an agent with the highest $p_a\left(\mu_1 / z_a, t\right)$.Remove this agent from the agent list. For target μ_2, assign from the updated agent list, the agent with the highest $p_a\left(\mu_2 / z_a, t\right)$ and so on. Each agent can perform this computation independently. However, the target assignment in this step is performed myopically. The final target each agent will reach depends on $p_a^N\left(\mu_j / z_a, t\right)$

b) Based on the above assignment, update the probability distribution $p_a^N\left(\mu_j / z_a, t\right)$. Since $p_a^N(.)$ is a probability distribution, it should obey the normalization condition:

$$\sum_{j=1}^{n} p_a^N\left(\mu_j / z_a, t\right) = 1 \Rightarrow \sum_{j=1}^{n} \sin^2\left(\frac{\gamma_a^{\mu_j}(t)}{2}\right) = 1 \tag{180}$$

Further, we require that each agent should reach unique targets, hence the following conditions are required to be satisfied at the final time

$$\sum_{a=1}^{n} p_a^N\left(\mu_1 / z_a, T\right) = 1,\ \sum_{a=1}^{n} p_a^N\left(\mu_2 / z_a, T\right) = 1, \ldots\ldots\ldots \sum_{a=1}^{n} p_a^N\left(\mu_m / z_a, T\right) = 1 \tag{181}$$

c) Compute the controller using the following equation

$$u = \sum_{j=1}^{m} p_a^N\left(\mu_j / z_a, t\right) u_j^* \tag{182}$$

Updation procedure for $p_a^N\left(\mu_j / z_a, t\right)$**:** The entanglement factors are updated using a differential equation. Suppose agent 'a' is assigned target μ_j, then the entanglement factor $\gamma_a^{\mu_j}$ is updated using a differential equation of the following form:

$$\dot{\gamma}_a^{\mu_j}\left(t\right) = p_a\left(./ z_a, t\right)^T\ p_{\sim a}^{mean}\left(./ z_{\sim a}, t\right) + K_g\left(\pi - \gamma_a^{\mu_j}\right),\ p_a^N\left(\mu_j / z_a, t_0\right) = \frac{1}{m}$$

(183)

where $p_{\sim a}^{mean}\left(./ z_{\sim a}, t\right) = \frac{1}{m-1}\sum_{\sim a} p_a\left(./ z_{\sim a}, t\right)$. The entanglement factor is constrained to lie within the range $\gamma_a^{\mu_j}\left(t\right) \in \left[0, \pi\right]$. In Eq. (184), the product term $p_a\left(./ z_a, t\right)^T\ p_{\sim a}^{mean}\left(./ z_{\sim a}, t\right)$ will have non-zero value whenever more than one agent compete for the same target. Further, the design parameter K_g ensures that $p_a^N\left(\mu_j / z_a, t\right)$ will increase even when the product term is zero. This parameter can be kept as a constant or inversely varied with respect to $\left(T - t\right)$

$$K_g \propto \frac{1}{T - t + c} \tag{185}$$

To maintain the normalization condition given in Eq. (180) the other entanglement factors $\gamma_a^{\sim\mu_j}$ are varied as follows:

$$\dot{\gamma}_a^{\sim\mu_j} = -\frac{\dot{\gamma}_a^{\mu_j}\sin\left(\gamma_a^{\mu_j}\left(t\right)\right)}{\sum_{k=1,\sim j}^{m}\sin\left(\gamma_a^{\mu_k}\left(t\right)\right)} \tag{186}$$

Since each agent synthesize their own composite representation of the system state, the agents are not exactly entangled in the quantum mechanics sense. That is, the actions of the agents are not naturally anti-correlated. Hence, the assignment procedure and the design parameter K_g both play a major role in ensuring that the agent's reach unique targets at the final time T. In the next section, another approach that directly uses the entanglement phenomenon in the quantum mechanics sense is proposed. This approach does not require any assignment procedure.

4.5.2. Approach 2. In approach 2 also, each agent has a set of equations of the form presented in Eq. (179). However, $p_a^N\left(\mu_j / z_a, t\right)$ is updated in a different way than that given in Eq. (183). A replicator dynamics model is used to describe the time evolution of $p_a^N\left(\mu_j / z_a, t\right)$.

$$\dot{p}_a^N\left(\mu_j / z_a, t\right) = p_a^N\left(\mu_j / z_a, t\right)\left(\chi\left(\mu_j, a\right) - \sum_{k=1}^{n} p_k^N\left(\mu_j / z_k, t\right)\chi\left(\mu_j, k\right)\right) \quad (187)$$

Here, $\chi\left(\mu_j, a\right)$ is the utility of agent a with respect to target μ_j. In approach 1, Eqs. (183) to (186) preserve the normalization condition given in Eq. (180). However, in approach 2, Eq. (187) preserves the following normalization condition:

$$\sum_{k=1}^{n} p_k^N\left(\mu_j / z_k, t\right) = 1 \quad (188)$$

To implement Eq. (187) in a decentralized approach, each agent should know the utilities every other agent receives. The target assignment mechanism for approach 2 is demonstrated using an example. Consider a problem in which three agents need to reach three unique targets at the final time. The entanglement equations of agents are given by,

Agent 1:

$$\left|\psi_1^1\right\rangle = i\sqrt{p_1^N\left(\mu_1/z_1,t\right)}\left|\mu_1,\sim\mu_1\right\rangle + \sqrt{1-p_1^N\left(\mu_1/z_1,t\right)}\left|\sim\mu_1,\mu_1\right\rangle \tag{189}$$

$$\left|\psi_1^2\right\rangle = i\sqrt{p_1^N\left(\mu_2/z_1,t\right)}\left|\mu_2,\sim\mu_2\right\rangle + \sqrt{1-p_1^N\left(\mu_2/z_1,t\right)}\left|\sim\mu_2,\mu_2\right\rangle \tag{190}$$

