University of Southern Queensland

Faculty of Health, Engineering & Sciences

Adaptive Markov Decision Control Of High-Frequency Drip Irrigation Systems

Volume 1

A dissertation submitted by

Damian Peckett

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Mechatronic Engineering

Submitted: October, 2013

Abstract

Global demand for food increases every year, despite this increase, productive farming land is limited. Meeting the demands of the future are dependent on improving the productivity of the land in use and a key factor behind this is intelligent agriculture. Fresh water remains in many situations the primary constraint on agriculture. This paper seeks to investigate machine intelligence irrigation scheduling methods in order to increase yield and improve crop irrigation efficiency.

For the purposes of research the model organism *Arabidopsis Thaliana* was selected primarily due to its rapid life cycle and historical significance. Research began with the development of a water balance crop model for emphArabidopsis Thaliana. Upon completion of the model several experiments were performed to weakly verify its behaviour.

The final step looked at implementing an artificially intelligent control algorithm for maximizing crop yield in water constrained environments. Several possible approaches were identified. The winning approach utilized a Markov decision process implementing a Radial Basis Function network as a value estimator. Performance was seen to exceed the best hand coded algorithms by up to 10% and it successfully outperformed the normal irrigation schedule control by a factor of 6.5x far exceeding the initial goals of the project!

This solution was developed using a highly optimized, high performance computing MDP solver programmed by the author. Utilizing an Amazon Cluster Compute server possessing 32 Intel Xeon cores the author was able to outperform an initial naive, yet straight from the textbook, MATLAB MDP solver by a factor of 10^4!

University of Southern Queensland

Faculty of Health, Engineering & Sciences

ENG4111/2 *Research Project*

Limitations of Use

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Damian Peckett

0061006271

Signature

Date

Acknowledgments

Firstly I'd like to acknowledge my supervisor Dr Alison McCarthy. It was her tick of approval all those months ago that made this project a reality and for that I am utterly grateful. I'd also like to thank my co-supervisor Dr Nigel Hancock. It was his pushing in the early stages that was so key to bringing this project to completion.

I'd like to thank my grandfather John Ward for his resource support during the early stages of the experiment work. Without his support the project would have taken a far less ambitious experimental approach. It's this ambition and overreaching that is so important to science, in the words of the brilliant science fiction writer Arthur C. Clarke, *"The only way of discovering the limits of the possible is to venture a little way past them into the impossible."*

I'd like to finally acknowledge my girlfriend Sophia. Her support throughout the many mood swings of such a personally invested project was invaluable to keeping me level-headed and on track. As always, thank you!

A sad message to anyone reading this, I completely ran out of time, I'd love to have expanded further and been more thorough in my write up but alas, sadly 24 hours a day was a bit short.

Damian Peckett

University of Southern Queensland

October 2013

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Global demand for food is expected to rise by over 70years (New York Times 2009). Over one billion individuals individuals already suffer the effects of food scarcity (New York Times 2009). Despite this increase in demand, productive agricultural land remains constrained by the limits of the earth's biosphere. Meeting the demands of the future are dependent on improving the productivity of the land in use and a key factor behind achieving this is intelligent agriculture.

In many regions the primary limiting factor behind crop yield is water scarcity, Over 3 billion people live in regions affected by water scarcity, be that physical or economic in origin (United Nations 2007). By improving the efficiency of the utilisation of water in these regions one can directly tackle one of the key problems behind world hunger.

The stellar rise of consumer computing power over the last few decades has brought about the age of "big data." Big data as a concept refers to the automated analysis of vast sources of data. Vast sources of data beyond the scope of any humans ability to comprehend. This paper investigates the application of these big data approaches to optimising crop water usage in highly dynamic and demanding environmental conditions.

The remainder of this document will seek to develop an "intelligent" control alogrithm for the task of water efficient drip irrigation. The combining of the strengths of the two systems shall result in new opportunities and novel ideas.

Chapter 2

Literature Review

2.1 Crop Modelling

2.1.1 Crop Model History

The use of models to predict the behavior of crop growth to abiotic stressors has a long history within the literature. The early work of CT Dewit (1965) represented some of the first attempts at modeling plant behavior. CT Dewit's early work concentrated on accurately predicting the photosynthetic rate of plant canopies. Predicting plant photosynthetic rate is a nontrivial exercise and CT Dewits research despite its "pioneering" (Yin & Struik 2009) nature was highly sophisticated, taking into account complex factors such as crop leaf distribution and the effects of restricted carbon dioxide exchange.

The latter work of CT Dewit (1978) expanded into the development of sophisticated models for whole organism behavior. These models explored the dynamics of processes such as transpiration and biomass accumulation. CT Dewit's work focused on predicting the yield of maize. Further work by DA Charles-Edwards and MJ Fisher (1980) investigated the use of these analytical methods in conjunction with empirical observations to successfully develop models for the legume pasture crop *Stylosanthes Humilis.*

Continued development was made over the years and modern crop models are highly sophisticated simulators of crop behavior. Many particular models exist, one such example being APSIM (B.A Keating et al. 2003). APSIM was developed by a joint venture of CSIRO and Queensland state government agencies (B.A Keating et al. 2003). The APSIM model simulates crop response to a variety of abiotic conditions and models parameters such as soil water balance, nitrogen/phosphorous balance and crop yield. Significant research commitment was involved in APSIMs development and it is released under a limited restrictive licensing scheme. Modules exist that allow APSIM to simulate the behavior of a variety of legumes (M.J. Robertson et al. 2002), grasses (B.A Keating et al. 1999, Keating & Agriculture 2001) and cotton (Hearn & Roza 1985) crops.

Despite APSIM's wide range of modules, the *Brassicaceae* family is significantly underrepresented. The only example for the *Brassicaceae* Family identified was developed by MJ Robertson et al. (1999) for modelling the yield of the commercial crop canola (*Brassica Napus L*).

2.1.2 Research Potential

Despite the historical use of crop models to predict commercial crop yields their potential versatility cannot be understated. KJ Boote et al. (1995) present a variety of scenarios where the versatility of crop models can be applied to bettering research understanding.

With the utility of crop models in a research setting being established one must consider an organism on which to base the model upon. As established earlier, there is a long history of using crop models to forecast the performance of a variety of legume (CHARLES-EDWARDS & FISHER 1980) and maize species (de Wit 1978). However the results of MJ Fisher (1969) indicate a life cycle of roughly eleven weeks until flowering for the legume species *Stylosanthes Humilis* and similarly the research of T Farrell and K O'Keefee (2007) suggests an average time until maturity of roughly 15 weeks for a collection of maize varieties.

In the context of a laboratory setting these relatively long maturation life cycles are detrimental to an experimental design. Combined with the large physical size of these species they prove poor candidates for most experiments.

2.1.3 Introducing *Arabidopsis Thaliana*

In the literature, the organism *Arabidopsis Thaliana* proves to be highly utilized and experimented upon. *Arabidopsis Thaliana* has a long history in the biology community and generally it is considered a model organism (D.W. Meinke et al. 1998). Model organisms are species that are thoroughly researched in order to understand more generalized biological behaviors. EM Meyerowitz (2001) states that the first person to suggest using *Arabidopsis* as a model organism for the Angiosperm kingdom was Friedrich Laibach in 1943.

F. Laibach (cited in EM Meyerowitz 2001, p. 1) suggested that *Arabidopsis Thaliana* be considered as a model organism on the basis of its short maturation time, as short as six weeks in some strains (D.W. Meinke et al. 1998), ease of propagation and it's propensity to mutagenesis experiments.

MH Hoffmann (2002) suggests that in wild, *Arabidopsis Thaliana* is native to western Eurasia, being found in primarily cool climates, with an average winter temperature of roughly negative four degrees Celsius and a summer temperature of roughly eighteen degrees. It grows primarily in sandy and loamy soils up to a maximum altitude of roughly two thousand meters above sea level. However in the presence of plentiful precipitation it has been observed to grow significantly outside of this temperature range (Hoffmann 2002). This climatic background lends itself easily to laboratory conditions being often propagated at a high humidity, constant twenty two degrees Celsius (Hoffmann 2002).

2.1.4 Application To Crop Understanding

While *Arabidopsis* may represent an ideal laboratory species it has little to no direct significance as a commercial crop (Meyerowitz 1987). However this lack of direct application is not a hindrance to useful research. As the words of YH Chew and KJ Halliday (2010 p. 1) state: *"The substantial expansion in our knowledge of abiotic stress tolerance and avoidance strategies in Arabidopsis provides increased potential for exploitation in crops."* By improving the understanding of stress in the Arabidopsis Thaliana organism one can improve the understanding of crop stress as a generalized whole.

However a counter view is illustrated in the paper *"Abiotic Stress Tolerance in Grasses. From Model Plants to Crop Plants"* by M Tester and A Bacic (2005). Applying insights gained through the study of *Arabidopsis* is often difficult to apply to many commercial crops. M Tester and A Bacic explain this is often due to fundamental structural differences between *Arabidopsis* physiology and that of most major crops.

Arabidopsis is an example of a dicotyledonous species (van Dodeweerd 1999), in that it possesses two embryonic leaves at germination (cotyledons). Many major commercial crops are monocotyledons, examples primarily belonging to the grass family (van Dodeweerd 1999), rice, maize/corn, wheat, barley, etc. These monocotyledons are generally far less succulent than their dicot counterparts (Tester & Bacic 2005). The observations of EM Meyerowitz and CR Somerville (1994) indicate an average dry matter ratio of approximately eight percent for the *Arabidopsis Thaliana* species. This represents that roughly 92 percent of the total plant mass is compromised as water. As an example of a monocotyledonous species, the results of OAM Lewis et al. (1988) indicate a moisture content of roughly 87 percent for common maize species. This represents a difference of biomass content of roughly 50 percent per unit fresh weight.

Comparing salinity tolerance between monocots and dicots is generally unsuccessful (Tester & Bacic 2005). Due to the differences in moisture content, salt accumulates very differently between the groupings. Dicotyledonous species generally accumulate greater quantities of sodium ions in their shoots than their monocot counterparts (Tester & Bacic 2005). This ability to exclude sodium from the growth shoot is an important determinant of salinity tolerance (R. Jaarsma et al. 2013). Due to these fundamental differences applying information learned from *Arabidopsis* salinity stress experiments to grass/cereal species is non-ideal.

However not all commercial crops belong to the grass family, examples of dicotyledonous commercial crop families include: Legumes, *Cucurbitaceae*, *Brassicaceae* and *Solanaceae* (R. Jaarsma et al. 2013). Applying an understanding of salt tolerance gained from Arabidopsis experimentation to these significant crop families is potentially a useful asset.

However the research described herein is primarily focused on the effects of water stress. The work of K Nakashima et al. (2009) demonstrates that *Arabidopsis Thaliana* and the monocot cereal crop rice (*Oryza Sativa*) share common genomic regulatory mechanisms in response to osmotic and heat stressors. This water stress regulatory mechanism being primarily mediated by an increase Abscisic acid (ABA) concentration (D. Aimar et al. 2011). This common mechanism is shared between dicots, monocots and cereal crops alike (D. Aimar et al. 2011).

Abscisic acid, through a complex chemical cascade, is primarily involved in stimulating the action of stomata guard cells on the leaf surfaces (Z.M. Pei et al. 2011, G. Jakab et al. 2005). These guard cells act to decrease the stomatal conductivity. This decrease in stomatal conductivity is accompanied by a corresponding decrease in photosynthetic rate.

This reduction in photosynthetic rate was investigated in the work of WJS Downton et al. (1987). The reduction in photosynthetic rate can be entirely explained by the reduction in carbon dioxide exchange. On the basis of this phenomenon one can suggest that the relationships of transpiration loss will be similar between both grass species and the dicots (including *Arabidopsis*).

However *Arabidopsis* utilizes a C3 photosynthetic pathway (J.K. Ward et al. 2000) and many of the grass species utilize the C4 pathway (Giussani 2001). The C4 photosynthetic pathway is more efficient in limited stomatal conductivity environments (Morison & Gifford 1983). One therefore may expect C4 plants to accumulate greater biomass in a water stressed environment than the C3 *Arabidopsis* species. However most crops utilize the C3 pathway therefore an understanding of the water stress response of *Arabidopsis* can be well generalized.

