REINFORCEMENT LEARNING BASED CONTROLLER FOR PRECISION

IRRIGATION

A Dissertation

by

SHIVARAM IRUKULA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee, Jiang Hu
Committee Members, Peng Li
        Duncan M Hank Walker
Head of Department, Miroslav Begovic

December 2015

Major Subject: Computer Engineering

ABSTRACT

Water is a major contributing factor for plant growth and development. Agricultural water management is a major concern for agriculturists, as fresh water resources are being depleted and research in the area of water optimization for agricultural irrigation is in its initial stages. The main goal of the agricultural systems is to supply water spatially according to the soil/land conditions such as water holding capacity of the soil, texture, drainage, texture and topography. While currently existing irrigation systems provide constant irrigation throughout the field, this may result in over irrigation in some areas and under irrigation in the other areas. The main goal of my research is to minimize the excess application of water in a specific location according to the daily conditions like temperature, solar radiation, rainfall from weather data website and soil water content reported by the sensors.

The precision techniques that are presently being used are solely based on sensor data obtained from different sources and leverage supervised learning technique i.e. the system is provided with the solutions to all different environments initially. The main disadvantage of this approach is that, the actual scenario might differ widely from the programmed/provided cases. So the system needs to adapt for variable weather, soil and plant conditions and learn from the past experience as well as try new methods. In this thesis a novel technique to use reinforcement learning (an adaptive learning system) on crop system models to make the irrigation system adaptive is discussed.

The main goal is to optimize the water consumption as much as possible without affecting the crop yield by using reinforcement learning algorithm on maize crop

simulation model. This is done by using the soil parameters, crop parameters, weather data (temperature, probability of rainfall and solar radiation) on DSSAT (Decision Support System for Agro-technology Transfer) maize crop simulation model. Then the water consumption is minimized adaptively by using Q-learning to optimize the daily irrigation. Using simple modular approach of DSSAT to calculate daily yield and leaf area index along with the proposed reinforcement learning controller, almost 40% decrease in water consumption is achieved in comparison to constant irrigation method of the DSSAT model without any significant decrease in yield and leaf area index.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Water consumption for agricultural irrigation alone consumes around 70 percent of the available fresh water used per annum [Goodwin and O'Connell 2008]. In the present scenario, in spite of the widespread promotion, precision irrigation is still not widely accepted as the cost of the infrastructure required in the initial setup exceeds the profits. However, this might change in forth coming years, as it is imminent to conserve and optimize the consumption of water due to increase in population, climate change and depleting underground water resources. Another important factor towards adopting precision irrigation is the observed negative effects of over irrigation. Over irrigation causes drastic increase in total maximum daily loads of temperature, salinity of water and nitrates [Chapman 1994]. Total maximum daily load is a maximum value of the pollutant that can be received by a water resource and still meet the water quality standards. Water quality parameters which include pH, dissolved oxygen and total suspended solids are affected by run off due to over irrigation. All the above causes of over irrigation in turn affects the crop yield along with water and energy wastage. Hence the precision irrigation system, which takes into consideration all these parameters and utilizes the water optimally without having marginal impact on the crop yield is necessary. My research work focuses on maximizing yield with minimized water consumption using reinforcement learning techniques. Daily irrigation is minimized using Q-learning technique keeping the yield constant in comparison to traditionally followed precision irrigation techniques.

2. BACKGROUND

2.1 Precision Irrigation

According to [Raine, Meyer et al. 2007], precision irrigation is a method of applying the right amount of irrigation/water as per the requirements of the individual plants with less impact on the environment. The precise meaning of precision irrigation is application of right and optimal amount of water at different locations at different times taking into account the spatial, temporal and sensor data. Presently this task is performed by variable rate sprinklers with capability of position determination to apply water at variable rates at different locations. Process involved in the precision irrigation cycle is shown in Figure 1.

2.1.1 Precision Irrigation Definition

Precision irrigation is an irrigation system that have knowledge of [Smith et al. 2010]:

1) what action to take next

2) how to do it

3) the previous actions

4) and learns from its previous actions

Precision irrigation is a system that is able to adjust to the existing weather and soil conditions. The main goal of the precision irrigation system is to maximize water use efficiency and yield. Major losses in agriculture is due to the wrong weather prediction and wrong irrigation methods. So there is an imminent need for the improvement of the

irrigation systems so that that farmers would be able to irrigate the crops only when required and misuse of water is minimized.

2.1.2 Benefits of Precision Irrigation

Previous reported work show that the precision irrigation may be used to decrease the water use efficiency up to 80-90% in comparison to the surface irrigation methods where only 40-45% of water can be conserved [Dukes, 2004].

1) Water Usage Efficiency [Shah, Das 2012]

Site specific irrigation is one of the major contributing factors for decrease in water consumption of precision irrigation methods. Previous reports show that around 25% of the water savings can be achieved by efficient application of site specific irrigation models [Hedley, Yule 2009]. Although a maximum of 80-90% water use efficiency is predicted, only up to 25% is able to be achieved till now due to

- Weather data (rainfall) prediction not fully accurate

- Soil conditions are not same at all places, so making a unified model that considers soil conditions in different regions as well as nutrients in the soil for yield prediction is still in initial stages.

2) Crop Yield and Cost Reduction [Shah, Das 2012]

Use of the precision irrigation for agriculture has shown that the yields have been improved over the span of two consecutive years. It has been reported that the potato yield has been improved after following the uniform irrigation methods [King et al. 2006]. The soil erosion has reduced due to decrease in runoff due to usage of precision irrigation techniques. These techniques allowed the use of water efficiently and conservatively

without decrease in the yield and cost of growing the crops, which in turn increased the profits.

2.1.3 Major Steps in Precision Irrigation

Precision Irrigation Cycle [Smith, Bailie et al. 2010]:

- Data Gathering/Acquisition

- Data Interpretation

- Irrigation Control and

- System Evaluation

Figure 1: Precision Irrigation Cycle

3. REINFORCEMENT LEARNING

Reinforcement learning is a popular artificial intelligence/machine learning approach, it learns by interacting with the environment. Reinforcement learning model adaptively learns by trial and error method to find out which action to take to yield maximum reward in a particular state. It tries to maximize the reward/performance of a process by choosing an action for a certain state based on experience as well as exploration. Q-learning is an important reinforcement learning approach and is studied in many different disciplines, which includes game theory, multi-agent systems, genetic algorithms, operations research, statistics, control theory and swarm intelligence [RL]. State space, action space and reward/penalty function are the governing factors of a reinforcement learning system. State space is the set of all available states of the environment, whereas action space is a set of all possible actions for each state and reward function is a function that evaluates the reward for each action taken in any state.

Reinforcement learning does not require a model and is used to find an optimal action for any state. The reinforcement learning agent learns the control policy dynamically at runtime and issues an action based on the current state, then the state transition occurs in the environment after the action is taken. Each action results in reward/penalty. The best action for any state gives high reward in comparison to other actions. Now the algorithm remembers the state-action pair and the Q-value (cumulative reward) associated with it. Each time the system visits a state, it compares the rewards of all possible state-action pairs and chooses the action so that the cumulative reward is maximized in the long run. The Q-value matrix is initialized to a constant and is updated

every time the system takes a certain action and gets the reward. Whenever the system visits a certain state it decides to take a particular action based on the Q-value function so that the long term cumulative reward is maximized. Reinforcement learning model does not require any prior knowledge of the system but if any prior data is available, then it will use the available data to speed up the convergence. Leaf area index and irrigation are the only major objectives and time is not the major constraint for the system for the proposed controller. The controller uses the offline trained data to get maximum leaf area index and minimum irrigation objectives.

3.1   Temporal Difference Learning Methods

Temporal difference learning is a value prediction method used in dealing with the reinforcement learning problems. These methods estimate the value functions using long term reward algorithms such as Q-learning, SARSA. Traditional methods calculate value function by updating the final estimation only after the final action is completed and final reward is received. Whereas the temporal difference methods update the estimated final reward using the intermediate results calculated for each state-action pair at each state in the process.

There are two types of temporal difference methods:

1) On-Policy Methods

On-Policy methods follow a policy and learn the value of the policy for making decisions. The value functions which are long term reward functions, are updated using the followed policy. These methods usually include exploration as a part of the policy. Example of these type of methods is SARSA algorithm.

2) Off-Policy Methods

Off-Policy methods learn a policy from one process and use the policy on another process i.e. different policies are used for behavior and estimation. During estimation, off-policy method value functions are updated based on estimated/hypothetical actions and not actual actions. Even off-policy methods involve certain amount of exploration similar to on-policy methods. Off-Policy methods can follow and differentiate between exploration and behavior/control, which differs from on-policy methods as they cannot have different policies. Example of these kind of methods is Q-learning algorithm.

3.2 Q-learning Algorithm

The equation for Q-learning algorithm [Watkins and Dayan, 1989] is shown below:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma * \max(Q(s_{t+1}, a)) - Q(s_t, a_t)) \qquad (3.1)$$

where s_t, a_t are state and action at time t, and s_{t+1}, a_{t+1}, r_{t+1} are state, action and reward at time t+1. $Q(s_t, a_t)$ is the old Q-value of the state-action pair and $Q(s_{t+1}, a)$ is the maximum possible future value. α is the learning rate and is between 0 and 1. Learning rate of 0 implies the Q-value of the state action-pair remains unchanged, whereas a value of 1 implies the new Q-value replaces the old Q-value. γ is the discount factor and the value is in between 0 and 1. 0 discount factor implies the system is highly dependent on current state and discount factor of 1 implies it is highly dependent on future Q-value. Hence α and γ are chosen somewhere in the middle, so that equal importance is given to the current and future Q-values. The main aim of the algorithm is to maximize the long term reward by mapping of the available states and actions.