Agent 2:

$$\left|\psi_2^1\right\rangle = i\sqrt{p_2^N\left(\mu_1/z_2,t\right)}\left|\mu_1,\sim\mu_1\right\rangle + \sqrt{1-p_2^N\left(\mu_1/z_2,t\right)}\left|\sim\mu_1,\mu_1\right\rangle \tag{191}$$

$$\left|\psi_2^2\right\rangle = i\sqrt{p_2^N\left(\mu_2/z_2,t\right)}\left|\mu_2,\sim\mu_2\right\rangle + \sqrt{1-p_2^N\left(\mu_2/z_2,t\right)}\left|\sim\mu_2,\mu_2\right\rangle \tag{192}$$

Agent 3:

$$\left|\psi_3^1\right\rangle = i\sqrt{p_3^N\left(\mu_1/z_3,t\right)}\left|\mu_1,\sim\mu_1\right\rangle + \sqrt{1-p_3^N\left(\mu_1/z_3,t\right)}\left|\sim\mu_1,\mu_1\right\rangle \tag{193}$$

$$\left|\psi_3^2\right\rangle = i\sqrt{p_3^N\left(\mu_2/z_3,t\right)}\left|\mu_2,\sim\mu_2\right\rangle + \sqrt{1-p_3^N\left(\mu_2/z_3,t\right)}\left|\sim\mu_2,\mu_2\right\rangle \tag{194}$$

Furthermore,

$$\sum_{k=1}^{3} p_k^N\left(\mu_1/z_k,t\right) = 1\,;\ \sum_{k=1}^{3} p_k^N\left(\mu_2/z_k,t\right) = 1 \tag{195}$$

Agent 1 has three possible actions: i) to choose target μ_1 ii) to choose target μ_2 or iii) to choose target μ_3. Equations (189) and (190) are interpreted in the following way:

$\left|\mu_1,\sim\mu_1\right\rangle$ - Only Agent 1 chooses μ_1.

$\left|\sim\mu_1,\mu_1\right\rangle$ - Agent 1 chooses μ_2 or μ_3 and one of the other agents chooses μ_1

$\left|\mu_2,\sim\mu_2\right\rangle$ - Only Agent 1 chooses μ_2.

$\left|\sim\mu_2,\mu_2\right\rangle$ - Agent 1 chooses μ_3 and one of the other agents chooses μ_2

The above interpretation implicitly represent the preference order of agent 1. It prefers the three targets in the following order: μ_1, μ_2 and μ_3. This can be represented by assigning a probability distribution over agent 1's choices

$$\begin{aligned} &\Pr_1(t) = \left[p_1^N(\mu_1 / z_1, t) \quad \Delta_1 \quad \Delta_2 \right]^T \\ &\Delta_1 = \left(1 - p_1^N(\mu_1 / z_1, t)\right) p_1^N(\mu_2 / z_1, t); \\ &\Delta_2 = \left(1 - p_1^N(\mu_1 / z_1, t)\right)\left(1 - p_1^N(\mu_2 / z_1, t)\right) \end{aligned} \tag{196}$$

Then, agent 1 randomly chooses an action using Eq. (196). This information is communicated to agent 2. Suppose agent 1 chooses μ_3, it means that the state $\left|\sim \mu_2, \mu_2\right\rangle$ in Eq. (190) is realized. For agent 2, this is equivalent to realizing one of the following states in Eq. (191): $\left|\mu_1, \sim \mu_1\right\rangle$, $\left|\mu_2, \sim \mu_2\right\rangle$ Hence, agent 2 can either choose μ_1 or μ_2. A probability distribution is assigned to agent 2's action choices using Eq. (191)

$$\Pr_2(t) = \left[p_2^N(\mu_1 / z_2, t) \quad 1 - p_2^N(\mu_1 / z_2, t) \quad 0 \right]^T \tag{197}$$

Suppose agent 2 chooses μ_1, then the state $\left|\mu_1, \sim \mu_1\right\rangle$ in Eq. (191) is realized. For agent 3, this is equivalent to realizing state $\left|\sim \mu_1, \mu_1\right\rangle$ in Eq. (193). Hence, agent 3 can only choose μ_2. Accordingly agent 3's probability distribution vector is given by

$$\Pr_3(t) = [0 \quad 1 \quad 0]^T \tag{198}$$

The above selection process mimics the entanglement phenomenon in quantum mechanics sense. If a quantum computer is available, then the agents need not have to communicate the current state information to each other. Table 4.2 lists the possible ways Eq. (197) and Eq. (198) can vary depending on agent 1's choice. The described selection process transfers information from agent 1 to agent 2 and agent 2 to agent 3. Any other preferred order can also be used.

Table 4.2. Probability distribution of agent 2 and agent 3

Agent 1’s choice	Agent 2’s Probability vector	Agent 2’s choice	Agent 3’s Probability vector
μ_1	$\Pr_2(t)=\left[0 \quad p_2^N(\mu_2/z_2,t) \quad 1-p_2^N(\mu_2/z_2,t)\right]^T$	μ_2	$\Pr_3(t)=\left[0 \quad 0 \quad 1\right]^T$
		μ_3	$\Pr_3(t)=\left[0 \quad 1 \quad 0\right]^T$
μ_2	$\Pr_2(t)=\left[p_2^N(\mu_1/z_2,t) \quad 0 \quad 1-p_2^N(\mu_1/z_2,t)\right]^T$	μ_1	$\Pr_3(t)=\left[0 \quad 0 \quad 1\right]^T$
		μ_3	$\Pr_3(t)=\left[1 \quad 0 \quad 0\right]^T$
μ_3	$\Pr_2(t)=\left[p_2^N(\mu_1/z_2,t) \quad 0 \quad 1-p_2^N(\mu_1/z_2,t)\right]^T$	μ_1	$\Pr_3(t)=\left[0 \quad 1 \quad 0\right]^T$
		μ_2	$\Pr_3(t)=\left[1 \quad 0 \quad 0\right]^T$

4.6. SIMULATION RESULTS

Simulations were performed to compare the performance of all the three approaches. It was assumed that all the agents have same dynamics. The noise variance σ of the agent’s dynamics (See Eq.(163)) is taken as 0.1. Simulations were performed for the following scenarios

i) Five agents with all the agents starting from the origin.

ii) Five agents with two of the agents starting from origin and three other agents starting from $(0,12)$.

iii) Ten agents with all the agents starting from the origin.

iv) Twenty agents with all the agents starting from the origin.