2.1.5 Existing *Arabidopsis Thaliana* Modelling

A variety of the key regulatory mechanisms of *Arabidopsis* have been computer modelled over the years, one such prominent model being developed by AM Wilczek et al. (2009). AM Wilczek et als' model looked at analysing the life cycle behaviour of a collection of *Arabidopsis Thaliana* species. This life cycle analysis focused on predicting the number of days until flowering based upon input parameters such as the genetic variant and photothermal development units. These photothermal development units being based upon a calculated model of day time temperature and solar insolation. AM Wilczek et als' model was able to successfully predict over ninety two percent of the overall observed variance.

Another regulatory mechanism to be modelled was the distribution of the growth hormone Auxin within the *Arabidopsis* shoot tip (P.B. de Reuille et al. 2006). PB de Reuille et als' model investigated the Auxin flux within of a growing shoot apex. Through their modelling they were able to identify several key components to the distribution which were at the time previously undiscovered (P.B. de Reuille et al. 2006).

Further individual mechanism models being developed by, L Mendoza et al. (1998), GD Bilsborough et al. (2010) and AN Dodd et al. (2006). L Mendoza et als' model looked at the genetic regulatory mechanisms behind *Arabidopsis* flower morphogenesis. The differentiation of the cells that make up *Arabidopsis's* flower structure depend primarily on eleven key genes. These genes act upon each another through a feedback network to form one of five key states. L Mendoza et als' model successfully predicted the existence of six differentiated states. The sixth state being one not seen in nature but could be induced experimentally.

GD Bilsborough et al. investigated a peculiar phenomenon that occurs in the development of *Arabidopsis* leaves. As the leaves of the *Arabidopsis* plant mature they form regular protrusions along the leaf boundary (serrations). The mechanism behind this change in leaf shape previously being unexplained. GD Bilsborougha et al. modelled the interactions of the growth hormone Auxin and the CUC2 transcription factor. By modelling the two feedback loops that operate upon the process they were able to develop a model hat exhibited and explained this peculiar property of leaf maturation (G.D. Bilsborough et al. 2011).

AN Dodd et als' research looked at modelling *Arabidopsis* guard cell behaviour in the presence of cold induced stress. By building a model of an individual cells response and then summing together a colony of them (an analogue of the greater guard cell structure). They were able to accurately model calcium signalling within the stomata. Their results being confirmed experimentally (A.N. Dodd et al. 2006).

While the literature turned up a variety of previous attempts at modelling the various subsystems of the *Arabidopsis* organism no previous examples of modelling at a macro scale were identified. Also few existing crop models for the greater *Brassicaceae* family were identified as a generality. Those existing models being designed primarily for

Canola (M.J. Robertson et al. 1999) and being encumbered by restrictive licensing and bulky legacy codebases.

2.1.6 Conclusion

At this time the literature lacks a simple crop model tailored toward investigating *Arabidopsis* water stress response. By improving the understanding of *Arabidopsis* water stress response one can generate generalized conclusions for many commercial crops.

2.2 Applying Machine Learning To Irrigation Control

2.2.1 Irrigation Control History

Up until the early 1970's much of the developed agricultural community relied on using static unchanging irrigation schedules based upon empirical observations and historical results (Jensen 1972). Evaluations of irrigation practices by OW Israelsen et al. (1944) suggested this approach had remained virtually unchanged in over twenty five years.

ME Jensen (1972) suggested a variety of reasons as to why this static inefficient approach remained highly utilized by farmers. For a long time the cost of irrigation water had been considered rather insignificant and the indirect costs caused by nutrient run off and yield reductions had been difficult to quantify.

Also many of the damages caused by poor water utilization are not directly apparent to the farmers causing them. Run off from upper area irrigators' caused soil and crop yield damages to the farmers operating in the lower laying regions (Jensen 1972). This lack of financial recourse combined with poor regulation lead to limited incentives to improve crop irrigation practices.

However increases in the cost of water and land in the late 1940's brought about an interest in optimizing water usage and increasing crop yield (O.W. Israelsen et al. 1944). By the late 1960's with the introduction of cheap computing power (Jensen 1972) and the theoretical work of HL Penman (1952) in modelling crop water usage the time had arisen for the introduction of improved irrigation practices.

HL Penman's (1952) pioneering work was on modelling the evapotranspiration demand of crops. He achieved this by deriving a set of equations for the water demand of a reference crop and then developed a coefficient based approach for generalizing the suggested values for different species. The pioneering concept of his 1952 paper was the introduction of the dynamic behaviour of stomatal conductivity. This work was used in the early 1970's to create computer generated irrigation forecasts for many large areas of farmland (Jensen 1972).

2.2.2 Soil Water Methods

Despite the introduction of computerised irrigation forecasts, crop irrigation remained a manual and labour intensive process. The computer forecasts were used by farmers to supplement their existing irrigation techniques (Jensen 1972). However with the advent of the microchip and the introduction of new sensor technologies the introduction of automated approaches was soon to be apparent.

One such key sensor technology being developed by CJ Phene et al. (1971). Prior to 1971 measuring soil water content automatically was a difficult task (C.J. Phene et al. 1971). At the time the leading method of observing soil moisture was through the use of a device known as a tensiometer. The tensiometer uses a pressure differential across a porous membrane to evaluate current soil water matric potential.

At the time of the early 1970's the existing pressure transducer technologies exhibited poor performance and accuracies in the range of soil matric pressures expected under many field conditions (C.J. Phene et al. 1972). CJ Phene et al. were able to overcome these limitations with the design of a highly novel sensor topology.

Rather than relying on properties such as soil resistivity or the pressure differential across a membrane, CJ Phene et als' design relied on the thermal dissipation of a porous block buried in the soil. By using such an indirect measurement style they were able to overcome complicating factors such as soil texture, salinity and temperature (Phene & Howell 1982).

This novel sensor design led to the development of one of the first automated crop irrigation approaches (C.J. Phene et al. 1972). CJ Phene et al. (1972) were able to apply this new sensor technology in the development of an automated irrigation system that relied on the soil matric potential and hence the soil water content. The system was configured as a bang-bang controller with every time the soil matric potential fell below a pre-set threshold a drip irrigation system was triggered which applied a fixed amount of water to the field. The system was successfully tested in September 1971 on several plots of corn (*Zea Mays*).

CJ Phene et al. (1982) revisited the thermal disspation sensor approach in 1982, with the introduction of new high precision microchip temperature sensors it was possible to greatly improve the accuracy and response time. A field trial using this new design was performed on the commercial tomato species (*Lycopersicon Esculentum* Mill, UC82B). As a control, the trial utilized the irrigation forecast produced by HL Penman's (1952) equations. CJ Phene et als' (1982) trial was able to achieve comparable results to the control using evapotranspiration predictions, however this was achieved without significant human interaction and considerably fewer data inputs.

More recent work by S Dabach et al. (2011) looked at comparing the results of a computer model with the real world results of soil moisture based irrigation schedule. Through the use of the HYDRUS soil and crop model it was suggested high efficiency crop irrigation using water status and dynamically calculated irrigation quantities is a distinct possibility. This application of dynamic irrigation quantity is of importance to high efficiencies utilizing soil water matric potential threshold based approaches.

2.2.3 Evapotranspiration Methods

The requirement for variable irrigation quantities for soil matric threshold based approaches is due to the changes in crop evapotranspiration in response to growth and meteorological conditions (R.G. Allen et al. 1998). Crop related factors include but are not limited to: albedo, crop height and ground cover. Primary meteorological factors include solar radiation, air temperature, humidity and wind speed (R.G. Allen et al. 1998). While the listed factors may explain the vast majority of the observed changes in evapotranspiration they are far from complete (Jackson 1985). Most calculations of evapotranspiration continue to rely on empirical "magic number" correction factors (HL Penman 1952).

Though for many practical applications these empirical formulae prove sufficient. However prior to the advent of the PC and the microcontroller the concept of automating the task of evaluating these empirical formulaes at a local scale was somewhat impractical with time share computing and manual data entry required (Jensen 1969). Despite the issues with analytically calculating evapotranspiration responses there existed another method to access the valuable information contained in this process and that was observation (Decker & Skau 1964).

The historical method of observing transpiration was through the measurement of sap flow (Decker & Skau 1964). By applying a pulse of heat to the stem of a plant and measuring how long it takes for the warm sap to reach a thermocouple sensor located further up the stem one can determine an approximation for the rate of sap flow (Y. Cohen et al. 1981).

One example of an application of sap flow transpiration measurements was in a study by CHM van Bavel et al. (1996). CHM van Bavel et al. used an array of four sap flow sensors which were placed on representative specimens from the target crop. When combined with a rain gauge placed on the crop site and a basic soil evaporation model it was possible to calculate an approximate soil water balance without the use of soil moisture sensors. This approach lent itself well to data constrained environments.

Another approach to indirectly observing evapotranspiration came through the monitoring of leaf to air differential temperature (R.D. Jackson et al. 1981). However the differential temperature has to look at greater meteorological conditions before it can be considered useful (Fuchs 1990). Despite the lack of this normalization step, several studies have successfully used a simple threshold based approach (S.R. Evett et al. 1996, S.R. Evett et al. 2000).

This threshold approach used the concept of a thermal kinetic window, there exists a temperature range in which the enzymes involved in photosynthesis for a particular plant species operate optimally (S.R. Evett et al. 1996). Naturally a plant will seek to regulate leaf temperature to within this range, however under environmental stress (eg. soil water depletion) the plant will likely struggle to maintain temperature regulation (Jackson 1982).

The experiments of SR Evett et al. (1996) on maize (*Zea Mays*) indicated through the use of such temperature threshold approaches one could achieve yields and water usage efficiencies comparable to the more complex analytical evapotranspiration predictions of H. Penman's (1952) equations.

2.2.4 Crop Model Methods

The work of HL Penman (1952) provided an excellent frame work for predicting the evapotranspiration demand of many crops. However its basis in empirical correction factors and its purely reactionary approach (eg. transpiration demand could only be predicted in the present) left much room for improvement. This improvement came in the form of computerised "Crop Models". Crop models exist as prediction models for the behaviour of plants (de Wit 1965). By supplying various input parameters such as soil moisture and meteorological conditions a crop model can be used to predict evapotranspiration demand as well as investigate the effects of various irrigation schedules upon biomass accumulation.

This ability to predict the effects of different irrigation schedules on biomass accumulation has been used highly successfully to optimize irrigation techniques in several studies (K.S. Raju et al. 1983, Alison C. McCarthy et al. 2010, J.E. Bergez et al. 2002). The primary focus of many of these studies has been optimizing water usage over absolute yield (J.E. Bergez et al. 2002). This is a distinct strength that is unique to crop model based approaches when compared to traditional evapotranspiration demand modelling.

The work of K.S. Raju et al. (1983) represents some of the earliest applications of crop models to irrigation scheduling the author could identify. K.S. Raju et al. (1983) used a rather unusual approach in their model, rather than using the analytical grounded approach of CT DeWit (1965) the model was based upon a purely empirical observation of the relationship between soil moisture and crop yield in corn (*Zea Mays*). The model operated in a temporally piecewise manner, with each segment of the growth phase independently calculating and represented as a state space control system (Fahrmeir 1999).

This model was validated using independently gathered data. However due to the fact it operated under the assumption that the limiting factor was soil moisture an empirical normalization constant was required to be calculated for each site. By evaluating the crop yield model over a variety of cost structures and sites it was possible to generate a table of optimal irrigation strategies and compare their profitability and expected yields (K.S. Raju et al. 1983).

A slightly different approach was used in a study by JE Bergez et al. (2002). Rather than maximising absolute yield when compared to cost, JE Bergez et al. looked at optimising water efficiency. This water efficiency being calculated by the yield per unit volume of applied irrigation water. This approach was tested using the MODERATO crop model. The MODERATO crop model is more complex than purely predictive, it also contains a decision engine (J.E. Bergez et al. 2001). By supplying irrigation constraints this decision engine can suggest "optimal" irrigation schedules. JE Bergez et al. successfully tested the theoretical performance of the MODERATO decision engine against a basic baseline strategy. The results of the experimentation suggesting favourable outcomes for the crop model approach.