Pseudo code for Q-learning Algorithm

Inputs: Set of actions A

Set of states S

Discount factor γ

Learning rate α

For all pairs of state-action, the Q-value is initialized to 0 in Q-value

matrix.

Get the current state s_t

Loop:

Select an action a_t for the present state s_t

Get the future state s_{t+1} and reward value r_{t+1}

Update the Q-value matrix using the equation

$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma * \max(Q(s_{t+1}, a) - Q(s_t, a_t)))$

Change present state to future state i.e. $s_t = s_{t+1}$

Repeat loop

3.2.1 Properties of Q-Learning

The model does not have any prior knowledge/information regarding the action-reward pair or the environment. The Q-learning algorithm needs to run for sufficient amount of time to explore all possible state-action pairs to converge to an optimal policy. Another way to improve convergence of Q-learning is by decreasing the learning rate slowly. In the end an optimal action is chosen, although some bad action choices might be made during learning through exploration. It is an off policy algorithm and does not give

8

importance to the policy being followed but only to the best Q-value. Policy is something that tells the controller how to behave in a certain state.

3.2.2   Learning Rate

Learning rate is the rate at which the new information needs to be learnt by the system and is set between 0 and 1. Learning rate of 0 implies that that the system is not learning anything at all and the value of Q is unchanged at all times. While a learning rate of 1 implies the old Q-value is replaced with the new Q-value, so the system is not learning anything from the past and is solely dependent on the present.

3.2.3   Discount Factor

Discount factor implies the importance of the future rewards in comparison to immediate/current rewards and is also between 0 and 1. A discount factor of 0 will make the system give importance only to current/immediate rewards, while a value of 1 makes the system to give very high importance to future rewards. So generally a value closer to 1 is chosen so that high long term cumulative reward is obtained.

3.2.4   Ɛ-Greedy Method

An action with highest estimated immediate reward known as greedy action is chosen most of the time with probability (1- Ɛ) and an action is chosen randomly for the remaining time with probability Ɛ. The Ɛ-greedy method ensures that after sufficient amount of trials optimal action is discovered. The value of Ɛ is chosen to be 0.3 which is found out by trial and error. Initially Ɛ is chosen as 0.1, then experimented by slowly increasing the value of Ɛ. At Ɛ = 0.3 the empirically best water consumption values were

attained, as ℇ is increased further the water consumption started increasing and so I fixed the value of ℇ to be 0.3.

3.3 Sarsa

SARSA algorithm is similar to the Q-learning algorithm except for the future state Q-value calculation. SARSA simply reflects the variables required by the algorithm to update the Q-value matrix i.e. (s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}). In Q-learning maximum value of all possible future states is considered whereas in SARSA there is randomness involved, which implies it is not always the state with high Q-value. SARSA follows a ℇ-greedy method in which most of the time it chooses the maximum Q-value state (which accounts for experience) and the remaining time it chooses the state randomly to explore any other best state available.

Equation governing SARSA algorithm [Sutton and Barto, 1998] is:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma(Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))) \qquad (3.2)$$

where a_{t+1} is the future action based on ℇ–greedy method that will be taken by the agent. Using the SARSA algorithm for the controller improves the convergence in comparison to Q-learning algorithm. But SARSA is an on-policy algorithm that strictly updates the value functions on basis of experience gained by following some policy, which is not the case with our work. Our work requires the controller to be trained offline as the data for online learning is very less. So we chose Q-learning algorithm which is an off-policy algorithm for the precision irrigation controller.

3.4 Markov Property

A process is said to have Markov property if the future states of the process is solely dependent on the present state and the past states have no impact on the decision making of the future state. A process which possesses the Markov property is said to be a Markov process.

3.5 Markov Decision Process (MDP)

A reinforcement learning or dynamic programming process that possesses the Markov property is said to be a Markov decision process [Sutton and Barto, 1998]. Also a process in which the number of actions and states are finite is known as finite Markov decision process. Most of the modern reinforcement learning is based on finite MDP models. Transition probabilities and rewards are the most important aspects of the Markov Decision Processes.

Transition probability can be defined as the probability that the state goes from one state to another after taking an action. For a current state s and future possible state s' the transition probability for an action a is defined as [Sutton and Barto, 1998]:

$$P_{ss'}^{a} = P\ [S_{t+1} = s'\ |\ S_t = s] \tag{3.3}$$

State transition matrix $\mathbb{P}$ contains all transition probabilities from each state to all possible future states.

$$\mathbb{P} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \tag{3.4}$$

where each row of the matrix sums up to 1, which implies the sum of the transition probabilities of a state is equal to 1.

Future reward is calculated using the current state, future predicted state and current action [Sutton and Barto, 1998].

$$R_{ss'}^a = E\ [r_{t+1} \mid S_{t+1} = s',\ S_t = s,\ a_t = a] \tag{3.5}$$

Transition probabilities are used in the calculation of reward function during the offline training of the Q-learning controller. Future state is chosen according to the transition probabilities for an action. As we don't have reward data explicitly for offline training we get the data from the reward matrices provided to the model initially.

4. DSSAT-CSM

Decision Support System for Agrotechnology Transfer (DSSAT) Crop System Model (CSM) is a software program that constitutes around 28 crop simulation models [DSSAT].The crop simulation models predict the yield and plant growth by taking soil conditions, water availability and weather conditions as inputs to the model. The DSSAT package contains the experimental data to simulate the crop models, so that the user can compare the simulated with the observed yield, growth data. The DSSAT model is being used in over more than 100 countries for research purposes, farmers, educators and policy /decision makers over a period of more than 20 years [Hoogenboom, Jones]. DSSAT CSM model is provided with the scenarios and solutions to the scenarios. For example, if 100mm of water is supplied everyday by traditional irrigation methods, While DSSAT supplies only required amount of water i.e. 100mm – reported soil water content/rainfall.

DSSAT contains separate modules for crop simulation, soil simulation, weather simulation and plant growth simulation data bases. It integrates all the simulation modules to accurately simulate/predict the yield and plant growth over a period of the crop development stage.

International Benchmark Sites Network for Agrotechnology Transfer project with scientists from all over the world developed the decision support system for agrotechnology transfer which is used in agronomic research of crop models [IBSNAT, 1993; Jones et al. 1998l; Uehara, 1998; Tsuji, 1998].
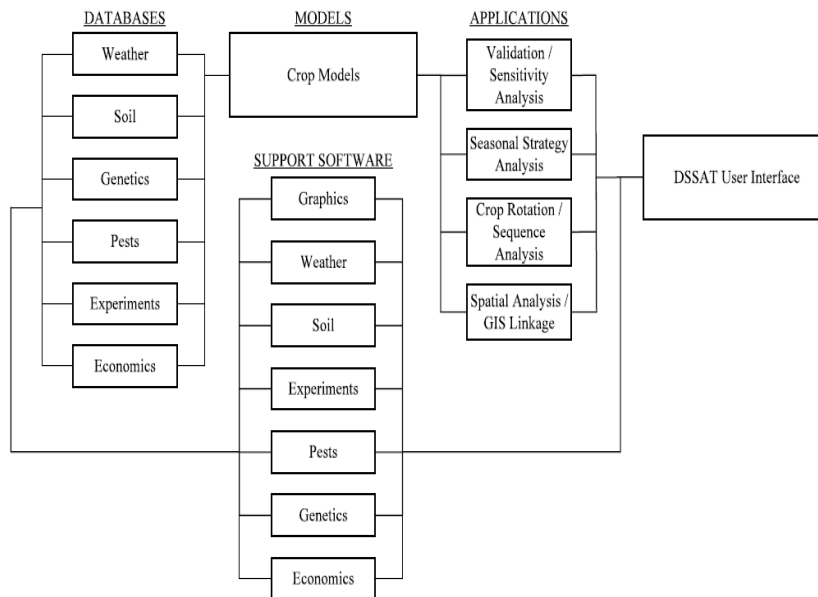
Figure 2: Diagram showing Databases, Models, Support Software and Applications of

Crop System Models on DSSAT [Jones et al. 2003]

From the above Figure 2 we can see that DSSAT comprises of crop models and different independent modules that simulate together to produce the required outputs. Databases contain data regarding weather conditions, soil conditions, genetic information and experimental conditions. Models block contains the data specific to the crop models and support software is for the user to simulate and compare the results with the observed real time data.

The main goals of DSSAT are [Jones et al. 2003] [Tsuji. 1998]:

1) Simulation of crop system models irrespective of location and with minimum available inputs.

2) Platform to incorporate the biotic and abiotic parameters which includes soil phosphorous amount along with different plant diseases.

14

3) Platform which can easily compare different crop system models and be easily able to evaluate, improve and document.

4) Capability to implement the crop system model in new applications at modular level.

The basic structure of the DSSAT CSM which includes Databases, Models, Support Software and Applications is shown in Figure 2.

4.1 Ceres Maize Model

It is one of the most used and widely recognized model since its release in 1986 [Jones and Kinry, 1986] for comparing the developments in maize yield, and growth simulation. It is one of the first crop simulation model that was integrated into the initial version of DSSAT. Many new developments were proposed based on the CERES model and were incorporated into the DSSAT-CSM. Although leaf area is not predicted accurately but total plant biomass and grain yield are predicted/simulated very accurately by the CSM-CERES model [Anapalli et al., 2005; Mastrorilli et al., 2003; Tojo-Soler et al., 2007]. A modified version of CERES-Maize known as CSM-IXIM [Lizaso, et al., 2011] which incorporates the recent developments in leaf area simulation, grain yield, plant ear growth and kernel number and was distributed for DSSAT 4.5v. The modified version of CERES-Maize model is used for the research to test the reinforcement learning controller for precision irrigation.