In all the above cases, the final target points lie at equal intervals on a semi-circle of radius 6 units. The final time T was fixed at 10 seconds. The cost function parameters were chosen as $R=1$ and $\alpha=100$. Comparisons were made by using the same noise realization for all the three approaches. 100 test cases were simulated for each of the approaches to study the noise effect. The design parameters used for simulation are listed below (Table 4.3):

Table 4.3. Design parameter values

Classical Game theory approach	$V_{\mu_j}=1,\ \beta=1$
Entanglement Approach 1	$K_g=10$
Entanglement Approach 2	$\chi(\mu_j,a)=10+10p_a(\mu_j/z_a,t)$ $\chi(\mu_{\sim j},a)=-10p_a(\mu_j/z_a,t)$

Case I: Figure 4.2 shows the results obtained using classical game theory approach for one of the sample runs. The total cost accrued in each of the sample runs for all the three approaches are compared in Figure 4.3. It can be observed that the differences in performance between the three approaches is very minimal. Table 4.4 lists the maximum, minimum, average and standard deviation of the total costs obtained with each of the approaches. The entanglement method using approach 1 performs slightly better compared to the other approaches. The time taken for the simulation to run with each of the approaches is also listed in Table 4.4. Classical game theory based approach takes the maximum time as compared to the entanglement based approaches. A main contributing factor to the computational time is that each agent calculates the maximum expected utility by going through all the possible unique assignment profiles. For case I, this amounts to 120 unique assignment profiles. Among the entanglement methods, second approach takes more computational time than the first approach. This is due to the fact that the second approach uses a random number generator for the action selection process.

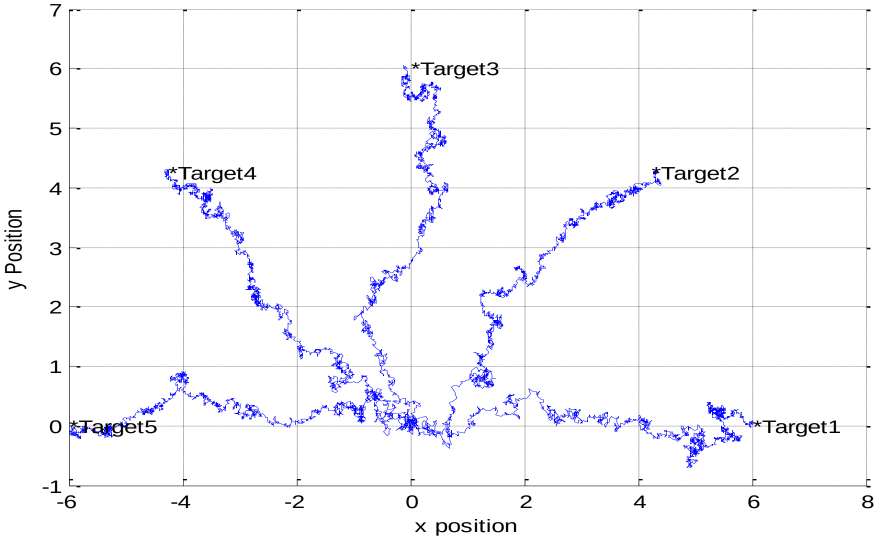


Figure 4.2. Classical game theory approach (sample run (case I))

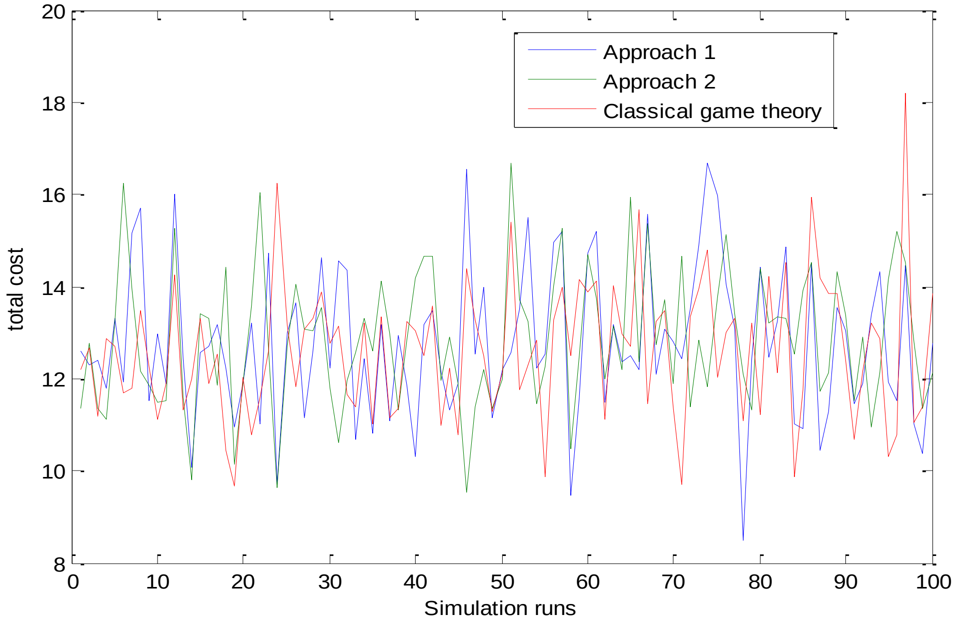


Figure 4.3. Total cost comparison for case I

Table 4.4. Performance comparison for case I

(100 cases)	Entanglement method (Approach 1)	Entanglement method (Approach 2)	Fictitious play
Minimum	8.4798	9.5440	9.6653
Maximum	16.6769	16.6859	18.1994
Average	12.7734	12.8617	12.5975
Standard deviation	1.6165	1.4742	1.4752
Time (s)	141.7714	216.0761	337.4022

Case II: In case II, the agents start from different initial positions. Figure 4.4 shows the results obtained using the entanglement method (Approach 1) during one of the sample runs. The total cost accrued during each sample run is compared in Figure 4.5. The entanglement method (Approach II) performs much better than the other two approaches. Table 4.5 lists the performance statistics obtained for case II. Simulation results indicate that Approach I is sensitive to initial conditions. Hence, the effect of the design parameter

K_g on the performance of Approach I was studied. Table 4.6 shows the performance statics for different values of K_g. Variations in performance indicate that the designer need to tune the value of K_g to obtain desired performance. However, for both case I and case II, the second entanglement approach performed consistently better than that of the classical game theory approach.