More recent research was performed by AC McCarthy et al. (2010) using the OZCOT crop model. AC McCarthy et als research looked at the application of crop modelling to adaptive irrigation control. Within a crop, water demand varies spatially and temporally in response to a variety of factors (which will be expanded upon later in this review). By independently applying crop modelling to spatially varied subplots AC McCarthy et al. attempted to optimize overall crop efficiency. The results of a case study showed great promise for the use of crop modelling and adaptive irrigation to improve crop yield and water usage.

2.2.5 Machine Learning Methods

Machine learning is a branch of artificial intelligence that deals with computer systems that can learn from data. The simplest example of a machine learning algorithm would be the ever common "line of best fit". By analysing a set of known data the line of best fit can be extrapolated to predict values outside of ranges already seen. It can also be used to interpolate predicted values for gaps between known values. This line of best fit is known as linear (straight line) regression. Regression isn't limited to strictly linear predictions, by scaling the input features one can generate polynomial curves and through the introduction of further variables even hyper-planes (in greater than two-dimensional space) (Ng 2012).

To use regression to generate non-linear fits one must first identify what to scale each variable with. With increases in the number of features (input variables) this problem can rapidly become very difficult. However there exist methods that allow the prediction of non-linear functions without the difficult manually selected scaling functions, the primary two of these methods being Artificial Neural Networks and the newer Support Vector Machines. These methods will only be visited briefly in this review as they are large topics deserving of much greater attention. However the essential basics behind the methods are that they can select non-linear scaling factors on their own without first being identified by a human. Fundamentally at an external black box scale they can be observed to operate similarly to regression.

Crop yield in response to meteorological conditions and irrigation scheduling is a great example of a complex non-linear system (Schlenker & Roberts 2006). Therefore it's apt to say machine learning might possibly be well utilized in the task of irrigation control. Several studies (Q. Zhang et al. 1996, F. Capraro et al. 2008, Karimaldini, F. et al. 2012, P. Mart et al. 2013) have approached this premise with promising results.

Two previous approaches to applying machine learning techniques to irrigation control have been identified in the literature these approaches being fuzzy logic and artificial neural networks.

The earliest identified fuzzy logic approach was by Q Zhang and CH Wu (1996) in 1996. Q Zhang and CH Wu's approach utilized three sensor inputs, air temperature, humidity and soil electrical resistivity. Using one thousand lines of C code they were able to develop a fuzzy logic learning system that dynamically adjusted the fuzzy irrigation thresholds in order to maintain stable soil moisture content when confronted with a growing plant and changing weather conditions. Interesting to note is that the primary sensor used in the trial was a soil resistivity probe.

The best example of an artificial neural network approach is described in the paper *"Neural Network-Based Irrigation Control for Precision Agriculture"* (F. Capraro et al. 2008). F Capraro et al. developed an artificially intelligent SISO (single-input,single-output) irrigation controller. The input to the controller being a soil capacitance sensor located in the root zone of the crop and the output being the a solenoid activation signal. The input features to the artificial neural network included several of the past soil moisture observations along with the current soil moisture. By acting upon these basic cues the controller was capable of responding rapidly and accurately to changing soil properties and environmental conditions.

A study by F Karimaldini et al. (2010) looked at using artificial neural networks to predict daily evapotranspiration in areas with limited meteorological data. When supplied with a daily maximum and minimum temperature and average wind speed the artificial neural network approach was able to outperform all existing evapotranspiration estimations short of those generated by the data intense HL Penman (1952) model.

The second study by P Marti et al. (2013) looked at estimating stem water potential in citrus trees based upon basic environmental parameters and soil moisture observations. Stem water potential is a great predictor of crop stress and evapotranspiration demand (P. Mart et al. 2013). In some ways the behaviour of the artificial neural network could be likened to that of a simple crop model. Despite the simplicity the ANN approach was able to successfully predict stem water potential, and therefore crop stress, with a determination coefficient of $R^2 = 0.926$, a remarkably good fit.

2.2.6 Spatial Variability

Amongst a large field soil properties and crop properties can vary dramatically (Peck 1983). This varying in soil properties can lead to inefficient irrigation schedules if the field is considered as one uniform entity. A study conducted on cotton (Evans 2006) has shown water efficiency can be improved by up to 44 percent in certain cotton crops. Work conducted by AW Warrick and SR Yates (1987) bolsters this theory. The suggested mechanism behind this improvement is that in a water constrained environment it is better to have patches of well watered crop rather than too little water everywhere (Warrick & Yates 1987).

Significant work has been performed by AC McCarthy et al. (2008 and 2010) in developing spatially variant irrigation algorithms. By separating a field into many subplots and running separate instances of the irrigation algorithm. This approach has shown great promise in field testing (Alison C. McCarthy et al. 2010).

2.2.7 Conclusion

In conclusion the background research has presented a variety of historical approaches to automated irrigation scheduling. The recent years have seen the introduction of artificially intelligent approaches however they have been limited to simple sensor inputs and not aware of spatial variation. Given the strength of machine learning when presented with "big data" the author believes there is significant room for improvement and there exists great promise in artificially intelligent adaptive approaches.

2.3 Machine Learning

2.3.1 Reinforcement Learning

Many machine learning applications involve simple classification, taking in a collection of input data variables and producing a single output. This single output could be something like you "probably" have malaria or you probably don't. There exist two or more independent states that can be clearly checked, it is possible to know the correct answer. Systems involving these approaches are trained through feeding back known correct answers and then using the system to extrapolate for cases where the answer is unknown (Ng 2012).

The issue is for many other applications, there is no way of determining the correct answer. A large class of these problems involves optimization problems. Optimization is the task of producing a solution for some system that produces as close to possible the desired response. For agricultural applications this may represent something like the largest possible yield or the most yield per unit of irrigated water.

One may believe and rightly so it might be a better option to develop a conventional linear control algorithm for these basic goals. However by using machine learning we can optimize for incredibly complex and nonlinear reward functions. A non-linear reward function may be an optimal irrigation strategy that achieves the maximum possible yield using a fixed water supply quota. Using computer control to reach such targets is highly novel and a very exciting prospect.

Most machine learning algorithms operate very similar to a form of complex regression (Russell & Norvig 2010). Applying regression natively to an optimization problem is non-ideal as approaches for training the system become complex (Sutton & Barton 1998). Thankfully the solution to the problem of how to solve reinforcement problems was first proposed by the Richard Bellman in 1957 (Bellman 1957). This approach named in honor of Russian mathematician Andrey Markov's work on stochastic processes came to be known as "Markov Decision Processes".

2.3.2 Markov Decision Process

The Markov Decision Process exists as a mathematical framework for treating and solving a variety of reinforcement problems. The name "Markov Decision" refers to the fact that the framework deals with solutions for problems that can be represented as Markov chains (Sutton & Barton 1998).

A Markov chain is a mathematical system that exists as a set of states that transition between one another. The next state depends only on the current state and the current action. There exists no knowledge of historical states and so the system is referred to as being memory less (Sutton & Barton 1998). Systems meeting these criteria are referred to as fulfilling the Markov property (Sutton & Barton 1998).

The vast majority of reinforcement learning problems can be described as Markov Decision Process (Sutton & Barton 1998). Once one has a Markov decision process existing as a discrete set of states and actionable inputs it comes to the problem of solving the system.

Bellmans equations (Bellman 1957) describe a mathematical frame work involving dynamic (Howard 1960) programming to solve the complex task. Belmann's equations seek to assign a value to each possible state. This value is a function of the current state and all the possible future states (Sutton & Barton 1998). Rather than looking simply at the current state and taking an action, Belmann's equations look out into the future and calculate the probability of all future possibilities (Sutton & Barton 1998). This ability to look into the future represents the key strength of the Markov decision framework.

2.3.3 Value Estimation

The traditional Markov decision process involves a discrete number of possible states. In the real world input features are generally continuous. One could split the continuous state space into a discrete number of states however with high dimensional feature spaces this very quickly becomes extremely computationally intensive (Ng 2012).

On the basis of this computation complexity the problem must be reduced in complexity and therefore approximated (Ng 2012, Sutton & Barton 1998). This approach is known as value estimation (Ng 2012). Several techniques exist in order to reduce the complexity of the problem state space; common approaches include regression, course coding, tile coding and radial basis networks (Sutton & Barton 1998).

Regression uses conventional line of best fit approaches, such as gradient descent to attempt to fit some linear combination of parameters to generate an approximation of the value function (Sutton & Barton 1998, Ng 2012). Often tuning such a system and choosing parameters is incredibly difficult (Ng 2012).

The second approach, course/tile coding attempts to discretize the value function by using an adaptive grid (Sutton & Barton 1998). While tile based discretization is impractical it is sometimes possible to break the value function into a discrete set of regions and evaluate any point on the state space using interpolation (Sutton & Barton 1998). This approach however is very noisy, in that it generates abrupt transitions in the value function between adjacent regions (Sutton & Barton 1998). Course coding can also be difficult and time consuming to tune.

The final approach and a personal favorite of the author involves radial basis functions. The use of radial basis functions can be liked to a continuous variant of the course coding approximation method (Sutton & Barton 1998). They retain the property of being centered on regions within the state space however they generate smooth continuous transitions.

2.3.4 Radial Basis Function Networks

Radial Basis Function networks are a special class of artificial neural networks in which the activation function is represented by some Radial Basis Function. There exists a great variety of possible radial basis functions, and in fact almost any function can be used. However common functions include the Linear, Polynomial, Fourier, Gaussian and Inverse Multiquadratic kernels (Recht 2005). Each of these kernel functions possesses its own unique shape and distribution.

By summing together the outputs of a set of Radial Basis Functions one can begin to approximate nonlinear functions. One of these kernels, the Gaussian kernel, possesses the special property of operating in infinite dimensional feature space (Ng 2012). This property comes from the Taylor series expansion of the exponential function. The presence of an infinite dimensional feature space allows for linear combinations to act as universal function approximations (EJ Hartman et al. 1990).

The Gaussian kernel operates on two input parameters, a function centre and a kernel variance. This variance determines how fast or slowly the function asymptotes to zero as the distance from the centre is increased (Ng 2012). In order to approximate a fairly complex function one would need a considerable number of Radial Basis Functions cantered at various key regions in the state space. Each of these Radial basis functions would possess its own variance. This leads to a large number of possible configurations.

However the problem can be simplified, the work of J Park and IW Sandberg (1991) proved that for most cases a single layer network consisting of Gaussian Radial Basis Functions and possessing a fixed universal and uniform variance was capable of producing accurate estimations (Park & Sandberg 1991).

2.4 Conclusion

A wide variety of agricultural control tasks can be phrased as optimization problems. The Markov Decision Process is a mathematical framework that can be used to solve reinforcement problems. However for a contiinuous state space, such as that of a crop in a field, value estimation must be used. Single layer Gaussian Radial Basis Function networks can be utilized as efficient universal value estimators.

Chapter 3

Materials and Methods

3.1 Arabidopsis Thaliana Model

3.2 Machine Learning Algorithm

3.3 Model Validation (Published Observations)

3.4 Model Validation (Field Testing)

3.5 Algorithm Performance (Crop Model)

3.6 Algorithm Performance (Field Testing)

A two stage process was used to develop and investigate the performance of an artificially intelligent irrigation scheduling method. The first stage involved the development of a crop model for the subject organism and the subsequent simulated training of several potential scheduling algorithms.

The second stage investigated the real world performance of the trained algorithms against a current best practices normal irrigation schedule. Performance was scored upon the ratio of leaf area to the quantity of applied irrigation. The experimental trial was performed in the months of July/August 2013 at a test site located on the University Of Southern Queensland campus.

3.6.1 Design Decisions

Background research indentified several key techniques for automated irrigation scheduling. There existed a significant gap in the application of machine learning to irrigation control despite promising experimental results (Q. Zhang and C.H. Wu 1996, F. Capraro et al. 2008, F. Karimaldini et al. 2010 and P. Marti et al. 2012). All the identified approaches employed the use of Artificial Neural Networks, this experiment investigated the performance of several different approaches.

The algorithms were trained with an adaptive reward based learning approach, this was chosen in order to closely respond to the dynamic conditions represented in crop scenarios. In order to supply highly important priori information to the models a plant behavior simulator (crop model) was developed. The use of simulators to supply priori information to machine learning models is a common approach.