Yield and Leaf Area Index are the important outputs concerning the model. Leaf Area Index is the ratio of the area covered by the leaves on the ground in a meter square area. The model takes minimum temperature of the day, maximum temperature of the day,

15

solar radiation, rainfall, plant density, field capacity, irrigation and soil water content as the inputs and provides Leaf Area Index and Plant yield as outputs. The plant density is the number of plants per m^2 area in the field. Field capacity is the maximum water holding capacity of the field. The change in leaf area is calculated by the equation [Porter et al. 1999] [Papajorgji and Beck, 2004]:

$$dLAI = SWFAC*PT*PD*EMP1*dN*(\frac{a}{1+a}) \qquad (4.1)$$

where SWFAC is the soil water stress, PT and PD are the growth rate reduction factor and the plant density, EMP1 is the maximum leaf area expansion and dN is change in number of leaves.

$$a=e^{EMP2(N-nb)} \qquad (4.2)$$

$$PT = 1-0.0025((0.25*tmin + 0.75*tmax)-26)^2 \qquad (4.3)$$

where tmin and tmax are minimum and maximum temperature, EMP2, nb are coefficients in the exponential equation and are constants and N is the number of leaves.

Grain Yield is simulated using the equation [Lizaso, et al., 2011]:

$$F_E = \frac{PE*PGR}{1+exp[-0.02(tt-225)]} \qquad (4.4)$$

where F_E is the weight of ear that is subtracted from total dry mass daily, PGR is the daily plant growth rate, PE is an ear partition parameter, tt is thermal time. Thermal time is the sum of the average temperatures of the day from the beginning of ear growth which is assumed to be after 60 days in our experiment.

4.2 Major Types of Precision Irrigation

There are two major types of precision irrigation techniques currently followed, which are 1) Center Pivot Irrigation and 2) Lateral/Linear Move Irrigation.

4.2.1 Center Pivot Irrigation (Water Wheel and Circular Irrigation)

For this method of irrigation the crop lands are designed to be in circular form and the equipment used for irrigation moves around a pivot along the radius of the circular field and the crops are irrigated using sprinklers [Mader, Shelli 2010][USDA NAL].

Central Pivot Irrigation systems are powered using electric motors which replaced the water powered motors. It is highly efficient and helps conserve water efficiently. They use less water in comparison to the other follower surface irrigation and furrow irrigation techniques [FNR]. It also helps in reducing soil tillage, soil erosion and human effort as most of the ground irrigation techniques require channels for the flow of water and soil erosion occurs due to water runoff [FNR].

Center pivot irrigation is currently being used in many countries worldwide and is majorly supplied by Valley, Zimmatic and Reinike manufacturers. Due to heavy initial costs involved in setting up of the irrigation system, it is only followed by a few hundreds of farmers. An example of center pivot system by Rainfine Irrigation Co. LTD is shown in Figure 3.

Figure 3: Center Pivot Farm Irrigation System [Rainfine Irrigation Co. LTD]

4.2.2   Lateral/Linear Move Irrigation

The fields for this type of irrigation are either rectangular or square fields. The irrigation equipment with water sprinklers move in a straight line/linearly. That's why they are called linear move, lateral move and side-roll irrigation systems [Evans, 1997]. They are less common as they need complex management and guidance system compared to center pivot systems and are being followed only by the farmers who do not want to change the rectangular fields to circular fields [Irrigation Models]. An example of the lateral move system by Valley Irrigation is shown in Figure 4.

Figure 4: Valley Two Wheeled Linear Irrigation System [Valley Irrigation]

5 RELATED WORK

The research into irrigation techniques, most importantly precision irrigation in USA began in early 1990's [Smith et al., 2010]. Most of the research at that time was on center pivot and lateral/linear move systems for spatially variable application of water and fertilizers with databases of spatially referenced data being used for system control [King et al., 1996] [Evans et al., 1996] [Duke et al., 1997] [Camp et al., 1998] [Sadler et al., 2000]. A comprehensive review of research that has happened in the area of precision irrigation is reviewed in [Camp et al., 2006]. Variable rate sprinklers for time proportional pulsing and time proportional control has been introduced in [Kincaid and Bulchleiter, 2004] [King and Kincaid, 2004]. More recent works in the area of precision irrigation is comprised of usage of infrared thermometers to develop automatic scheduling of irrigation and control [O'Shaughnessy et al., 2008] [Peters & Evett, 2004, 2005, 2007, 2008].

Another important development in the area of precision irrigation is the usage of radio transmitters and installation of LEPA (Low Energy Precise Application) and sprays on the same irrigation system [Camp, et al., 2006]. Crop Yield patterns for the application of water using non-stationary irrigation systems is studied in New Zealand [Hedley and Yule, 2009] [Yule et al., 2008]. Limited work into the actual benefits of usage of precision irrigation for different crops is done on cotton crop [Clouse, 2006] [Bronson et al., 2006], soybean [Paz et al., 2001] and potatoes [King et al., 2006].

In microprocessor designs, dynamic power management of peripheral devices using reinforcement learning has been published recently in [Shen et al. 2013]. The Q-learning controller proposed in the model does not require any previous knowledge of

20

system load and it learns adaptively from the incoming real time data. A two level performance control model was discussed in the paper which tries to achieve maximum performance at a given power constraint level. Q-learning controller was able to achieve better performance in terms of power in comparison to existing power management techniques. They also extended the Q-learning algorithm to CPU power management and were successfully able to achieve energy requirements along with the performance and temperature constraints.

Although much work is done on precision irrigation techniques and reinforcement learning techniques separately in the past few years, none of them focuses making the precision irrigation control adaptive using reinforcement learning techniques. The controller discussed in this thesis is an attempt to make the precision irrigation control system adaptive using reinforcement learning techniques.

6.   PROPOSED Q-LEARNING BASED IRRIGATION CONTROL

DSSAT crop simulation model for Maize is used to develop the reinforcement learning based controller for precision irrigation [Jones et al. 2003] [Porter et al. 1999]. DSSAT-CSM model is currently used for simulating over more than 18 crop models. Maize crop system model is used to implement and test the functionality of the reinforcement controller. Proposed irrigation system of the reinforcement learning controller for precision irrigation is shown below in figure 5.
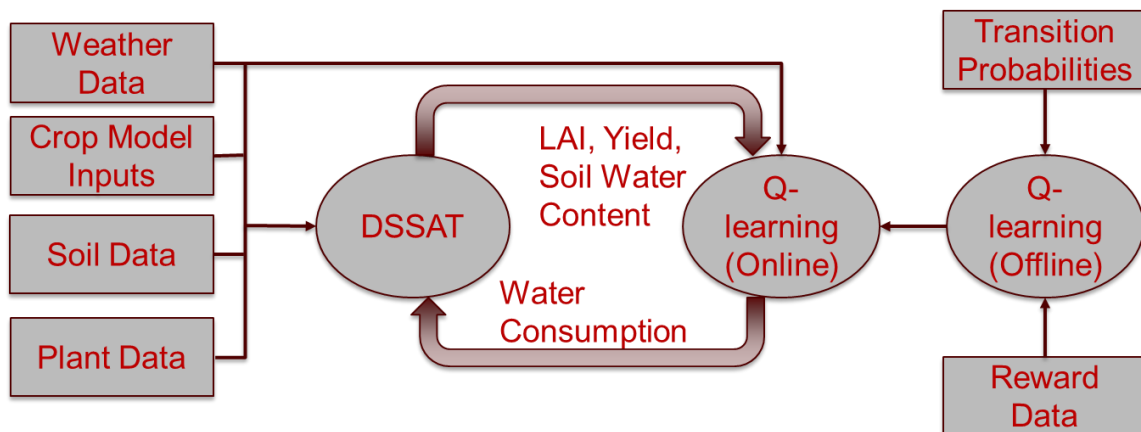
Figure 5: Overview of the Q-learning based Irrigation Control System

6.1   Overview

As we don't have enough data to train the controller on real time data, the controller is trained offline initially using the transition probability and reward matrices which are provided as inputs. After the offline training we get a Q-matrix with all set of state-action pair reward values. As soon as we receive the weather data, crop model inputs, soil data and plant data we get the current state, an action is chosen based on Ɛ-greedy

algorithm using the offline trained Q-value matrix. Action implies the amount of irrigation to be supplied to the crop on that day. It is supplied as input to the DSSAT crop simulation model and we get leaf area index and yield as its outputs. These outputs along with the soil water content are fed back to the Q-learning online controller for calculation of next state and reward function. Then the Q-value matrix is updated once we get the next state and reward function. This process is continued till the end of the season.

Center Pivot Irrigation method is assumed for my research and this method takes approximately 48-72 hours to complete each rotation i.e. it takes almost 3 days for the center pivot sprinkler arm to reach the starting point again. So time interval between irrigation is taken to be 3 days. The simulation is run over a period of three months and the crop is irrigated once in every three days. All the parameters of the maize crop and the sensor data regarding the soil, plant and weather conditions are provided to the model. Temperature and Precipitation input for the model is obtained from US Weather Data website [US Climate Data]. The leaf area index and water consumption are set as the major objectives for the model. The complete overview of the proposed controller is shown in Figure 5.