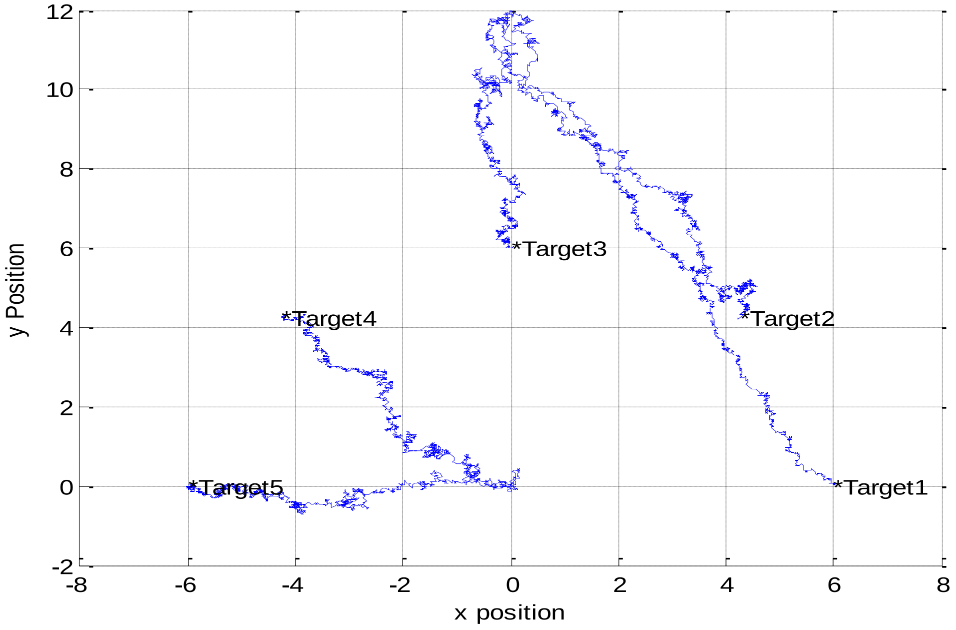


Figure 4.4. Entanglement (Approach I) method (sample run (case II))

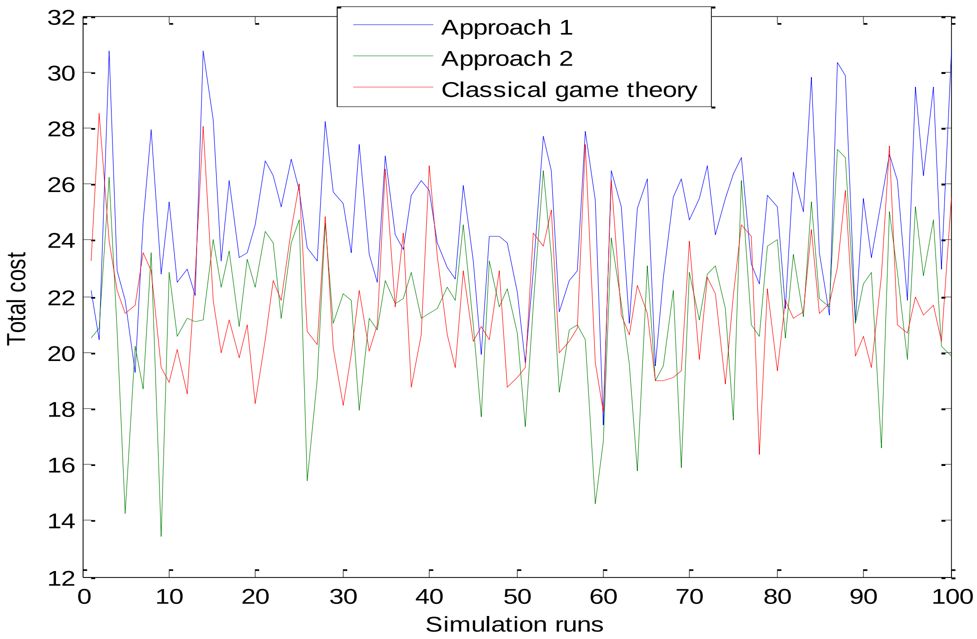


Figure 4.5. Total cost comparison for case II

Table 4.5. Performance comparison for case II

(100 cases)	**Entanglement method (Approach 1)**	**Entanglement method (Approach 2)**	**Fictitious play**
Minimum	17.4118	13.4201	16.3560
Maximum	30.7587	27.2374	28.5442
Average	24.7072	21.5058	21.7343
Standard deviation	2.6826	2.7119	2.4431
Time (s)	150.197935	216.0761	333.3727

Table 4.6. Design parameter effect on the performance of Approach I

(100 cases) Entanglement method(Approach 1)	$k_g = 5$	$k_g = 10$	$k_g = 10/(T-t+c)$
Minimum	17.4240	17.4118	17.8537
Maximum	34.2371	30.7587	48.7480
Average	24.8178	24.7072	26.9622
Standard deviation	2.8478	2.6826	4.3496
Time	151.7901	150.1979	151.2624

Case III and Case IV: For case III and case IV simulation, the number of agents were increased to ten and twenty respectively. The classical game theory based approach cannot be employed for both these cases, since the number of unique target assignment profiles exceeds 300000. However, both the entanglement based approaches can be easily utilized to coordinate the agents. Figures 4.6 and 4.7 shows the results obtained during two of the sample runs using Approach II. Table 4.7 and 4.8 compares the performance obtained for cases I, III and IV with Approach I and Approach II, respectively. It can be observed that for both the approaches, computational time increases almost linearly with increase in number of agents. However, Approach II performs consistently better than Approach I in terms of maximum cost, minimum cost and standard deviation in costs. As demonstrated earlier, the design parameter k_g, played a crucial role in the performance of Approach I. For the 20 agents case, the value $K_g = 10$ resulted in very poor performance. Hence, this gain was increased to 30 to obtain the performance listed in Table 4.7.