Once supplied with priori information the models were trained and tested in an experimental trial. The trial was performed at the University Of Southern Queensland campus (-27.61, 151.93) during the months of July-August 2013. The location and time of testing being chosen on the basis of availability rather than some predefined basis.

The experiment was performed in an open field environment with the specimens placed within half liter pots elevated approximately 30cm above the soil surface. The experimental trial was run as an analogue for shallow rooting specimens. Water was applied through the use of drip irrigators, a common practice in water constrained agriculture.

3.6.2 Subjects

Background research idenitified the model organism Arabidopsis Thaliana as a commonly used labaratory specimen. Due to its role as a model organism it has been extensively studied in the literature. This extensive study is invaluable to the design of the first stage crop simulation model. The background research also identified the Arabidopsis Thaliana organism as a potential analogue for a variety of commercial crops (Y.H. Chew and K.J. Halliday 2010).

On the basis of its literature presence and its applicability to future commercial crop study, experiments will be performed using the Arabidopsis Thaliana organism. The experimental trial employed the use of approximately 50 specimens arranged into four trial groups. A careful statistical analysis will have to be performed on the statistical significance of conclusions drawn from a such a small dataset.

3.6.3 Measures

Performance of the various scheduling algorithms was evaluated by comparing the quantity of applied irrigation to the total leaf area. As biomass is roughly proportional to the leaf area this measure effectively investigates the relationship between biomass production and water usage.

Many commercial crops, such as pastures and the majority of the brassicaceae family, have a direct relationship between biomass and yield. Therefore this measure represents a realistic crop performance scoring mechanism, albeit somewhat limited in scope. For more complex crops a differing score mechanism will have to be developed

3.6.4 Experimental Program

Chapter 4

Theory

4.1 Arabidopsis Thaliana Model

In order to develop the irrigation algorithm the author will require some way of testing it. It's never practical to try it out on plants right away. Ten years will be spent trying to figure out where the compiler bugged out. Something faster and more interactive is required and that is naturally a simulator, or a model.

The first part of any modelling activity is to identify the inputs of the system. By identifying the inputs you can begin the task of finding ways to simulate those inputs and their behaviours.

For the following right up we will consider the *Arabidopsis* organism. There's two main reasons to consider its use, it's very fast to grow and its very well documented. Both of those things are the key limits on a time constrained project like the one undertaken.

Arabidopsis is like any plant, it's effected by abiotic conditions such as weather and it is effected by biotic things such as growth phase, insect damage, pathway saturation etc.

These biotic processes by their own nature are very commonly nonlinear and they can prove very difficult and time consuming to model. Both of these things aren't really compatible with this thesis. The task is to develop an irrigation algorithm, not to build an even larger rube-goldberg machine.

For the rest of this project we will only consider the abiotic factors involved, the key biotic factors can be briefly summarised as, water availability, light availability and evaporative demand. Those three things are pretty large umbrella words for a bunch of individual effects.

But the key part of these effects is they rely on the environment around the organism and if we are talking about a plant out in the open that means the weather. In order to develop a model we will need some sort of virtual "weather environment" to test our algorithms.

As to where to source this information, well thats an interesting question. As we plan to grow it outside in Toowoomba it's pretty much a straight forward deduction that we will want to develop the algorithm in a weather environment like our own. Maybe last years measurements?

And that'd be perfect but there is one catch and that is data availability. The Bureau of Meteorology holds a pretty tight grip on data release and the data isn't all that great to begin with. Good solar observations are limited to a few key sites over the country but none of them are toowoomba.

Luckily a hop and a skip over the ocean and you have the United States, due too their government structure publically run organisations like NOAA and NASA have to release all their research into the public domain. That means great weather data. Unlike Australia, the United States runs a detailed set of solar observatories. The data from these observatories is available as the NSRDB (National Solar Radiation Database).

While the weather observations from NOAA can be obtained immediately, the publically available NSRDB data is delayed by approximately 2 years. So we can't choose last year but you know we can easily choose 2010.

Now its a case of picking a location, one thing I knew off the top of my head is that plenty of people contrast California with parts of Australia. It's dry, hot and yet a food bowl. A little researching later, honestly floating google earth over the californian state revealed an interesting patch of green, Fresno California.

Fresno is a large farming region and its hot and its ridiculously dry, a lot like most of australia, we're talking 300mm of rain a year. In order to grow crops in an environment like that you need irrigation, and that comes from the californian aqueduct system. It's a high tech. economy that relies on water constrained human irrigation. It's a perfect fit for the proposed technologies.

There's a final advantage, right in the middle of that green patch is the Fresno Air Terminal and it's high accuracy meterology observation station. The data from the Fresno location for the entire year of 2010 is trivial to obtain online.

There is issues however, the data has holes, from sensors going down, upgrades, etc. Holey data is terrible for our model, we're not trying to formulate some kind of plant/swiss cheese hybrid. So we need a way to fill those holes from a different source.

Thankfully the author has worked with the Global Forecast System (GFS) before. The GFS is a weather model run by NOAA that generates predicted forecast for the entired globe. The resolution is a little poor but it is one of the very few models that allow public access to its data.

As such, the author undertook the task of downloading the entire GFS dstaset for the year of 2010 (All 100GB Worth). Due to dodgy peering agreements, the american servers were horribly slow, averaging maybe 20kB/s. Luckily the author recognized this and configured a cheap linode virtual server located in america to scrape the data and then relay it back to australia.

After about 7 days the entire dataset was collected and collated, it weas then processed with the opensource tool wgrib and the relevant records for sea level conditions were extracted. These included air temperature, relative humidity and wind speed/direction. These entries were then lineary interpolated for the latitude and longitude of the fresno air terminal and were compiled into a set of large text files each containing roughly 9000 entries. Each entry being measured in hours since the start of the year.

Due to conflicts with timezones the author decided to start the reference year in the simulator on january 2nd 2010. Once the text files were collected and appropriately formatted, the author then processed the NSRDB to get a yearly solar flux.

Some experiments were then used to compare how good a fit the processed GFS data was for the Fresno site. There was some significant variances between the fits. The author decided then to fuse the holey actual observations with the smooth but sometimes not 100 accurate GFS dataset and aftet some filtering the weather dataset was born.

There was two remaining parameters needed for modelling the plant and these were plant leaf temperature and soil temperature. The soil temperature was generated from the weather observations using a model developed by B Horton at CSIRO. The leaf temperature was then calculated using the stomatal conductivity values for arabidopsis, sourced from the three papers *"Arabidopsis homeodomain-leucine zipper IV proteins promote stomatal development and ectopically induce stomata beyond the epidermis"*, *"Phytochrome B Enhances Photosynthesis at the Expense of Water-Use Efficiency in Arabidopsis"*, *"The ERECTA gene regulates plant transpiration ef?ciency in Arabidopsis"*.

Using the conductivity values identified and the leaf geometry for common arabidopsis strains the author was able then to use some math from the textbook "environmental biophysics" to develop a leaf temperature model.

With all these inputs it was just a case of a lot of reading, which can be seen by perusing the comments of the attached crop model code (in appendix).

4.2 Field Experiment Block Diagrams

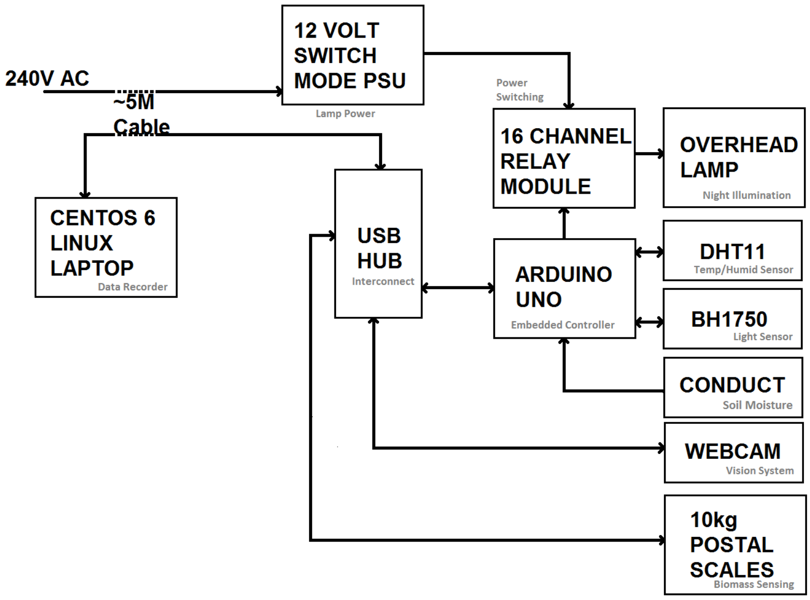


Figure 4.1: Block Diagram of The Model Validation Experiment

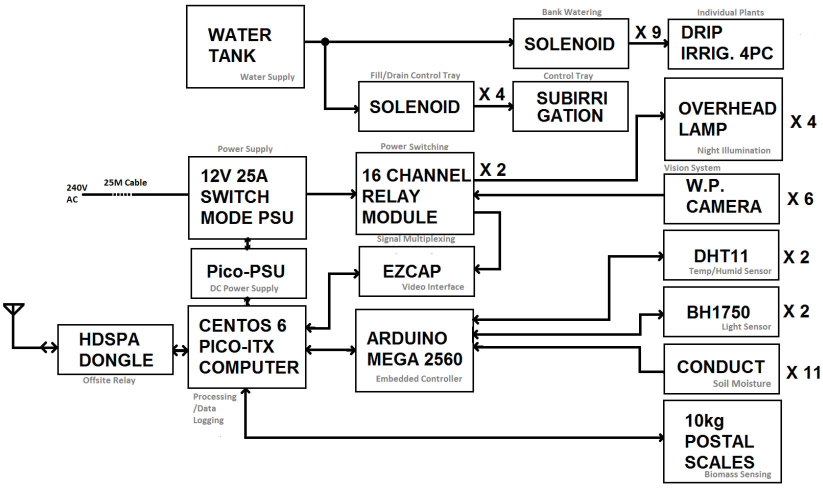


Figure 4.2: Block Diagram of The Algorithm Field Test Experiment

Chapter 5

Results

5.1 Arabidopsis Thaliana Model

5.1.1 Summer Weather

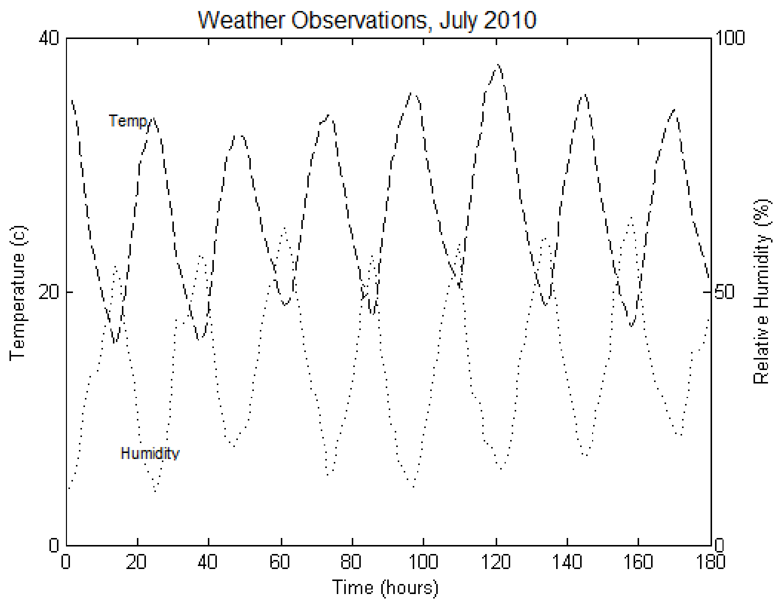


Figure 5.1: Temperature/Humidity Observations at Fresno Air Terminal, July 1st-8th (2010)