6.2   Leaf Area Index and Plant Yield

Leaf Area Index is the proportion of the area covered by the leaves on the ground in meter square area. It is a dimensionless quantity and is used on a scale of 0 to10. Scale of 0 implies no part of the ground is covered with leaves and a scale of 10 implies ground is fully covered with leaves. Plant yield is the yield (weight) of the plants in a square meter area in grams. Leaf area index and plant yield are the major objectives that govern the

proposed reinforcement learning irrigation system along with the water consumption for irrigation. These are obtained as outputs of the DSSAT model when all the required inputs are provided for each time interval.

6.3 States and Actions

State of a system is comprised of three parameters rainfall probability, plant condition (leaf area index) and soil water content. Each parameter has three possible states which is a set of conditions {Good (H), Average (M), and Bad (L)}. So total number of possible states is $3^3=27$. Now take states MHH and LHH, plant condition and soil water content in both the cases is high so medium or low probability of rainfall are similar states as field is almost at its full capacity, so we merge these two states. So merging the cases/states that are very similar i.e., (HHL, HHM, MHH and LHH) = HHM, (HMM, HML, MMH and LMH) = HMM, (HLL, HLM, LLH and MLH) = HLL, (MHM, MHL and LHM) = MHM, (MMM, MML and LMM) = MMM, (MLL, MLM and LLM) = MLL, we get a consolidated set of 12 states.

Possible States = [HHH, HHM, HMH, HMM, HLH, HLL, MHM, MMM, MLL, LHL, LML, LLL]

For each state there are 9 possible actions on basis of which the system goes from one state to another state. Number of actions in the action space is chosen by trial and error method. Even if we increase the number of actions more than 9 there was no significant improvement in the yield. So the maximum possible number of actions is set be 9. Action taken is the amount of irrigation to be supplied to the field on that day. Action 1 provides 0%, action 2 provides 12.5%, action 3 provides 25%… action 9 provides 100% of the maximum irrigation.

For example, assume the state 1 to be {bad, bad, bad} by taking an action 1 (no irrigation is supplied) it stays in {bad, bad, bad} condition and does not change the state and the final state of the system is in overall bad condition. This yields a very less reward as it is not preferred for the system to stay in very bad state, so the system tries to avoid this state next time as it yields very less reward. Similarly assume system has taken action 2 (12.5% irrigation required is supplied) and goes from state 1{bad, bad, bad} to state 5{average, average, average}, then there is a significant improvement in the condition of the crop and this yields a high reward value. The reward value is incorporated into the Q-matrix using the Q-learning algorithm (equation 3.1). Now there is a good probability that the system goes to state 5 state from state 1 in future.

6.4 Transition Probability Matrix

It contains the probabilities of a system going from its present state at any time t to next state at time t+1 based on the action taken. We have 12 possible states and 9 possible actions for each state. So we need to have 9 different transition tables each of 12x12 size for 9 different actions, so that all possible scenarios are covered. Sum of the probabilities of any row is equal to 1, implies each state has a finite probability to go to another state and the sum of the transition probabilities is equal to 1. $P_{ss'}$ is the probability of transition from a state at time t to next state at time t+1. The matrices are used for offline training of the controller. The transition probability is used to find the next state of the controller for offline training.

6.5 Reward Function Calculation

The reward of each action taken for real time data is based on the Leaf Area Index, Plant Weight benefitted by that action and water consumption every three days is given by:

$$R_{t+1} = \frac{Normalized\ Yield\ /\ each\ time\ step}{0.1 + (Normalized\ Water\ Consumption/each\ time\ step)^{\lambda}} * 100; \tag{6.1}$$

$$Normalized\ yield = \frac{Leaf\ Area\ Index\ calculated\ by\ the\ Q-learning\ Controller}{Maximum\ Leaf\ Area\ Index\ calculated\ by\ DSSAT\ model} \tag{6.2}$$

$$r_{t+1} = R'_{t+1}\ \text{for offline training} \tag{6.3}$$

$$r_{t+1} = R_{t+1}\ \text{for training on real time data.} \tag{6.4}$$

where R_{t+1} is the actual reward function used for online learning on real time data and r_{t+1} is the final reward function which is used in the Q-learning algorithm. R'_{t+1} is the reward for offline training obtained from the manually provided reward matrices. The denominator of the actual reward function contains a constant value 0.1, used to compensate for the action 1 i.e., when normalized water consumption is 0. Normalized Leaf Area Index is the ratio of leaf area index calculated by the reinforcement learning controller and leaf area index calculated by the DSSAT model alone with full amount of irrigation for each time interval. Normalized water consumption is the ratio of the irrigation provided for the day and maximum irrigation that can be supplied without over flow, which is maximum field capacity of the field.

6.6 Offline and Online Training

The time interval between two irrigation periods is 3 days and growth interval of the crop is 90 days. So we have only 30 real time sensor readings to be used in the learning. But it is nowhere sufficient for the irrigation control system to learn from the experience and function properly as intended. So the learning needs to run on offline data. The offline learning is used to predict the correct action for a particular state when real time data is received. SARSA is an on-policy learning method which does not work well with offline training (different policy than online learning), so that's why Q-learning based Monte Carlo sampling is chosen instead of SARSA for the research work.

During offline training, after we get the state we chose an action based on ϵ-greedy method similar to online learning. Then the next state is found out using the probability of the transition from the transition matrix for the action taken. The reward for offline training is chosen from the reward matrix provided as input to the controller. An action is chosen according to ϵ -greedy method in both offline and online training. First time an action is chosen at random as Q-matrix is initialized to 0 and all values of Q-matrix are 0 and based on the action we get next state. Using the ϵ-greedy algorithm we get the next action, from the obtained action, state, next state and next action we update the Q-matrix using Q-learning algorithm. For example, if we choose action 5 in state 1 and if we get next state as 5 and the value of $Q(1, 5)$ is updated using the Q-learning algorithm (equation 3.1). Similarly when the system goes to a state with all row values of Q-matrix equal to 0 then an action is chosen at random. If a state with row values in Q-matrix are non-zero then a

state action pair with high Q(s, a) is chosen from the nine possible action according to ℇ - greedy method. Q-matrix is updated every time an action is taken.

During online learning, the state of the system is decided by the received sensor results on plant condition, soil moisture and the rainfall probability. After getting the state an action is chosen according to the ℇ-greedy method. The next state is calculated from the leaf area index data obtained through the DSSAT model, next set of weather prediction and soil sensor data. And the reward is also calculated using the leaf area index predicted by the DSSAT model using the equation 6.1. Then the Q-matrix is updated using the Q-learning algorithm.

By the time the offline simulation completes, almost all state action pairs have been covered and converged to a range of values. Now as soon as real time sensor/weather inputs are received we use the already trained Q-value matrix for choosing an action. The action that yields high Q-value is chosen from the matrix using ℇ-greedy method. Based on the action the amount of irrigation at the location is decided according to the rainfall that has happened and existing soil water content.

For example, 100mm is the maximum water content that the field can hold. The soil water content sensor reported 40mm of water, then 100-40 = 60mm is the required amount of water if without using the reinforcement learning irrigation system. Now by reinforcement learning irrigation system, the controller decides how much water to be supplied based on the action taken. Feedback is received from the model on the basis of action in terms of yield and leaf area index. It is used in the calculation of reward function

28

and based on the reward, Q-matrix is updated and next action taken will be based on the

updated Q-matrix.

7.1 Experimental Setup

The following transition matrices and reward matrices are used during the offline training of the Q-learning controller. Transition Probability1 implies the transition probability matrix to be used when action 1 is taken. Similarly Reward1 is the reward obtained by taking an action 1 with the specific state transition.

Transition Probability1 [12] [12] =

0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15
0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15

Transition Probabilities 2[12][12] =

0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15
0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15

Transition Probabilities 3[12][12] =

0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15
0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15

Transition Probabilities 4[12][12] =

0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15
0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15

Transition Probabilities 5[12][12] =

0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15
0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15

Transition Probabilities 6[12][12] =

0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15
0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15

Transition Probabilities 7[12][12] =

0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15
0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15

Transition Probabilities 8[12][12] =

0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15
0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15

Transition Probabilities 9[12][12] =

0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0	0.05	0.1	0.1	0.1	0.05	0.05	0.15
0.3	0.1	0	0.1	0	0.1	0.1	0.1	0.05	0	0.1	0.05
0.2	0.15	0	0.1	0	0.05	0.1	0.1	0.1	0	0.1	0.1
0.2	0.1	0.1	0.1	0.1	0.05	0.1	0.1	0.05	0	0.05	0.05
0.1	0.2	0.1	0.1	0	0.1	0	0.1	0.1	0	0.1	0.1
0.2	0.2	0	0	0	0	0.2	0.1	0.05	0.1	0.1	0.05
0.2	0.1	0.05	0.05	0	0.05	0.1	0.15	0.1	0	0.1	0.1
0.1	0.05	0.1	0.1	0.05	0.15	0.1	0.1	0.1	0	0	0.15
0.05	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0	0.1	0.1	0
0.1	0.05	0.2	0.05	0.2	0.05	0	0.1	0.1	0	0.1	0.05
0.15	0.15	0.1	0.1	0.05	0.1	0	0.1	0.1	0	0	0.15

Reward1 [12] [12] =

30	10	0	10	0	10	10	10	5	0	10	5
20	15	0	10	0	5	10	10	10	0	10	10
20	10	10	10	10	5	10	10	5	0	5	5
10	20	10	10	0	10	0	10	10	0	10	10
10	5	20	5	20	5	0	10	10	0	10	5
15	15	10	10	5	10	0	10	10	0	0	15
20	20	0	0	0	0	20	10	5	10	10	5
20	10	5	5	0	5	10	15	10	0	10	10
10	5	10	10	5	15	10	10	10	0	0	15
5	10	10	10	15	10	10	10	0	10	10	0
5	5	5	5	10	10	10	10	10	10	10	10
10	10	10	10	0	5	10	10	10	5	5	15