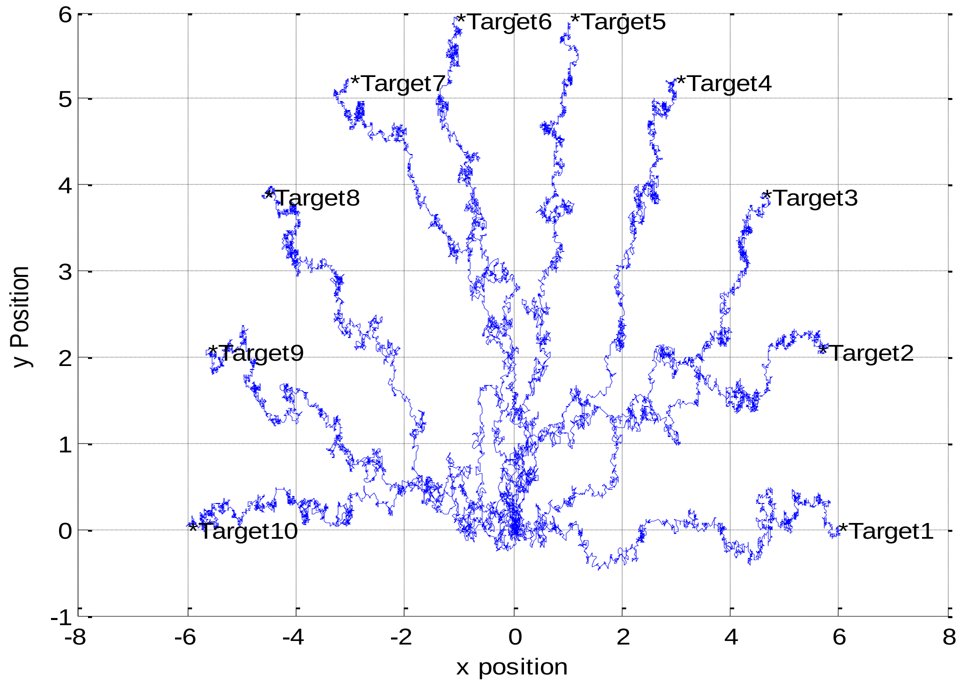


Figure 4.6 Entanglement (Approach II) with 10 agents

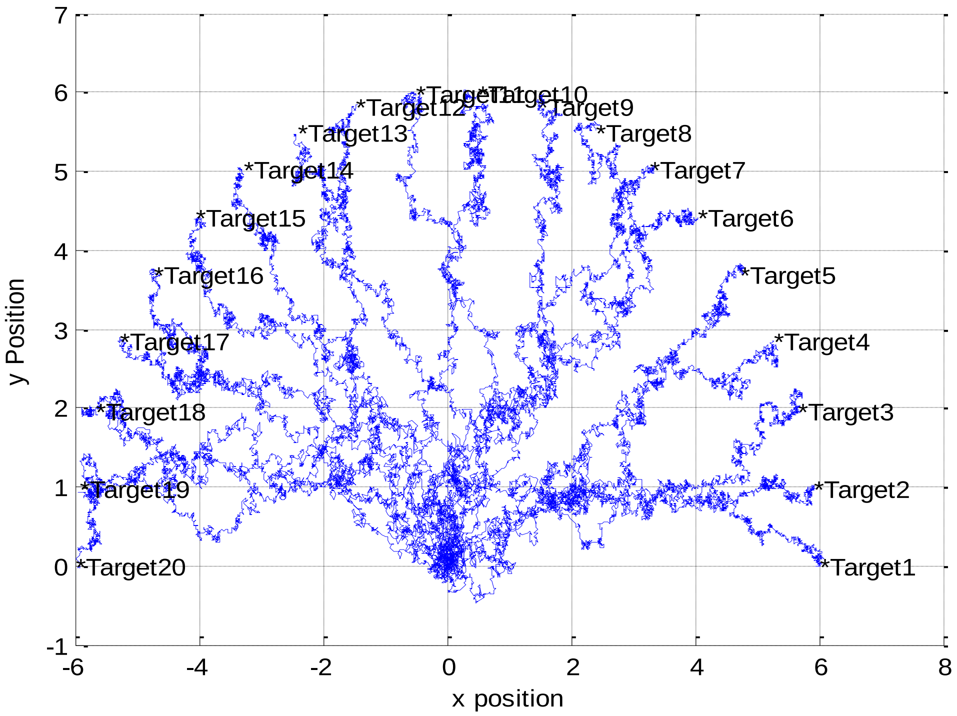


Figure 4.7 Entanglement (Approach II) with 20 agents

Table 4.7. Performance statistics for Case III and Case IV using Approach 1

(100 cases) **Entanglement method** **(Approach 1)**	**5 agents** $k_g = 10$	**10 agents** $k_g = 10$	**20 agents** $k_g = 30$
Minimum	8.4798	21.9928	45.5187
Maximum	16.6769	68.2732	156.4333
Average	12.7734	29.1979	68.4166
Standard deviation	1.6165	6.1664	21.2780
Time (s)	141.7714	302.9712	705.6765

Table 4.8. Performance statistics for Case III and Case IV using Approach 2

(100 cases) **Entanglement method** **(Approach 2)**	**5 agents**	**10 agents**	**20 agents**
Minimum	9.5440	20.2938	45.6062
Maximum	16.6859	31.6877	59.9715
Average	12.8617	26.2754	52.1360
Standard deviation	1.4742	2.4852	3.0041
Time (s)	216.0761	424.153	918.1783

4.7. CONCLUSIONS

Dynamic target assignment of multi-agent systems operating in a stochastic environment is a very complex problem. Three coordination approaches are developed to tackle the above problem. The first approach uses classical game theory ideas to dynamically assign targets. The other two approaches uses quantum inspired coordination models. The implementation of classical game theory approach was limited by the number of participating agents. However, quantum inspired approaches are scalable to large-scale multi-agent systems as demonstrated in 20 agent-20 target simulation results. Among the quantum inspired approaches, approach 2 performs consistently well, independent of the number of participating agents. However, the performance of approach 1 highly depends on the entanglement gain K_g. The implementation of Approach 2 either requires a quantum hardware or a communication network among agents to relay the information regarding realized entangled states. No such limitation exists for Approach 1.