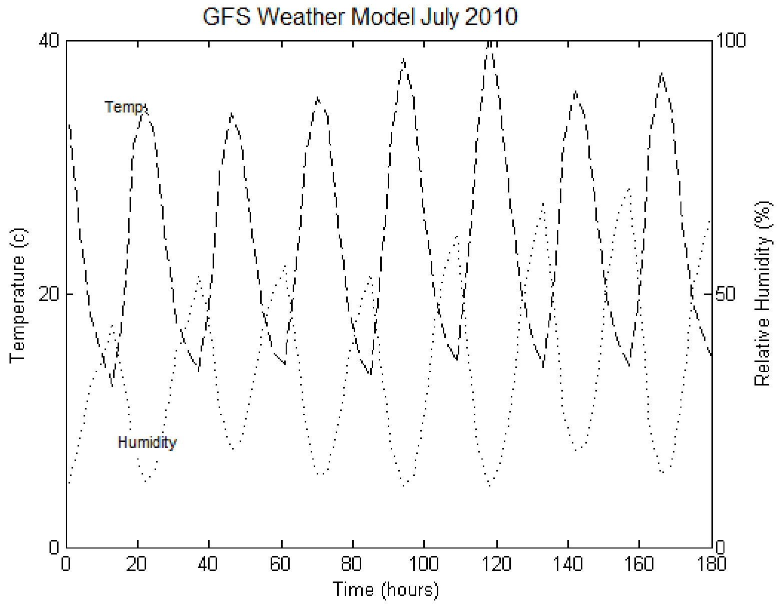


Figure 5.2: Interpolated Sea Level GFS 3h Predictions, Fresno California, July 1st-8th (2010)

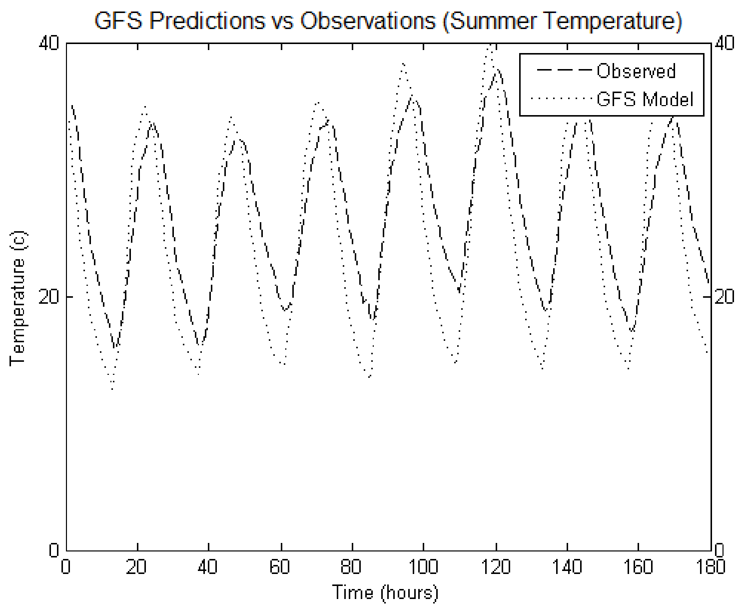


Figure 5.3: Comparison of GFS Predictions and NOAA Observations (Air Temperature), Fresno California, July 1st-8th (2010)

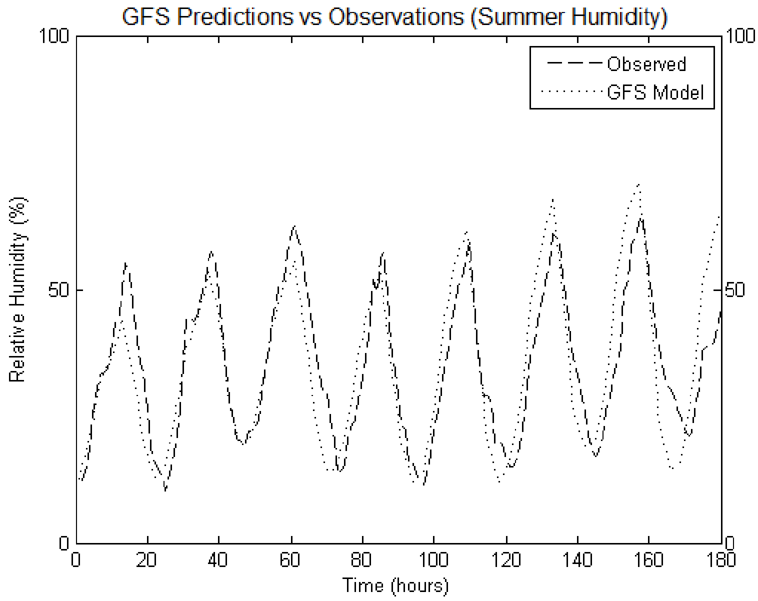


Figure 5.4: Comparison of GFS Predictions and NOAA Observations (Relative Humidity), Fresno California, July 1st-8th (2010)

Table 5.1: GFS Summer Season Performance, Fresno California, 2010

Fit	$\mathbf{R}^2$	**P**
Summer Temperature	77.9%	<0.05
Summer Humidity	74.2%	<0.05

5.1.2 Winter Weather

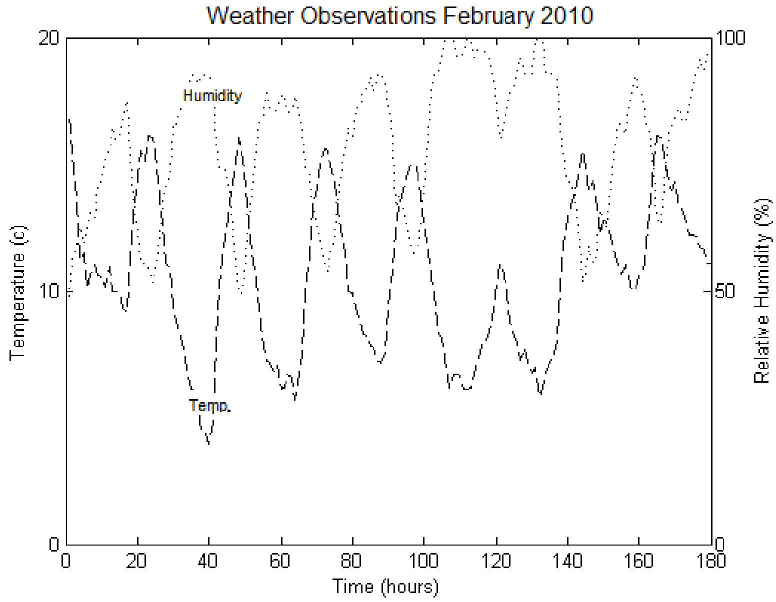


Figure 5.5: Temperature/Humidity Observations at Fresno Air Terminal, February 1st-8th (2010)

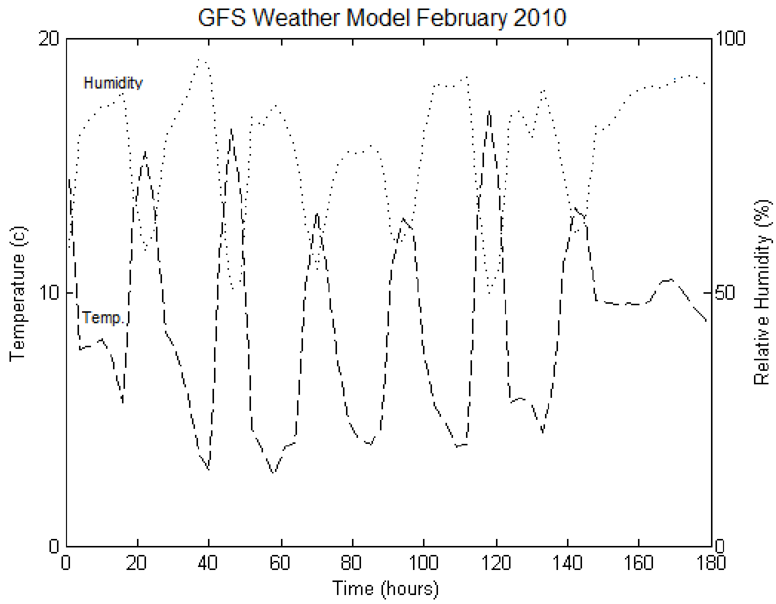


Figure 5.6: Interpolated Sea Level GFS 3h Predictions, Fresno California, February 1st-8th (2010)

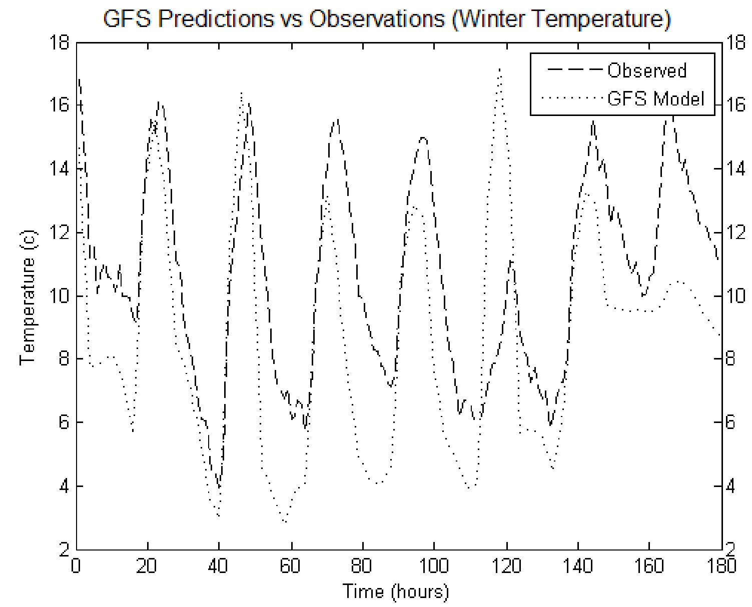


Figure 5.7: Comparison of GFS Predictions and NOAA Observations (Air Temperature), Fresno California, February 1st-8th (2010)

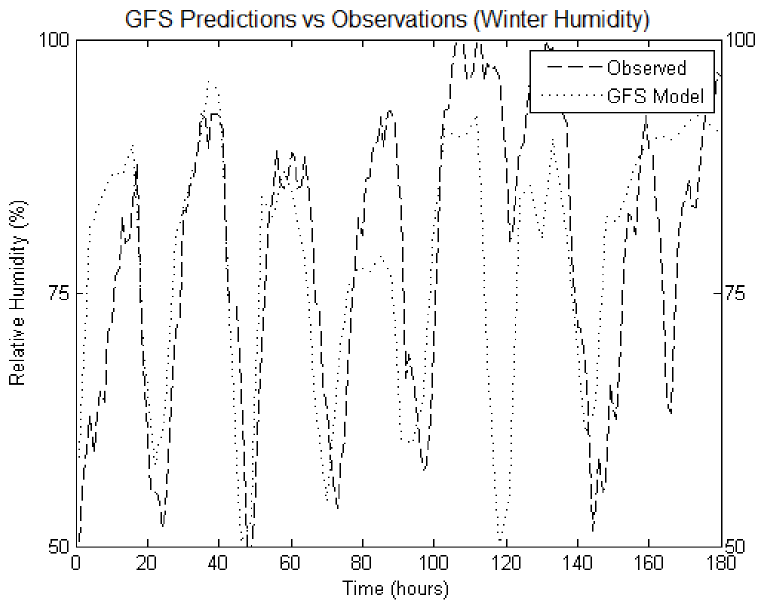


Figure 5.8: Comparison of GFS Predictions and NOAA Observations (Relative Humidity), Fresno California, February 1st-8th (2010)

Table 5.2: GFS Winter Season Performance, Fresno California, 2010

Fit	$\mathbf{R}^2$	**P**
Winter Temperature	22.8%	<0.05
Winter Humidity	2.3%	<0.05