Reward 2[12][12] =

20	20	0	0	0	0	20	10	5	10	10	5
20	10	5	5	0	5	10	15	10	0	10	10
10	5	10	10	5	15	10	10	10	0	0	15
5	10	10	10	15	10	10	10	0	10	10	0
5	5	5	5	10	10	10	10	10	10	10	10
10	10	10	10	0	5	10	10	10	5	5	15
30	10	0	10	0	10	10	10	5	0	10	5
20	15	0	10	0	5	10	10	10	0	10	10
20	10	10	10	10	5	10	10	5	0	5	5
10	20	10	10	0	10	0	10	10	0	10	10
10	5	20	5	20	5	0	10	10	0	10	5
15	15	10	10	5	10	0	10	10	0	0	15

Reward 3[12][12] =

$$
\begin{bmatrix}
5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 0 & 5 & 10 & 10 & 10 & 5 & 5 & 15 \\
30 & 10 & 0 & 10 & 0 & 10 & 10 & 10 & 5 & 0 & 10 & 5 \\
20 & 15 & 0 & 10 & 0 & 5 & 10 & 10 & 10 & 0 & 10 & 10 \\
20 & 10 & 10 & 10 & 10 & 5 & 10 & 10 & 5 & 0 & 5 & 5 \\
10 & 20 & 10 & 10 & 0 & 10 & 0 & 10 & 10 & 0 & 10 & 10 \\
20 & 20 & 0 & 0 & 0 & 0 & 20 & 10 & 5 & 10 & 10 & 5 \\
20 & 10 & 5 & 5 & 0 & 5 & 10 & 15 & 10 & 0 & 10 & 10 \\
10 & 5 & 10 & 10 & 5 & 15 & 10 & 10 & 10 & 0 & 0 & 15 \\
5 & 10 & 10 & 10 & 15 & 10 & 10 & 10 & 0 & 10 & 10 & 0 \\
10 & 5 & 20 & 5 & 20 & 5 & 0 & 10 & 10 & 0 & 10 & 5 \\
15 & 15 & 10 & 10 & 5 & 10 & 0 & 10 & 10 & 0 & 0 & 15
\end{bmatrix}
$$

Reward 4[12][12] =

$$
\begin{bmatrix}
30 & 10 & 0 & 10 & 0 & 10 & 10 & 10 & 5 & 0 & 10 & 5 \\
20 & 15 & 0 & 10 & 0 & 5 & 10 & 10 & 10 & 0 & 10 & 10 \\
20 & 10 & 10 & 10 & 10 & 5 & 10 & 10 & 5 & 0 & 5 & 5 \\
10 & 20 & 10 & 10 & 0 & 10 & 0 & 10 & 10 & 0 & 10 & 10 \\
10 & 5 & 20 & 5 & 20 & 5 & 0 & 10 & 10 & 0 & 10 & 5 \\
15 & 15 & 10 & 10 & 5 & 10 & 0 & 10 & 10 & 0 & 0 & 15 \\
20 & 20 & 0 & 0 & 0 & 0 & 20 & 10 & 5 & 10 & 10 & 5 \\
20 & 10 & 5 & 5 & 0 & 5 & 10 & 15 & 10 & 0 & 10 & 10 \\
10 & 5 & 10 & 10 & 5 & 15 & 10 & 10 & 10 & 0 & 0 & 15 \\
5 & 10 & 10 & 10 & 15 & 10 & 10 & 10 & 0 & 10 & 10 & 0 \\
5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 0 & 5 & 10 & 10 & 10 & 5 & 5 & 15
\end{bmatrix}
$$

Reward 5[12][12] =

$$
\begin{bmatrix}
20 & 20 & 0 & 0 & 0 & 0 & 20 & 10 & 5 & 10 & 10 & 5 \\
20 & 10 & 5 & 5 & 0 & 5 & 10 & 15 & 10 & 0 & 10 & 10 \\
10 & 5 & 10 & 10 & 5 & 15 & 10 & 10 & 10 & 0 & 0 & 15 \\
5 & 10 & 10 & 10 & 15 & 10 & 10 & 10 & 0 & 10 & 10 & 0 \\
5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 0 & 5 & 10 & 10 & 10 & 5 & 5 & 15 \\
30 & 10 & 0 & 10 & 0 & 10 & 10 & 10 & 5 & 0 & 10 & 5 \\
20 & 15 & 0 & 10 & 0 & 5 & 10 & 10 & 10 & 0 & 10 & 10 \\
20 & 10 & 10 & 10 & 10 & 5 & 10 & 10 & 5 & 0 & 5 & 5 \\
10 & 20 & 10 & 10 & 0 & 10 & 0 & 10 & 10 & 0 & 10 & 10 \\
10 & 5 & 20 & 5 & 20 & 5 & 0 & 10 & 10 & 0 & 10 & 5 \\
15 & 15 & 10 & 10 & 5 & 10 & 0 & 10 & 10 & 0 & 0 & 15
\end{bmatrix}
$$

Reward 6[12][12] =

$$
\begin{bmatrix}
5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 0 & 5 & 10 & 10 & 10 & 5 & 5 & 15 \\
30 & 10 & 0 & 10 & 0 & 10 & 10 & 10 & 5 & 0 & 10 & 5 \\
20 & 15 & 0 & 10 & 0 & 5 & 10 & 10 & 10 & 0 & 10 & 10 \\
20 & 10 & 10 & 10 & 10 & 5 & 10 & 10 & 5 & 0 & 5 & 5 \\
10 & 20 & 10 & 10 & 0 & 10 & 0 & 10 & 10 & 0 & 10 & 10 \\
20 & 20 & 0 & 0 & 0 & 0 & 20 & 10 & 5 & 10 & 10 & 5 \\
20 & 10 & 5 & 5 & 0 & 5 & 10 & 15 & 10 & 0 & 10 & 10 \\
10 & 5 & 10 & 10 & 5 & 15 & 10 & 10 & 10 & 0 & 0 & 15 \\
5 & 10 & 10 & 10 & 15 & 10 & 10 & 10 & 0 & 10 & 10 & 0 \\
10 & 5 & 20 & 5 & 20 & 5 & 0 & 10 & 10 & 0 & 10 & 5 \\
15 & 15 & 10 & 10 & 5 & 10 & 0 & 10 & 10 & 0 & 0 & 15
\end{bmatrix}
$$

Reward 7[12][12] =

$$
\begin{bmatrix}
30 & 10 & 0 & 10 & 0 & 10 & 10 & 10 & 5 & 0 & 10 & 5 \\
20 & 15 & 0 & 10 & 0 & 5 & 10 & 10 & 10 & 0 & 10 & 10 \\
20 & 10 & 10 & 10 & 10 & 5 & 10 & 10 & 5 & 0 & 5 & 5 \\
10 & 20 & 10 & 10 & 0 & 10 & 0 & 10 & 10 & 0 & 10 & 10 \\
10 & 5 & 20 & 5 & 20 & 5 & 0 & 10 & 10 & 0 & 10 & 5 \\
15 & 15 & 10 & 10 & 5 & 10 & 0 & 10 & 10 & 0 & 0 & 15 \\
20 & 20 & 0 & 0 & 0 & 0 & 20 & 10 & 5 & 10 & 10 & 5 \\
20 & 10 & 5 & 5 & 0 & 5 & 10 & 15 & 10 & 0 & 10 & 10 \\
10 & 5 & 10 & 10 & 5 & 15 & 10 & 10 & 10 & 0 & 0 & 15 \\
5 & 10 & 10 & 10 & 15 & 10 & 10 & 10 & 0 & 10 & 10 & 0 \\
5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 0 & 5 & 10 & 10 & 10 & 5 & 5 & 15
\end{bmatrix}
$$

Reward 8[12][12] =

$$
\begin{bmatrix}
20 & 20 & 0 & 0 & 0 & 0 & 20 & 10 & 5 & 10 & 10 & 5 \\
20 & 10 & 5 & 5 & 0 & 5 & 10 & 15 & 10 & 0 & 10 & 10 \\
10 & 5 & 10 & 10 & 5 & 15 & 10 & 10 & 10 & 0 & 0 & 15 \\
5 & 10 & 10 & 10 & 15 & 10 & 10 & 10 & 0 & 10 & 10 & 0 \\
5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 0 & 5 & 10 & 10 & 10 & 5 & 5 & 15 \\
30 & 10 & 0 & 10 & 0 & 10 & 10 & 10 & 5 & 0 & 10 & 5 \\
20 & 15 & 0 & 10 & 0 & 5 & 10 & 10 & 10 & 0 & 10 & 10 \\
20 & 10 & 10 & 10 & 10 & 5 & 10 & 10 & 5 & 0 & 5 & 5 \\
10 & 20 & 10 & 10 & 0 & 10 & 0 & 10 & 10 & 0 & 10 & 10 \\
10 & 5 & 20 & 5 & 20 & 5 & 0 & 10 & 10 & 0 & 10 & 5 \\
15 & 15 & 10 & 10 & 5 & 10 & 0 & 10 & 10 & 0 & 0 & 15
\end{bmatrix}
$$

Reward 9[12][12] =

$$\begin{bmatrix}
5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 0 & 5 & 10 & 10 & 10 & 5 & 5 & 15 \\
30 & 10 & 0 & 10 & 0 & 10 & 10 & 10 & 5 & 0 & 10 & 5 \\
20 & 15 & 0 & 10 & 0 & 5 & 10 & 10 & 10 & 0 & 10 & 10 \\
20 & 10 & 10 & 10 & 10 & 5 & 10 & 10 & 5 & 0 & 5 & 5 \\
10 & 20 & 10 & 10 & 0 & 10 & 0 & 10 & 10 & 0 & 10 & 10 \\
20 & 20 & 0 & 0 & 0 & 0 & 20 & 10 & 5 & 10 & 10 & 5 \\
20 & 10 & 5 & 5 & 0 & 5 & 10 & 15 & 10 & 0 & 10 & 10 \\
10 & 5 & 10 & 10 & 5 & 15 & 10 & 10 & 10 & 0 & 0 & 15 \\
5 & 10 & 10 & 10 & 15 & 10 & 10 & 10 & 0 & 10 & 10 & 0 \\
10 & 5 & 20 & 5 & 20 & 5 & 0 & 10 & 10 & 0 & 10 & 5 \\
15 & 15 & 10 & 10 & 5 & 10 & 0 & 10 & 10 & 0 & 0 & 15
\end{bmatrix}$$

Initially the DSSAT-CSM model is run with the sensor inputs for constant irrigation values i.e. if 100mm of irrigation is required, then 10 different sets of irrigation {10mm, 20mm, 30mm, 40mm, 50mm, 60mm, 70mm, 80mm, 90mm, 100mm} is provided to the model, then the irrigation amount at which maximum leaf area index and maximum yield are obtained is noted down.