5. CONCLUSIONS

Quantum theory provides a promising new research framework to find ideas that can reduce the computational cost of current machine learning and optimal control synthesizing algorithms. Generally to utilize the power of quantum mechanical phenomena like superposition, entanglement etc. one requires quantum computers, however the pioneers of quantum decision theory have shown us that quantum models can be effectively used to explain complex phenomena like irrational human behavior. Inspired by this idea, three major issues in learning and control of stochastic systems were studied for reformulation using ideas from quantum theory. These issues and proposed solutions are listed below:

- Exploration-exploitation trade-off in reinforcement learning algorithms

A new approach that uses Grover's algorithm to assign probability distribution over available actions is proposed. At every state, depending on the current Q-values, the agent's action state evolves quantum mechanically from the initial superposition state. Further, the degree of quantum evolution depends upon the relative difference between Q-values of the possible future states. Unless the relative difference is quite high, at any learning time-step the agent might prefer equally all the possible actions. This is one of the major difference between QiRL action selection mechanism and the currently popular approaches like ε-greedy, Softmax approach etc., wherein a user defined time-dependent parameter like ε or T decides the degree of exploration/exploitation. The effectiveness of the QiRL algorithm was demonstrated using a complex prey-predator problem, wherein it totally outperformed the Softmax approach.

- Synthesizing optimal controller for stochastic systems

A new single network adaptive critic approach that uses path integrals to adaptively estimate the optimal cost function is proposed. This approach iteratively estimates the optimal cost function using numerically computed state trajectories samples. For a stochastic system, to accurately estimate the optimal cost function large number of trajectory samples are required. However, the proposed approach minimizes this requirement by using an adaptive importance sampling technique. This technique is derived using the path integral formulation of stochastic optimal control theory. For a certain class of linear stochastic systems, the convergence of the proposed adaptive critic algorithm to optimal control solutions was theoretically demonstrated. Further, this methodology was also applied on a difficult stochastic Vanderpol oscillator problem. It was able to come up with stabilizing solution for all the test cases.

- Dynamic target assignment of multi-agent systems in stochastic environment

Two quantum inspired coordination mechanisms that are easily scalable to large-scale multi-agent systems are proposed. These approaches uses the entanglement phenomena to effectively reduce the dimension of the joint action space of the multi-agent systems. In Approach I, each agent models the influence of other agents on its action choices using an entanglement model. Hence, an agent’s action space and its belief space are entangled. In Approach II, all the agent’s action spaces are physically entangled. This physical entanglement is simulated classically by explicit communication. Approach II, performed consistently well in all the test cases. The performance of Approach I depended on the magnitude of a user-defined parameter. For both the approaches, the computational time increases linearly with the increase in number of agents.

BIBLIOGRAPHY

[1] Bertsekas, P. D., *Dynamic programming and optimal control*, Athena Scientific, 2012.

[2] Fleming, H. W., and Rishel, W. R., *Deterministic and stochastic optimal control,* Springer-Verlag, New York, 1975.

[3] Bryson, A. E., and Ho, Y. C., *Applied optimal control,* Taylor and Francis, 1975.

[4] Shor, P., "Polynomial-type algorithms for prime factorization and discrete logarithms on a quantum computer," Siam Journal of Scientific and Statistical Computing, Vol. 26, 1997, pp. 1484-1494.

[5] Grover, L. K., "A fast quantum mechanical algorithm for database search," Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996, pp. 212-219.

[6] Grover, L. K., "Quantum mechanics helps in searching for a needle in a haystack," Physics Review Letters, Vol. 79, No. 2, 1997, 325–327.

[7] Busemeyer, J. R., and Bruza, P. D., *Quantum models of cognition and decision*. Cambridge university press, 2012.

[8] Goldstein, H., Poole, P., C., and Safko, L. J., *Classical mechanics,* 3rd edition, Addison-Wesley, 2001.

[9] Dirac, P. A. M., *The principles of quantum mechanics*, Clarendon, Oxford, UK, 1958.

[10] Feynman, P. R. & Hibbs, A., *Quantum mechanics and path integrals: Emended edition,* Dover, 2005.

[11] Neumann, V. J., *Mathematical foundations of quantum mechanics,* Princeton, NJ, USA, 1955

[12] Einstein, A., Podolsky B., and Rosen N., "Can quantum-mechanical description of physical reality be considered complete?," Physics Review Letters, 1935, Vol. 47, pp. 777-780.

[13] Bell, J.S., "On the Einstein-Podolsky-Rosen paradox," *Physics,* Vol. 1, 1964, pp. 195–200

[14] Aspect, A., Grangier, P., Roger, G., "Experimental Tests of Bell's inequalities using time-varying analyzers," Physical Review Letters, Vol. 49, 1982, pp. 1804–1807.

[15] Penrose, R., *The emperor's new mind*; Oxford University: Oxford, UK, 1989.

[16] Holland, R. P., *The quantum theory of motion: An account of the de Broglie-Bohm causal interpretation of quantum mechanics*, Cambridge University press, 1995.

[17] Durr, D. and Teufel, S., *Bohmian mechanics: The physics and mathematics of quantum theory*, Springer edition, 2009.

[18] Papiez, L., "Stochastic optimal control and quantum mechanics," Journal of Mathematical Physics, Vol. 23, No. 6, 1982, pp.1017-1019.

[19] Nelson. E, *Dynamical theories of brownian motion*, Princeton University press, 1967.

[20] Guerra, F., and Morato, M. L., "Quantization of dynamical systems and stochastic control theory," Physical Review, Vol. 27, No.8, 1983, pp. 1774-1786.