5.1.3 Fused Data

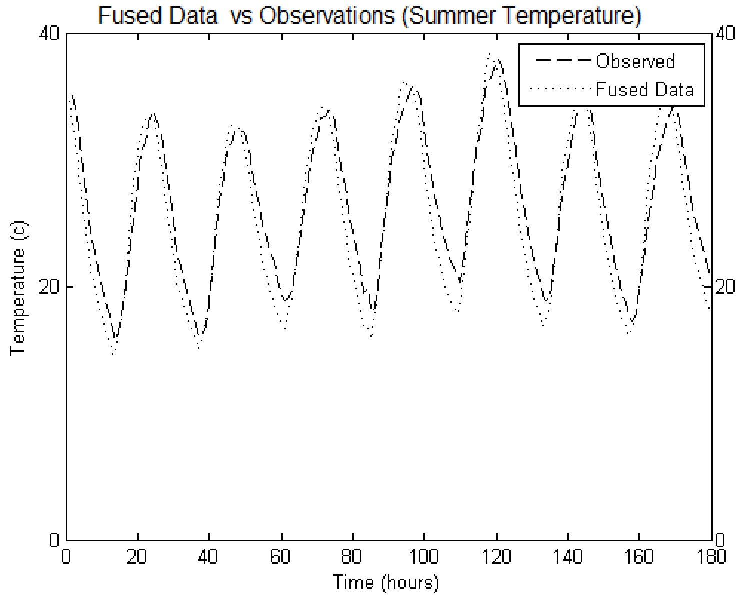


Figure 5.9: Comparison of Fused Data (An Interpolated Combination of GFS and NOAA Datasets) Against NOAA Observations (Air Temperature), Fresno California, July 1st-8th (2010)

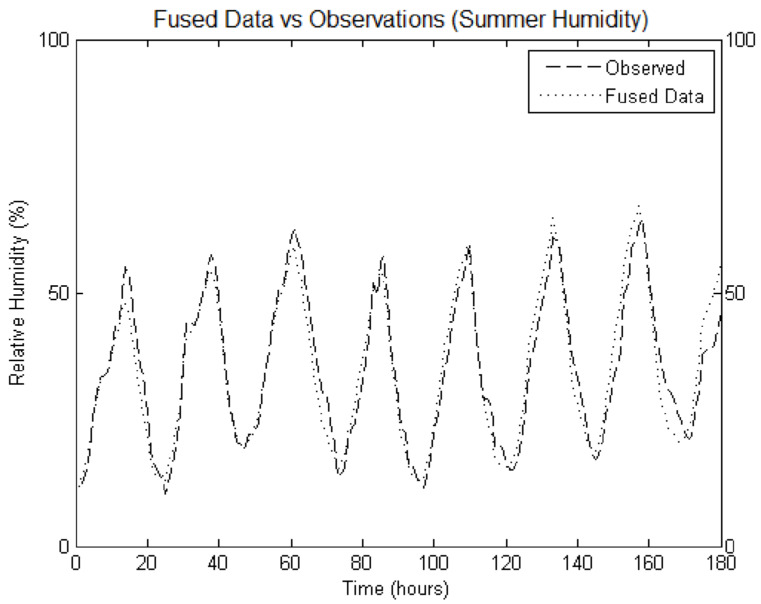


Figure 5.10: Comparison of Fused Data Against NOAA Observations (Relative Humidity), Fresno California, July 1st-8th (2010)

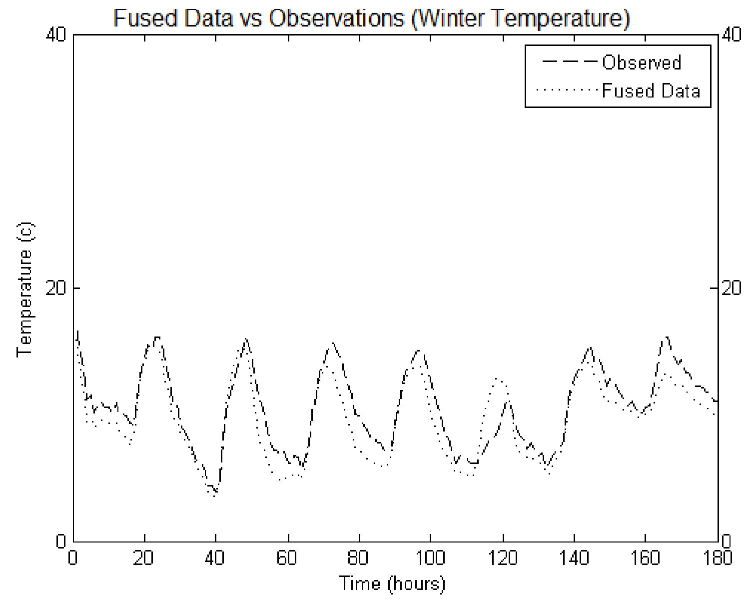


Figure 5.11: Comparison of Fused Data Against NOAA Observations (Air Temperature), Fresno California, February 1st-8th (2010)

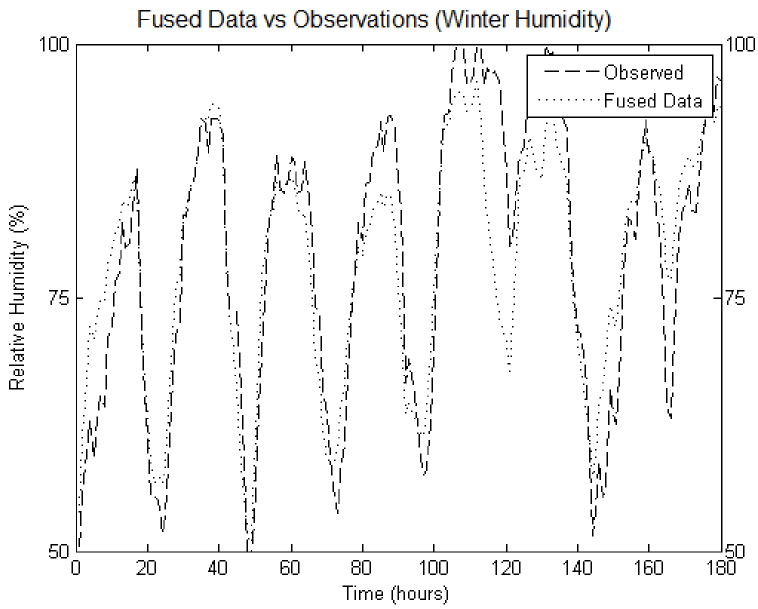


Figure 5.12: Comparison of Fused Data Against NOAA Observations (Relative Humidity), Fresno California, February 1st-8th (2010)

Table 5.3: Fused Dataset Performance, Fresno California (2010)

Fit	$\mathbf{R^2}$	**P**
Summer Temperature	92.3%	<0.05
Winter Temperature	68.2%	<0.05
Summer Humidity	92.7%	<0.05
Winter Humidity	45.2%	<0.05

5.1.4 Wind Measurements

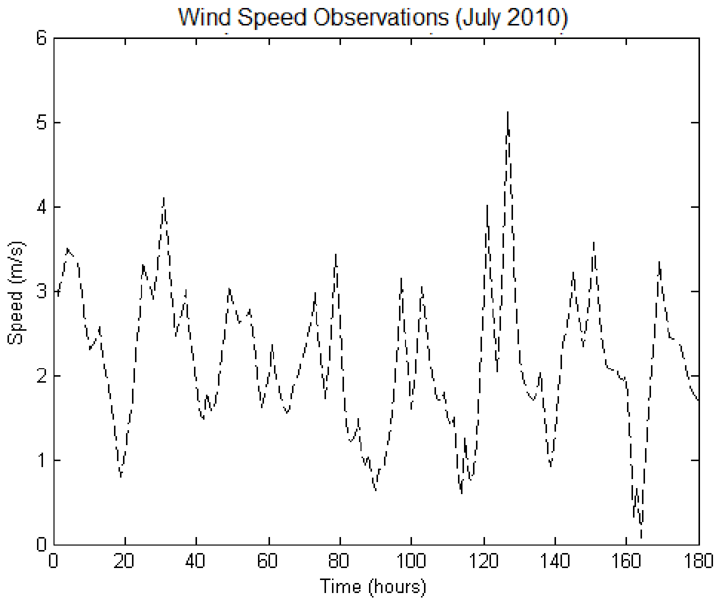


Figure 5.13: Summer Windspeed, Fresno California, July 1st-8th (2010)

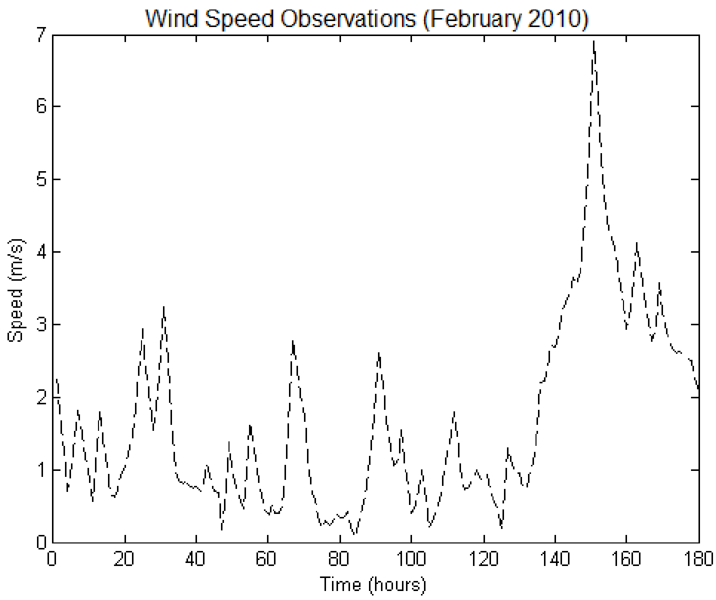


Figure 5.14: Winter Windspeed, Fresno California, February 1st-8th (2010)

5.1.5 Solar Flux Measurements

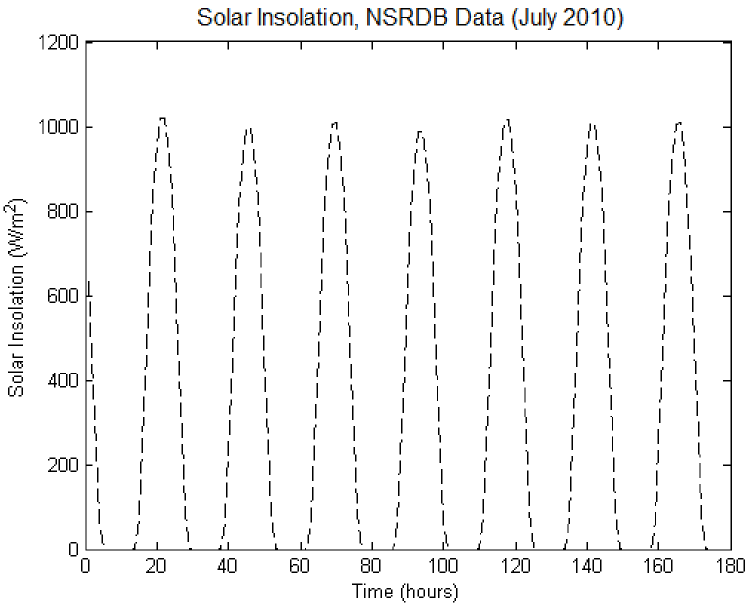


Figure 5.15: Summer Solar Insolation, METSTAT Solar Radiation Model, Fresno California, July 1st-8th (2010)

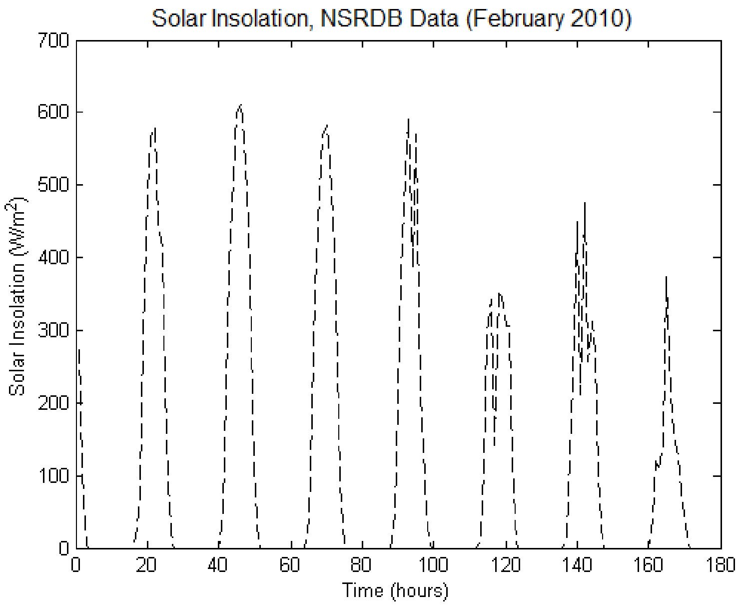


Figure 5.16: Winter Solar Insolation, METSTAT Solar Radiation Model, Fresno California, February 1st-8th (2010)