Table 1: DSSAT Model Parameters and Constants [Porter et al. 1999]

Parameter/Constant	Value
PD	$9.5/m^2$
EMP1	0.104
EMP2	0.1
nb	10
dN	0-2
PE	0.35

Table 1 contains the values of the constants and crop parameters used in DSSAT crop simulation model. The reinforcement learning irrigation control system tries to achieve the maximum yield and leaf area index with less amount of irrigation than the existing constant irrigation DSSAT model. Initially the controller is trained offline with the above provided transition matrix and reward matrix data, then uses the offline trained knowledge on real time data. The controller is run offline 1000000 times and it takes 126 seconds to complete simulation. The simulations are run for different values of λ {0.1, 0.2, 0.4, 0.5, 0.6, 0.8, 0.9, 1, 1.5, 2} for a set of weather data inputs of each region, then compared with the plots obtained using the different set of learning rate and discount factor values. Three sets of learning rate and discount factor values {[0.6, 0.8], [0.6, 0.6], [0.8, 0.6]} are taken for comparing the functioning of the irrigation control system. Our method is tested with the weather inputs from three different regions over United States.

Due to the randomness associated with the proposed irrigation system, I ran the irrigation system for 100 times and averaged the obtained yield and leaf area index. In the below shown Figures 6, 7, 8, 9, 10 and 11, blue curve is plotted with learning rate of 0.6 and discount factor of 0.8, cyan curve is plotted with learning rate of 0.6 and discount factor of 0.6, green curve is plotted with learning rate of 0.8 and discount factor of 0.6 and the red curve is plotted using constant irrigation system.

In the case with $\lambda=1$, the leaf area index of the crop is slightly lower but water consumption is noticeably decreased. If the value of λ is chosen to be less than 1, then there is a reasonable increase in the water consumption for irrigation with slight increase

in leaf area index and yield. If we take $\lambda > 1$, then there is a slight decrease in water consumption and output leaf area index in comparison to the case with $\lambda = 1$.

The value of $\lambda < 1$ implies the reward function is more dependent on normalized yield in comparison to normalized water consumption, this is preferred when we need to achieve maximum leaf area index constraint with loosened water consumption constraint. And if $\lambda = 1$ equal importance is given to both normalized yield and normalized water consumption, this is the case where yield and water costs are in similar range. In the case where λ goes to 0, the importance of water consumption in calculating the reward function decreases, which implies the irrigation system tries to decrease the impact of water consumption as low as possible so that the reward value of the Q-learning is dependent solely on yield and not water constraint.

7.2   Experiment Results

Plots containing plant yield vs water consumption for the weather data of College station, TX with three different sets of learning rate and discount factor on our irrigation control system along with constant irrigation system is shown in Figure 6.
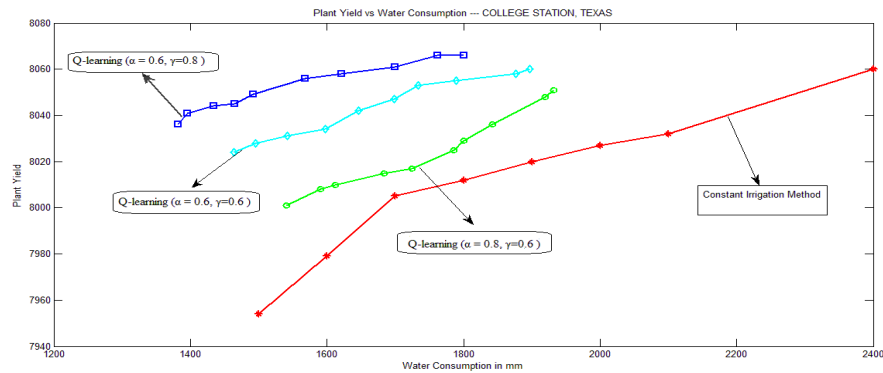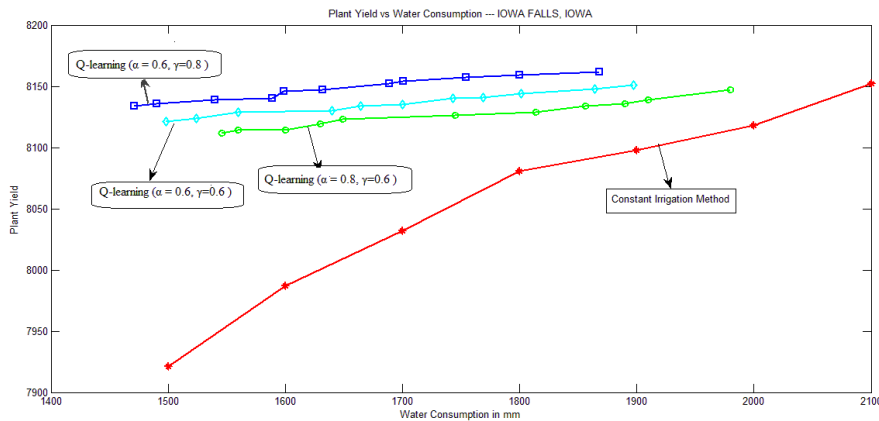

Figure 6: Plant Yield vs Water Consumption Plots for College Station, Texas.

38

Plots containing plant yield vs water consumption for the weather data of Cambridge, Nebraska with three different sets of learning rate and discount factor on our irrigation control system along with constant irrigation system is shown in Figure 7.
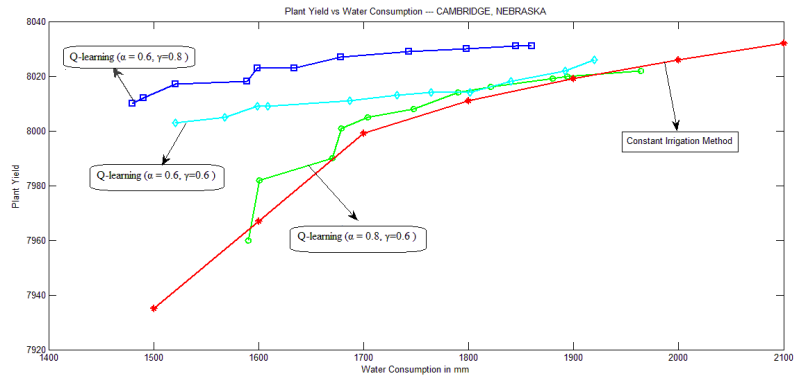
Figure 7: Plant Yield vs Water Consumption Plots for Cambridge, Nebraska.

Plots containing plant yield vs water consumption for the weather data of Iowa Falls, IOWA with three different sets of learning rate and discount factor on our irrigation control system along with constant irrigation system is shown in Figure 8.

Figure 8: Plant Yield vs Water Consumption Plots for Iowa Falls, IOWA.

Table 2: Plant Weight Comparison Table for Different Regions

Region	Constraint	Constant Irrigation	Q-learning (λ=0.1) $\alpha = 0.6$, γ=0.8	Q-learning (λ=0.1) $\alpha = 0.6$, γ=0.6	Q-learning (λ=0.1) $\alpha = 0.8$, γ=0.6
College Station	Maximum Plant Yield	8067	8066	8060	8051
	Total Irrigation	2400	1800	1897	1832
Cambridge	Plant Yield	8032	8031	8026	8022
	Total Irrigation	2400	1860	1920	1964
Iowa Falls	Plant Yield	8164	8162	8151	8147
	Total Irrigation	2400	1868	1897	1980

From the above Figures 6, 7 and 8 and Table 2 shown above we can see that the proposed method works best when learning rate is 0.6 and discount factor is 0.8 and consumes less water in comparison to other sets of learning rate and discount factor values for all the three regions. For all three sets of learning rate and discount factor λ =0.1 gives the best results i.e. the reinforcement learning irrigation system which is highly dependent on leaf area index and not on water consumption gave the best results. As the value of λ is increased, there is slight decrease in water consumption as well as leaf area index. So the value to λ to be chosen for the system depends on the cost of irrigation vs the increment in the leaf area index. In areas with high cost of irrigation, higher value of λ is chosen in comparison the areas with less cost of irrigation. The proposed reinforcement learning irrigation system correctly finds the estimated maximum plant yield with more than 40% water savings in comparison to traditional methods.