[21] Kotsiantis, B. S., "Supervised machine learning: a review of classification techniques," Informatica, Vol. 31, 2007, pp. 249-268.

[22] Sutton, S. R., and Barto, A., *Reinforcement learning: An Introduction*, MIT press, 1998.

[23] Kaelbling, P. L., Littman, L. M., and Moore, W. A., "Reinforcement learning: a survey," Journal of Artificial Intelligence Research, Vol. 4, 1996, pp. 237-285.

[24] Puterman, L. M., *Markov decision processes*: *Discrete stochastic dynamic programming*, Wiley-Interscience, 2005.

[25] Watkins, C. J. C. H., Learning from delayed rewards, PhD Thesis, University of Cambridge, England, 1989.

[26] Even-Dar, E., and Mansour, Y., "Learning rates for Q-learning," Journal of Machine Learning Research, Vol. 5, 2003, pp.1–25.

[27] Sutton, R., "Learning to predict by the methods of temporal difference," Machine Learning, Vol. 3, No. 1, 1988, pp.9–44.

[28] Thrun, B. S., "Efficient exploration in reinforcement learning," Technical report CMU-CS-92-102, Carnegie Mellon University, 1992.

[29] Brafman, R.I., Tennenholtz, M., "R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning," Journal of Machine Learning Research, Vol.3, 2002, pp. 213-231.

[30] Ishii, S., Yoshida, W., Yoshimoto, J., “Control of exploitation-exploration metaparameter in reinforcement learning,” Neural Networks, Vol. 15(4-6), 2002, pp. 665-687.

[31] Caelen, O., Bontempi, G, “Improving the exploration strategy in bandit algorithms,” In: Learning and Intelligent Optimization, LNCS, No.5313, Springer, 2008, pp. 56-68.

[32] Tokic, M., “Adaptive ε-greedy exploration in reinforcement learning based on value differences,” In: Dillmann, R., Beyerer, J., Hanebeck, U.D., Schultz, T. (eds.) KI, Springer, Heidelberg, 2010, LNCS, vol. 6359, pp. 203–210.

[33] Vermorel, J., Mohri, M., “Multi-armed bandit algorithms and empirical evaluation,” In: Proceedings of the 16th European Conference on Machine Learning (ECML'05), Porto, Portugal, 2005, pp. 437-448.

[34] Nielsen, M. A., and Chuang, I. L., *Quantum computation and quantum information*, Cambridge University press, 2010.

[35] Dong, D., Chen, C., Chu, J., and Tarn, T., “Robust quantum-inspired reinforcement learning for robot navigation,” IEEE Transactions on Mechatronics, Vol. 17, No. 1, 2012, pp. 86-97.

[36] Dong, D., Chen, C., Li, H., and Tarn, T.J., “Quantum reinforcement learning,” IEEE Transactions Systems Man Cybernetics- B, Vol.38, No. 5, 2008, pp.1207–1220.

[37] Chen, C. L., Dong, D. Y., and Chen, Z. H., “Quantum computation for action selection using reinforcement learning,” International Journal of Quantum Information, Vol.4, No. 6, 2006, pp. 1071–1083.

[38] Brassard, G., and Hoyer, P., “An exact quantum polynomial-time algorithm for simon’s problem,” Proceedings of 5th Israel Symposium on the Theory of Computing Systems, IEEE Computer Society Press, 1997, pp. 12-23.

[39] Brassard, G., Hoyer, P., Mosca, M., and Tapp, A., “Quantum amplitude amplification and estimation,” arXiv preprint quant-ph/0005055, 2000.

[40] Yong, J., and Zhou, X., *Stochastic controls –Hamiltonian equations and HJB equations,* Springer-Verlag, New York, 1999.

[41] Si, J., Barto, A. G., Powell, B. W., and Wunsch, D., *Handbook of learning and approximate dynamic programming, 2004.*

[42] Bertsekas, D. P., and Tsitsiklis, J. N., *Neuro-Dynamic programming,* Athena Scientific, Belmont, MA, 1996.

[43] Werbos, P. J., "*Approximate dynamic programming for real-time control and neural modeling,*" In D. A. White, & D. A Sofge (Eds.), Handbook of intelligent control, Multiscience Press., 1992.

[44] Werbos, P. J., "Back propagation through time: What it does and how to do it," Proceedings of the IEEE, Vol. 78, No.10, 1990, pp 1550–1560.

[45] Prokhorov, V. D., and Wunsch, D. C., "Adaptive critic designs*",* IEEE transactions on neural networks, Vol. 8, No. 5, 1997.

[46] Padhi, R., Unnikrishnan, N., Wang, X., and Balakrishnan, S. N., "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Networks,* Vol. 19, No. 10, 2006, pp 1648–1660.

[47] Balakrishnan, S. N., and Biega, V, "Adaptive-critic based neural networks for aircraft optimal control,*"* Journal of Guidance, Control and Dynamics, Vol. 19, No. 4, 1996, pp. 893–898.

[48] Han, D., and Balakrishnan, S. N., "Adaptive critics based neural networks for agile missile control", Journal of Guidance, Control and Dynamics, Vol. 25, 2002, pp 404–407.

[49] Venayagamoorthy, G. K., Harley, R. G., and Wunsch, D. C., "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbo generator," IEEE Transactions on Neural Networks, Vol. 13, 2002, pp 764–773.

[50] Kappen, H. J., "Path Integrals and symmetry breaking for optimal control theory," arXiv:physics/0505066v4.

[51] Broek, V. D. B., Wiegerinck, W., and Kappen, B., "Graphical model inference in optimal control of stochastic multi-agents," Journal of Artificial Intelligence Research, Vol. 32, 2008, pp. 95-122.

[52] Theodorou, E. A., "Iterative path integral stochastic optimal control: Theory and applications to motor control", PhD Thesis, University of Southern California, 2011.

[53] Wang, F., and Saridis, G. N., "On successive approximation of optimal control of stochastic dynamical systems," *In Modeling Uncertainty an Examination of Stochastic Theory, Methods and Applications,* Boston, Kluwer Academic Publishers, 2002, pp 333-358.