5.1.6 Leaf Temperature Model

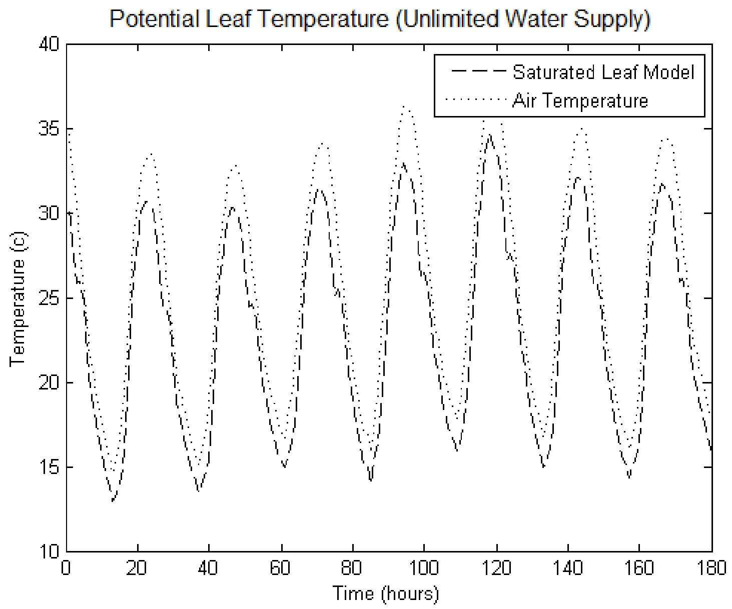


Figure 5.17: Potential *Arabidopsis* Leaf Temperature (Custom Physics Model), July 1st-8th (2010)

5.1.7 CSIRO Shallow Soil Model

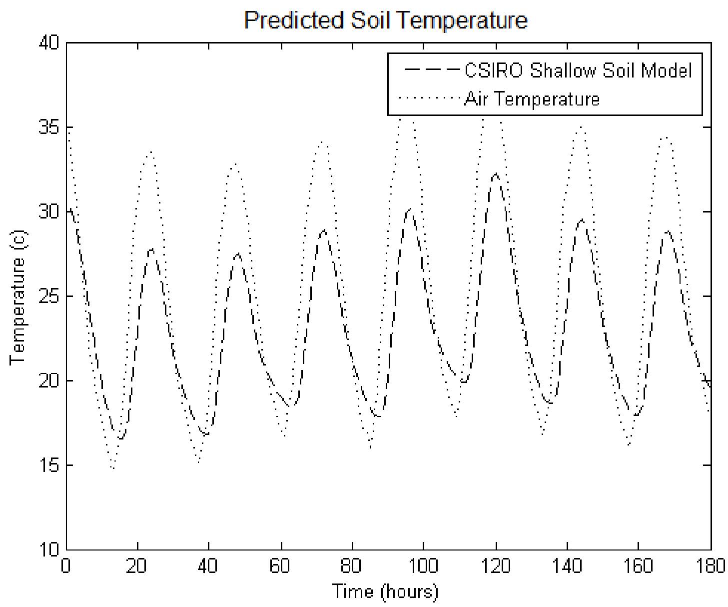


Figure 5.18: Predicted Shallow (5cm) Soil Temperature (Horton 2012)

5.1.8 Photosynthetic Rate vs Solar Insolation

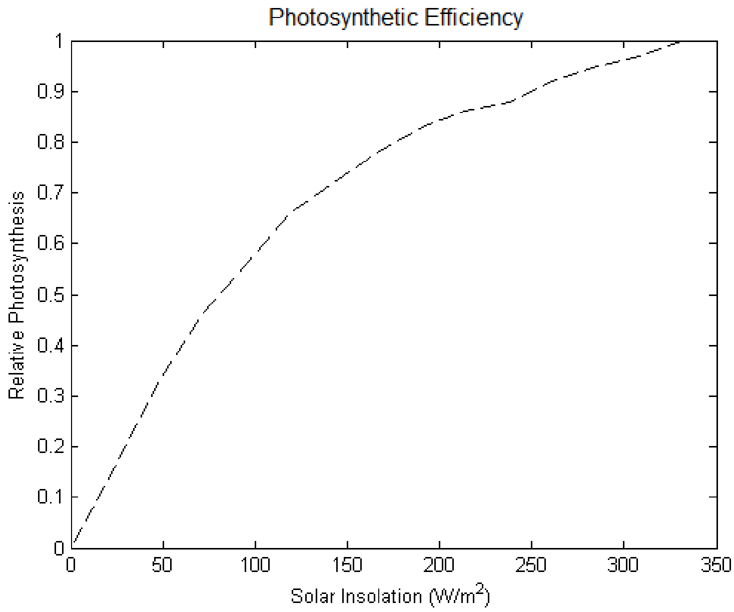


Figure 5.19: Relative *Arabidopsis* Photosynthetic Efficiency vs Solar Insolation