Plots containing LAI vs water consumption for weather data of College station, TX with three different sets of learning rate and discount factor on our irrigation control system along with constant irrigation system is shown in Figure 9.
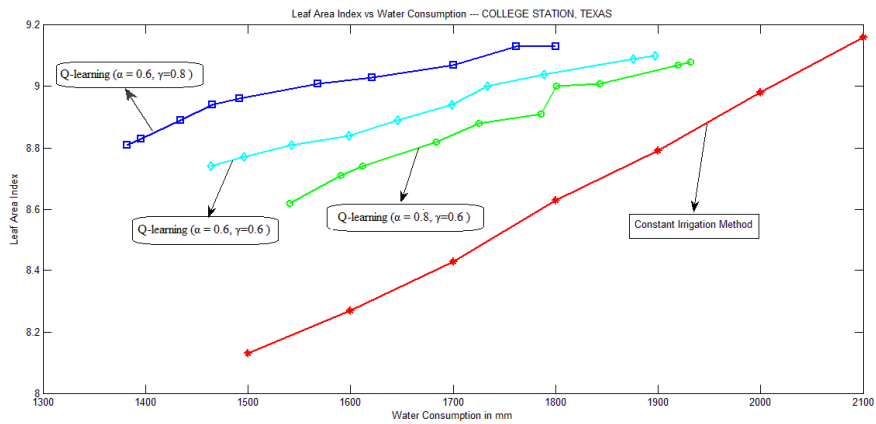

Figure 9: Leaf Area Index vs Water Consumption Plots for College Station, Texas.
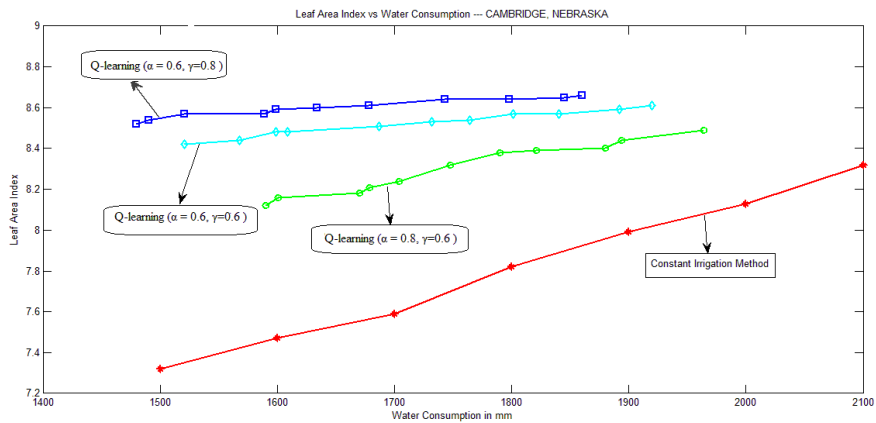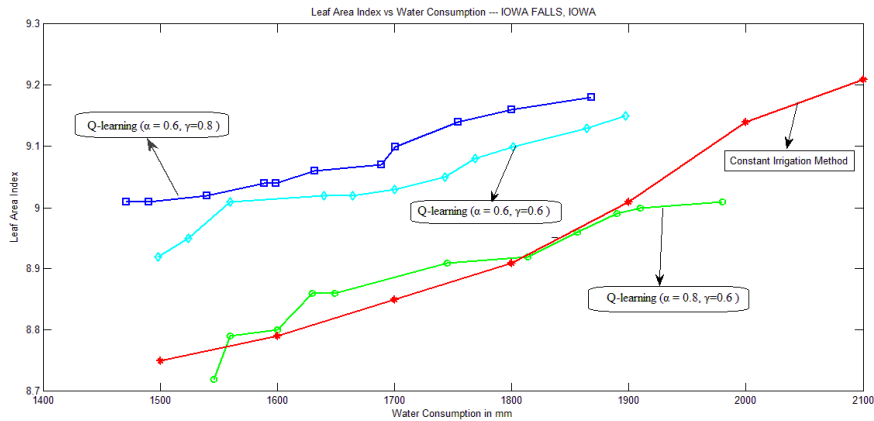

Figure 10: Leaf Area Index vs Water Consumption Plots for Cambridge, Nebraska.

Plots containing LAI vs water consumption for weather data of Cambridge, Nebraska with three different sets of learning rate and discount factor on our irrigation control system along with constant irrigation system is shown in Figure 10.

41

Plots containing LAI vs water consumption for the weather data of Iowa Falls, IOWA with three different sets of learning rate and discount factor on our irrigation control system along with constant irrigation system is shown in Figure 11.

Figure 11: Leaf Area Index vs Water Consumption Plots for Iowa Falls, IOWA.

Table 3: Leaf Area Index Comparison Table for Different Regions

Region	Constraint	Constant Irrigation	Q-learning (λ=0.1) α= 0.6, γ=0.8	Q-learning (λ=0.1) α= 0.6, γ=0.6	Q-learning (λ=0.1) α= 0.8, γ=0.6
College Station	Leaf Area Index	9.16	9.15	9.12	9.08
	Total Irrigation	2400	1800	1897	1932
Cambridge	Leaf Area Index	8.70	8.68	8.64	8.49
	Total Irrigation	2400	1860	1920	1964
Iowa Falls	Leaf Area Index	9.21	9.19	9.16	9.09
	Total Irrigation	2400	1868	1897	1980

From the above Table 3 and Figures 9, 10 and 11 we can confer that the proposed

Q-learning method is very efficient in terms of water consumption. All the weather data

for the three regions in Figures 6 to 11 are taken from US Climate Data Website [US

Climate Data]. In Figure 6/Figure 9 weather data for College Station is taken from April

to June and the data for Cambridge in Figure 7/Figure 10 is taken from months May to

July and the data for Iowa falls in Figure 8/Figure 11 is taken from months June to August.

The controller is able to achieve the maximum leaf area index constraint with less water

consumption for the growth period. Similar to the plant yield from Table 1, Q-learning

irrigation system with learning rate (α) = 0.6 and discount factor (γ) = 0.8 yields the best

possible outcome with all three regions.

8. CONCLUSION

Water plays an important role in plant growth and development. With the depleting underground water resources and other fresh water resources available for irrigation, water management techniques such as precision irrigation is very necessary. In this thesis a fully functional reinforcement learning controller compatible with the DSSAT model is developed. The proposed controller leads to crop yield similar to constant rate irrigation with almost 40% decrease in water consumption. It is evident from the data shown in tables 1 and 2 that the Q-learning irrigation system works best with learning rate of 0.6 and discount factor of 0.8. Reward function introduced in the work which depends on leaf area index as well as water consumption works as expected which helped the reinforcement learning controller achieve desired yield and leaf area index with minimum possible water consumption.

REFERENCES

Anapalli, S.S., Ma, L., Nielsen, D.C., Vigil, M.F., and Ahuja, L.R. (2005) "Simulating planting date effects on corn production using RZWQM and CERES-Maize models". Agronomy Journal, 97:58–71.

Bronson, K.F., Booker, J.D., Bordovsky, J.P., Keeling, J.W., Wheeler, T.A., Boman, R.K., Parajulee, M.N., Segarra, E. and Nichols, R.L. (2006) "Site-specific irrigation and nitrogen management for cotton production in the southern high plains". Agronomy Journal, 98:212-219.

Camp, C.R. and Sadler, E.J. (1998) "Site-specific crop management with a center pivot". Journal of Soil and Water Conservation, 53: 312-315.

Camp, C.R., Sadler, E.J. and Evans, R.G. (2006) "Precision Water Management: Current Realities, Possibilities and Trends. Handbook of Precision Agriculture". A. Srinivasan (ed), Binghamton, NY, Food Products Press.

Chapman, D. (1994) "Water Quality Assessments". World Health Organization.

Chapman, M., L. Chapman, L. and Dore, D. (2008) "National Audit of On-Farm Irrigation Information Tools". Final Report prepared for the Australian Government Department of Environment, Water, Heritage and the Arts through the National Water Plan. http://www.environment.gov.au/system/files/resources/2729e0e5-9403-4f98-8da7-6ff4109cb950/files/irrigation-information-tools.pdf.

Clouse, R.W. (2006) "Spatial Application of a Cotton Growth Model for Analysis of Site-Specific Irrigation in the Texas High Plains". PhD dissertation, Texas A&M University.

DSSAT (Decision Support System for Agrotechnology Transfer). (2015) www.dssat.net/about. Web. 25 May, 2015.

Duke, H.R., Buchleiter, G.W., Heermann, D.F. and Chapman, J.A. (1997) "Site specific management of water and chemicals using self-propelled sprinkler irrigation systems". In: Proc. of 1st European Conference on Precision Agriculture. Warwick, England, UK.

Dukes, M.D. and Scholberg, J.M. (2004) "Automated Subsurface Drip Irrigation Based on Soil Moisture". ASAE Paper No. 052188.

Evans, R.G., Han, S., Kroeger, M.W., and Schneider, S.M. (1996) "Precision center pivot irrigation for efficient use of water and nitrogen". Precision Agriculture, Proceedings of the 3rd International Conference, ASA/CSSA/SSSA, Minneapolis, Minnesota, June 23-26, p75-84.

Evans, R.O., Barker, J.C., Smith, J.T., Sheffield, R.E. (1997) "Center Pivot and Linear Move Irrigation System" (PDF). North Carolina Cooperative Extension Service, North Carolina State University. Retrieved June 6, 2012.

FNR. (2008) "Growing Rice Where it has Never Grown Before: A Missouri research program may help better feed an increasingly hungry world". College of Agriculture, Food and Natural Resources (FNR), University of Missouri. Retrieved June 6, 2012.