[54] Zhu, J., “On stochastic riccati equations for the stochastic LQR problem”, Systems & Control Letters, Vol. 54, 2005, pp. 119-124.

[55] Hastings, W., “Monte Carlo sampling methods using Markov chains and their applications,” Biometrika, Vol. 57, 1, pp. 97–109.

[56] Ceperley, D. M., “Metropolis methods for quantum Monte Carlo simulations,” AIP Conference Proceedings, 690, Vol. 85, 2003. http://dx.doi.org/10.1063/1.1632120.

[57] Tomlin, C., Pappas, J. G., and Sastry, S., “Conflict resolution for air traffic management: A study in multi-agent hybrid systems”, IEEE Transactions on Automatic Control, Vol. 43, No. 4,1998, pp. 509-521.

[58] Ren, W., and Beard, W. R., “Decentralized scheme for spacecraft formation flying via the virtual structure approach” , Journal of Guidance, Control and Dynamics, Vol. 27, No. 1, 2004, pp. 73-82.

[59] Balch, T., and Arkin, R. C., “Behavior-based formation control for multi-robot teams,” IEEE Transactions on Robotics and Automation, Vol. 14, No. 6, 1998, pp. 926–939.

[60] Olfati-saber, R., and Murray, M. R., “Distributed cooperative control of multiple vehicle formations using structural potential functions”, IFAC, 15th Triennial World Congress, Barcelona, Spain, 2001.

[61] Boutilier, C., “Planning, learning and coordination in multi-agent decision process”, Proc. 6th Conf. Theor. Aspects of Rationality and Knowledge, pp.195–210, Amsterdam, 1996.

[62] Bernstein, D. D., Zilberstein, S., and Immerman, N., “The complexity of decentralized control of Markov decision processes”, Uncertainty in Artificial Intelligence Proceedings, 2000, pp. 32-37.

[63] Guestrin, C., Koller, D., and Parr, R., “Multi-agent planning with factored MDPs”, Advances in Neural Information Processing Systems, NIPS-14, 2001.

[64] Neumann, V. J., and Morgenstern, O., *Theory of games and economic behavior*; Princeton University: Princeton, NJ, USA, 1944.

[65] Fudenberg, D., and Tirole, J., 1991, *Game theory*, MIT Press, Cambridge, MA.

[66] Basar, T., and Olsder, G. J., 1999, *Dynamic noncooperative game theory*, SIAM, Philadelphia.

[67] Brown, G. W., 1951, “Iterative solutions of games by fictitious play,” Activity Analysis of Production and Allocation, Koopmans, T. C., ed., Wiley, New York, pp. 374–376.

[68] Fudenberg, D., and Levine, D. K., *The theory of learning in games*, MIT Press, Cambridge, MA., USA, 1998.

[69] Smith, M. J., “The theory of games and the evolution of animal conflicts”, Journal of Theoretical Biology, Vol. 47, 1974, pp. 209-221.

[70] Smith, M. J., *Evolution and the theory of games*, Cambridge University Press, Cambridge, 1982.

[71] Flitney, A.P., and Abbott, D., “An introduction to quantum game theory”, [online publication], http://arXiv: quant-ph /0208069v2, Accessed: 11/2/2014

[72] Meyer, A. D., “Quantum strategies”, Physical review letters, Vol. 82, 1999, pp. 1052-1055.

[73] Eisert, J., Wilkens, M., and Lewenstein, M., “Quantum games and quantum Strategies”, Physical Review Letters, Vol. 83, 1999, pp. 3077-3088.

[74] Benjamin, C. S., and Hayden, M., P., “Multiplayer quantum games”, Physical Review A., 030301(R), Vol. 64, 2001.

[75] Yukalov, I. V., and Sornette, D., “Processing information in quantum decision theory,” Entropy, Vol. 11, 2009, pp. 1073-1120.

[76] Bordley, R. F., “Quantum mechanical and human violations of compound probability principles: Toward a generalized Heisenberg uncertainty principle,” Operations Research, Vol. 46, 1998, pp. 923–926.

[77] Busemeyer, J. R., Wang, Z., and Townsend, J. T., “Quantum dynamics of human decision-making,” Journal of Mathematical Psychology, Vol. 50, 2006, pp. 220–241.

[78] Pothos, E. M., and Busemeyer, J. R., “A quantum probability explanation for violations of ‘rational’ decision theory,” Proceedings of the Royal Society B, Vol. 276, 2009, pp. 2171–2178.

[79] Agarwal, P. M., and Sharda, R., “OR forum- Quantum mechanics and human decision making,” Operations Research, Vol. 61, 1, 2012, pp. 1-16.

[80] Martinez, I., “A connection between quantum decision theory and quantum games: The Hamiltonian of strategic interaction,” Journal of Mathematical Psychology, Vol. 58, 2014, pp. 33-44.

[81] Baras, J. S., “Multi-agent stochastic control: models inspired from quantum physics,” Proceedings of the physics and control conference, Vol. 3, 2003, pp. 747-758.

[82] Savage, L.J., *The foundations of statistics*, Wiley: New York, NY, USA, 1954.

[83] Arslan, G., Marden, R. J., and Shamma, S., J. “Autonomous vehicle-target assignment: A game-theoretical formulation,” Transactions of the ASME, Vol. 129, 2007, pp. 584-596.

[84] Oksendal, B. K., *Stochastic differential equations: An Introduction with applications*, 6th edition, Springer, New York, 2003.

VITA

Karthikeyan Rajagopal was born on May 1982, in Karur, Tamilnadu, India. He received his Bachelor's degree in Mechanical engineering from National Institute of Technology, Trichy, India on May 2003. He worked as a research scientist in Gas Turbine Research Establishment, Bangalore, India from 2004 to 2008. He was accepted for the direct Ph.D. program of the Aerospace engineering department at Missouri University of Science and Technology, Rolla in January 2010. In May 2015, he received his Ph.D. His research interests are Optimal control, Adaptive control and Intelligent autonomous systems.