Goodwin, I. and O'Connell, M.G. (2008). "The Future of Irrigated Production Horticulture – World and Australian Perspective". Acta Horticulturae, 792, 449–458.

Hedley, C.B. and Yule, I.J. (2009a) "A method for spatial prediction of daily soil water status for precise irrigation scheduling". Agricultural Water Management, 96(12): 1737-1745.

Hedley, C.B. and Yule, I.J. (2009b) "Soil water status mapping and two variable-rate irrigation scenarios". Precision Agriculture, 10: 342-355.

Hoogenboom, G., Jones, J.W., Wilkens, P.W., Porter, C.H., Boote, K.J., Hunt, L.A., Singh, U., Lizaso, J.L., White, J.W., Uryasev, O., Royce, F.S., Ogoshi, R., Gijsman, A.J., Tsuji, G.Y. and Koo. J. (2012) "Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.5 [CD-ROM]". University of Hawaii, Honolulu, Hawaii.

IBSNAT (International Benchmark Sites Network for Agrotechnology Transfer). (1993) The IBSNAT Decade. Department of Agronomy and Soil Science, College of Tropical Agriculture and Human Resources, University of Hawaii, Honoluly, Hawaii.

Irrigation Models. (2015) http://en.wikipedia.org/wiki/Center_pivot_irrigation#cite_note-FencePost-1. Web. 24 May, 2015.

Jones, J.W., Hoogenboom, G., Porter, C.H., Boote, K.J., Batchelor, W.D., Hunt, L.A., Wilkens, P.W., Singh, U., Gijsman, A.J., Ritchie, J.T. (2003) "The DSSAT cropping system model". European Journal of Agronomy 18 235-265.

Jones, J.W., Tsuji, G.Y., Hoogenboom, G., Hunt, L.A., Thornton, P.K., Wilkens, P.W., Imamura, D.T., Bowen, W.T., Singh, U. (1998) "Decision support system for agrotechnology transfer; DSSAT v3". In: Tsuji, G.Y., Hoogenboom, G., Thornton, P.K. (Eds.), "Understanding Options for Agricultural Production". Kluwer Academic Publishers, Dordrecht, the Netherlands, pp. 157-177.

Jones, C.A., and Kiniry, J.R. (1986) "CERES-Maize: A simulation model of maize growth and development". Texas A&M Univ. Press, College Station.

Kincaid, D.C. and Buchleiter, G. (2004) "Irrigation, Site-Specific". Encyclopedia of Water Science, 10.1081/E-EWS 120010137.

King, B.A. and Kincaid, D.C. (1996) "Variable flow sprinkler for site-specific water and nutrient management". ASAE Paper No 962074, St Joseph, MI.

King, B.A. and Kincaid, D.C. (2004) "A variable flow rate sprinkler for site-specific irrigation management". Applied Engineering in Agriculture, 20(6): 765-770.

King, B.A., Stark, J.C. and Wall, R.W. (2006) "Comparison of Site-Specific and Conventional Uniform Irrigation Management for Potatoes". Applied Engineering in Agriculture, 22(5), 677-688.

Lizaso, J.I., Boote, J.K., Jones, J.W., Porter, C.H., Echarte, L., Westgate, M.E., Sonohat, G. (2011) "CSM-IXIM: A New Maize Simulation Model for DSSAT Version 4.5". Agron.J. 103:766-779.

Mader, Shelli. (2010) "Center pivot irrigation revolutionizes agriculture". The Fence Post Magazine. Retrieved June 6, 2012.

Mastrorilli, M., Katerji,N. and Ben Nouna, B. (2003) "Using the CERES-Maize model in a semi-arid Mediterranean environment: Validation of three revised versions". Eur. J. Agron. 19:125–134.

O'Shaughnessy, S.A. and Evett, S.R. (2008) "Integration of wireless sensor networks into moving irrigation systems for automatic irrigation scheduling". Proceedings of the American Society of Agricultural and Biological Engineers International, Providence, Rhode Island, Paper No.083452.

O'Shaughnessy, S.A., Evett, S.R., Coliazzi, P.D and Howell, T.A. (2008) "Soil water measurement and thermal indices for center pivot irrigation scheduling". Irrigation Association Conference Proceedings Anaheim, California.

Papajorgji, P., Beck, H.W., Braga, J.L. (2004) "An architecture for developing service-oriented and component-based environmental models". Ecological Modelling 179 (1), 61-76.

Paz, J.O, Batchelor, W.D. and Tylka, G.L. (2001) "Method to use crop growth models to estimate potential return for variable-rate management in soybeans". Transactions of the ASAE, 44(5): 1335-1341.

Peters, R.T. and Evett, S.R. (2004) "Complete center pivot automation using the temperature-time threshold method of irrigation scheduling". 2004 ASAE/CSAE Annual International Meeting, Ottowa, Ontario, Canada, Paper No. 042196.

Peters, R.T. and Evett, S.R. (2005) "Mechanized irrigation system positioning using two inexpensive GPS receivers". 2005 ASAE Annual International Meeting, Tampa, Florida, Paper No. 052068.

Peters, R.T. and Evett, S.R. (2007) "Spatial and temporal analysis of crop stress using multiple canopy temperature maps created with an array of center-pivot-mounted infrared thermometers". Transactions of the ASABE, 50(3): 919-927.

Peters, R.T. and Evett, S.R. (2008) "Automation of a center pivot using the temperature-time-threshold method of irrigation scheduling". J. Irrig. Drain. Engr., 134: 286-291.

Porter, C.H., Braga, R., Jones, J.W. (1999) "An Approach for Modular Crop Model Development". Research Report No 99-0701, Agricultural and Biological Engineering Department, University of Florida, Gainsville, Florida.

Raine, S.R., Meyer, W.S., Rassam, D.W., Hutson, J.L. and Cook, F.J. (2007) "Soil-water and solute movement under precision irrigation: knowledge gaps for managing sustainable root zones". Irrigation Science, 26(1): 91-100.

Rainfine Irrigation Co.LTD. (2015)
http://i00.i.aliimg.com/photo/v2/1860849058_2/farm_irrigation_machine_of_center_pivot.jpg\. Web. 23 May, 2015.

RL (Reinforcement Learning). (2015)
http://en.wikipedia.org/wiki/Reinforcement_learning. Web. 24 May, 2015.

Sadler, E.J., Bauer, P.J., Busscher, W.J. and Millen, J.A. (2000) "Site-specific analysis of a droughted corn crop: II". Water use and stress. Agronomy Journal, 92(3), 403-410.

Sadler, E.J., Evans, R.G., Buchleiter, G.W., King, B.A. and Camp, C.R. (2000) "Design considerations for site specific irrigation". Proceedings of the 4th Decennial National Irrigation Symposium, p. 304–315.

Shah, N. G. and Ipsita Das. (2012) "Precision Irrigation: Sensor Network Based Irrigation, Problems, Perspectives and Challenges of Agricultural Water Management", Dr. Manish Kumar (Ed.), ISBN: 978-953-51- 0117-8, InTech, Available from: http://www.intechopen.com/books/problems-perspectives-and-challenges-ofagricultural-water-management/precision-irrigation-sensor-network-based-irrigation-to-improve-water-useefficiency-in-agriculture.

Shen, H., Tan, Y., Lu, J., Wu, Q., and Qiu, Q. (2013) "Achieving autonomous power management using reinforcement learning". ACM Trans. Des. Autom. Electron. Syst. 18, 2, Article 24 (March 2013), 32 pages. DOI: http://dx.doi.org/10.1145/2442087.2442095.

Smith, R. J., Baillie, J. N., McCarthy, A. C., Raine, S. R., Baillie, C. P. (2010) "Review of precision irrigation technologies and their application". National Centre for Engineering in Agriculture Publication 1003017/1, USQ, Toowoomba, November.

51

Sutton, R.S., Barto, A.G. (1998) "Reinforcement Learning: An Introduction", MIT press, volume 28.

Tojo-Soler, C.M., Sentelhas, P.U., and Hoogenboom, G. (2007) "Application of the CSM-CERES-maize model for planting date evaluation and yield forecasting for maize grown off -season in a subtropical environment". European J. Agronomy. 27:165–177.

Tsuji, G.Y. (1998) "Network management and information dissemination for agrotechnology transfer". In: Tsuji, G.Y., Hoogenboom, G., Thornton, P.K. (Eds.), Understanding Options for Agricultural Production. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 367-381.

Uehara, G., Tsuji, G.Y., (1998) "Overview of IBSNAT". In: Tsuji, G.Y., Hoogenboom, G., Thornton, P.K. (Eds.). "Understanding Options For Agricultural Production". Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 1-7.

US Climate Data. (2015) Available at: http://www.usclimatedata.com/climate/college-station/texas/united-states/ustx2165. Web. 20 April, 2015.

USDA, NAL, ddr.nal.usda.gov. Omary, M., Camp, C.R. , Sadler, E.J. (1997) "Center pivot irrigation system modification to provide variable water application depths". Applied engineering in agriculture Mar 1997. v. 13 (2).

Valley Irrigation. (2015) http://az276019.vo.msecnd.net/valmontstaging/irrigation---optimized-images/faq_2_rotator_web.jpg. Web. 15 May, 2015.

Watkins, C.J.C.H., Dayan, P. (1989) "Q-Learning". Machine Learning 8 (3), 279-292

Yule, I.J., Hedley, C.B. and Bradbury, S. (2008) "Variable-rate irrigation". 12th Annual

Symposium on Precision Agriculture Research & Application in Australasia. Sydney.