5.1.9 Factors Effecting Transpiration

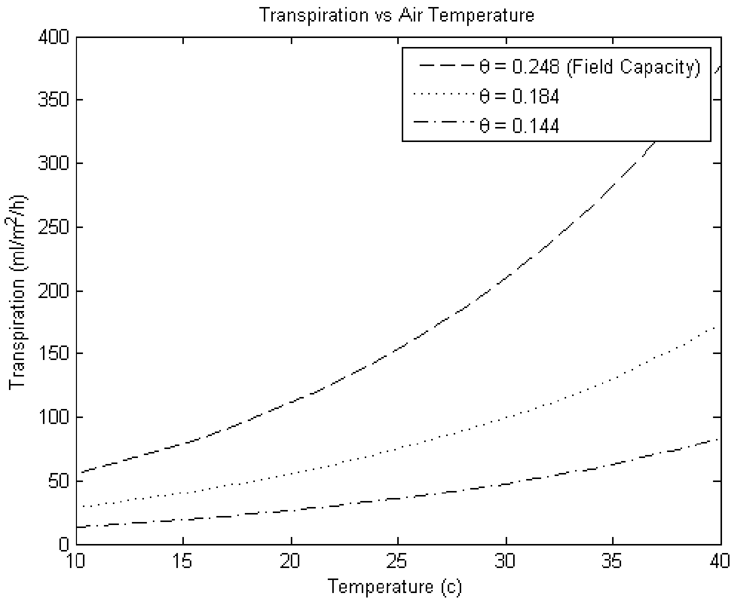


Figure 5.20: *Arabidopsis* Transpiration vs Solar Insolation

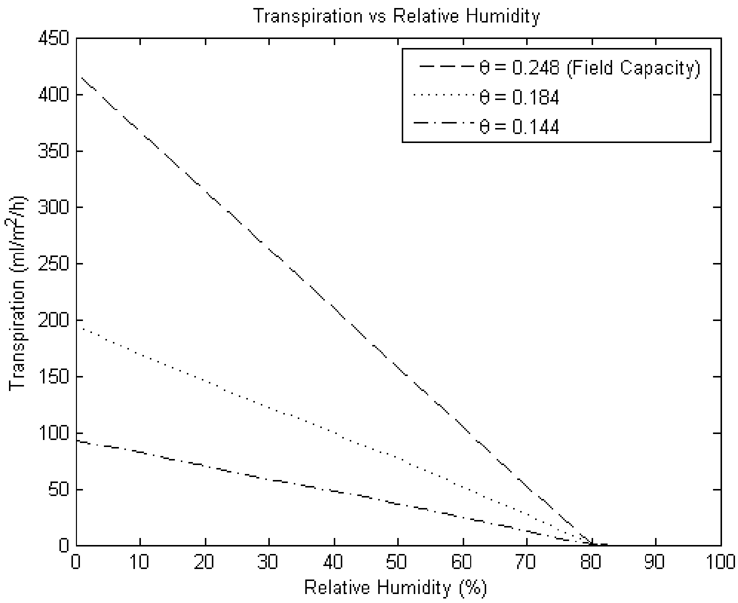


Figure 5.21: *Arabidopsis* Transpiration vs Relative Humidity

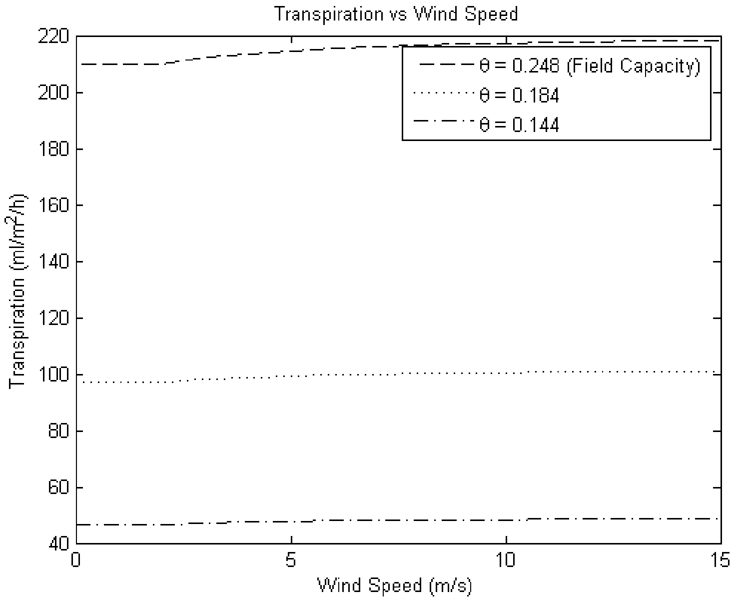


Figure 5.22: *Arabidopsis* Transpiration vs Wind Speed

5.1.10 Complete ***Arabidopsis*** Crop Model

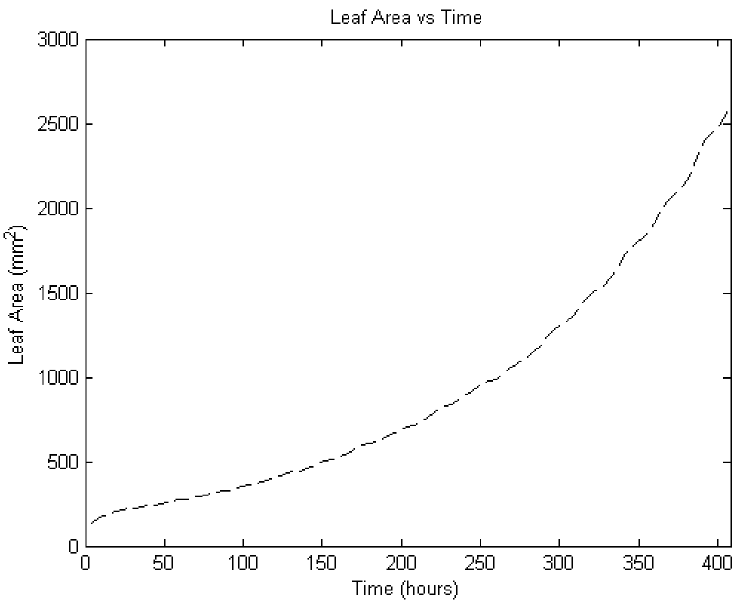


Figure 5.23: Predicted Leaf Area / Biomass

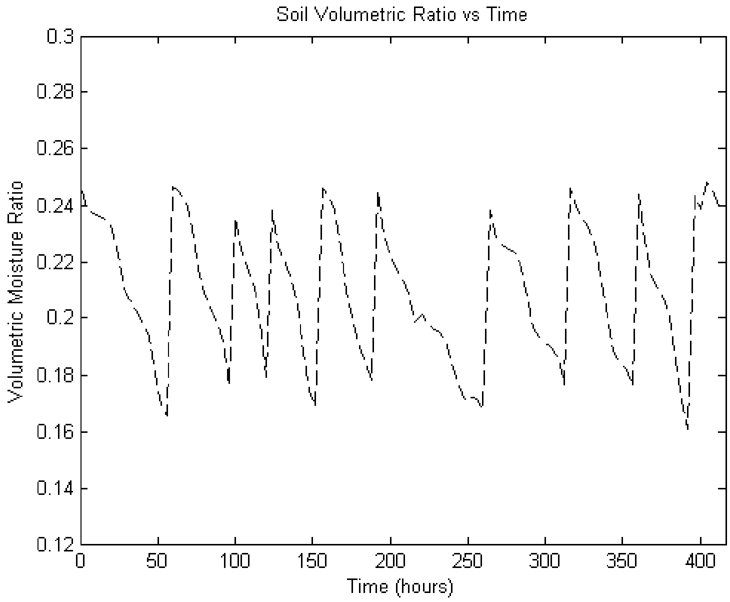


Figure 5.24: Soil Volumetric Water Ratio

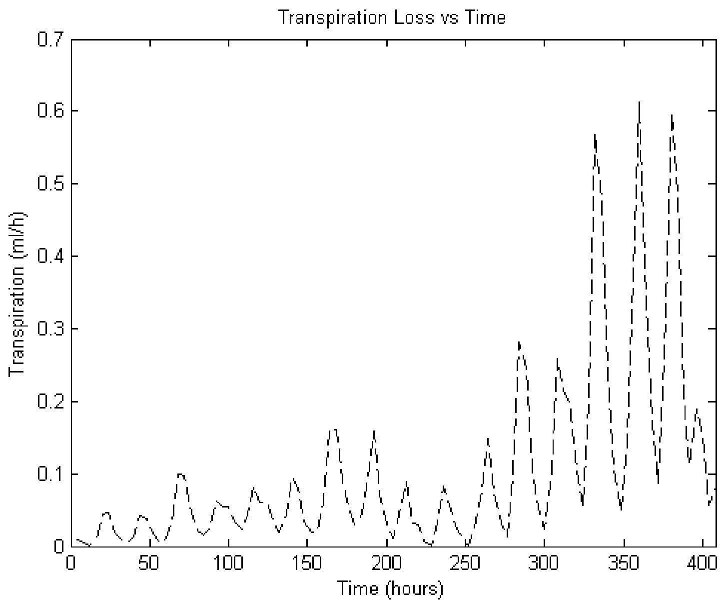


Figure 5.25: Modelled Transpiration Loss

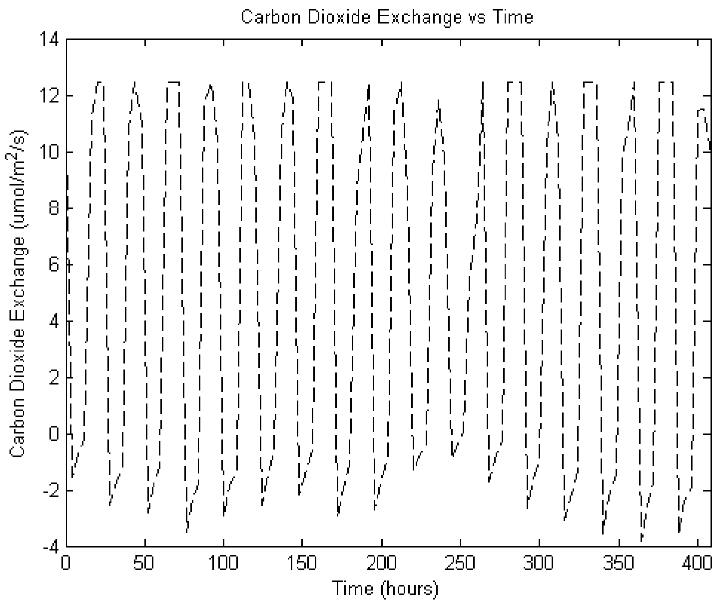


Figure 5.26: Modelled Carbon Dioxide Exchange

5.2 Machine Learning Algorithm

5.2.1 Dataset Clustering (k-means)

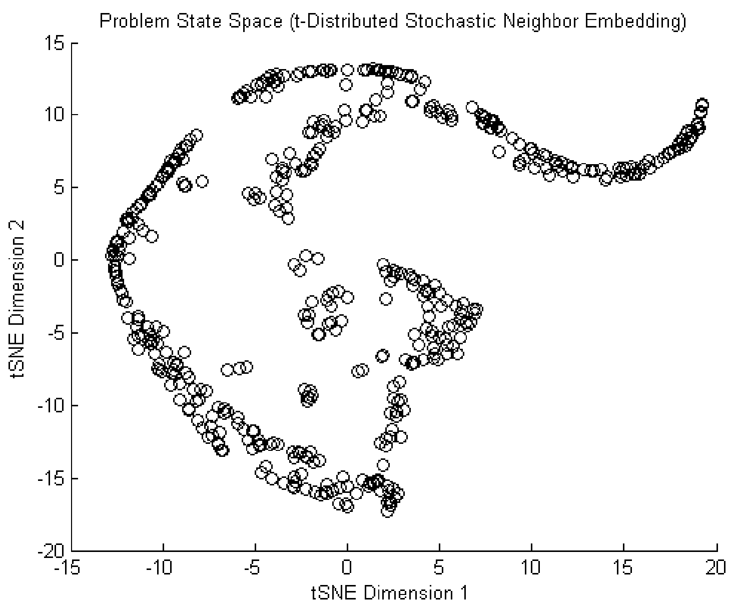


Figure 5.27: Graphical Representation Of Problem State-Space

Table of centroid values ....

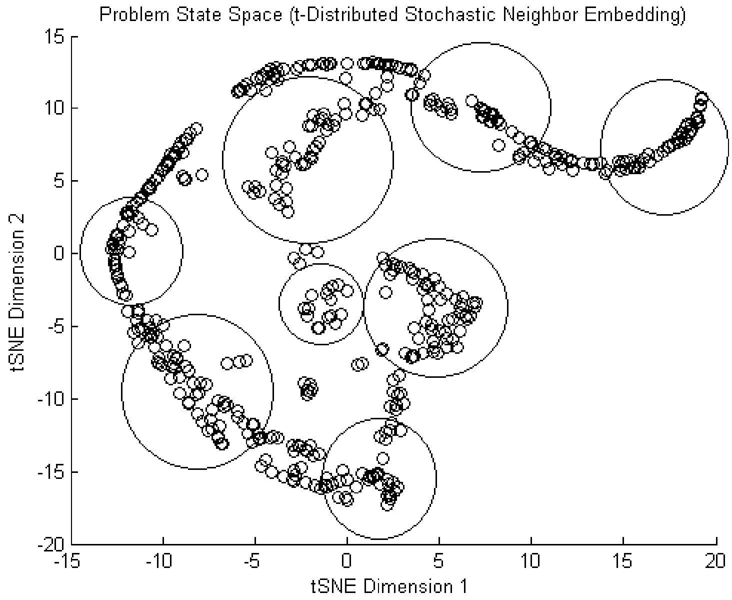


Figure 5.28: Identifying Cluster Boundaries In The Visual Data

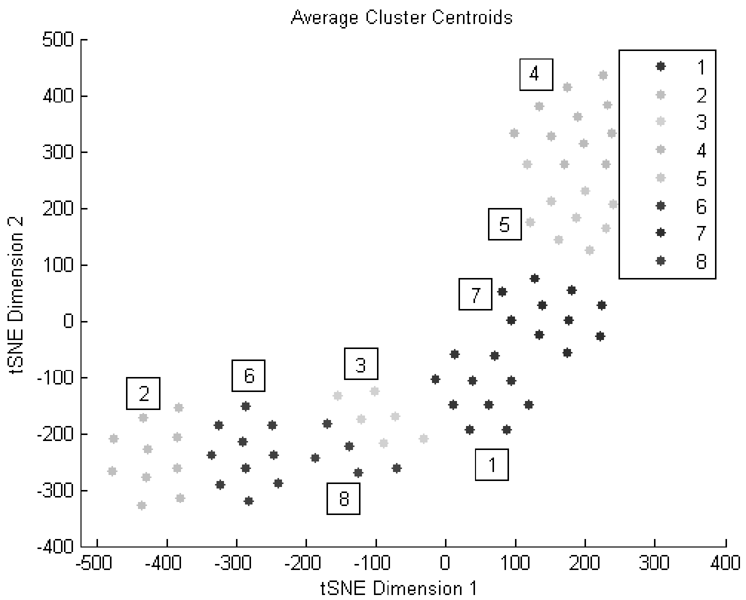


Figure 5.29: Running K-Means Repeatedly In Order To Average Out Local Optima

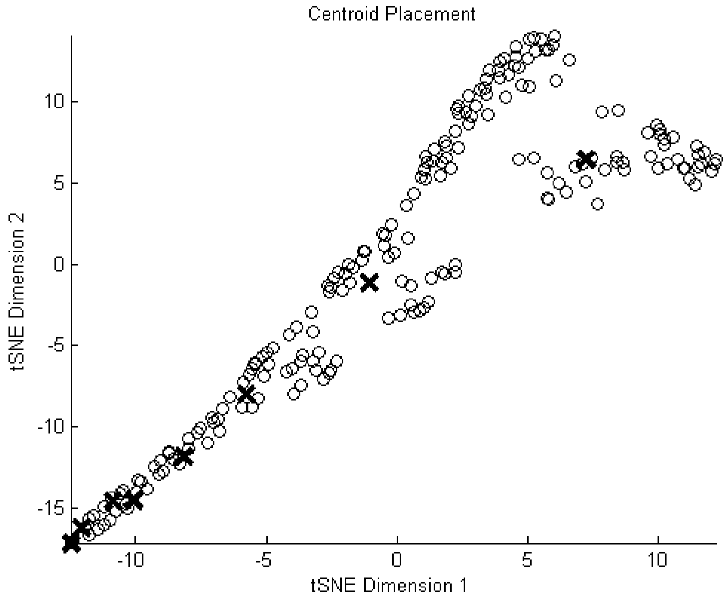
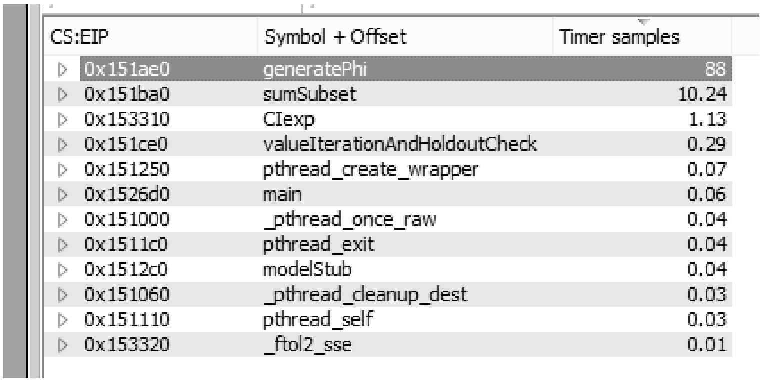


Figure 5.30: Final Positions of RBF Centers In The State Space

5.2.2 High Performance Computing

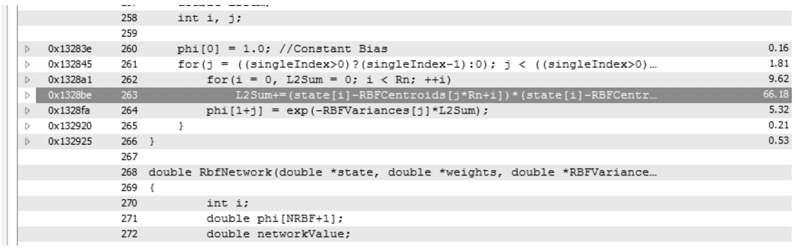
```
top - 02:30:49 up 14 min,  1 user,  load average: 0.23, 0.30, 0.24
Tasks: 549 total,   2 running, 547 sleeping,   0 stopped,   0 zombie
Cpu(s):  5.8%us,  1.7%sy,  0.0%ni, 92.5%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  62519768k total,  1117744k used, 61402024k free,    20520k buffers
Swap:        0k total,        0k used,        0k free,   569588k cached

   PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
130411 root      20   0  110m 6560  768 R 100.0  0.0   0:54.77 code
  8230 root      20   0 15400 1692 1008 R  0.7  0.0   0:00.35 top
    37 root      20   0     0    0    0 S  0.3  0.0   0:00.03 ksoftirqd/8
   105 root      20   0     0    0    0 S  0.3  0.0   0:00.02 ksoftirqd/25
     1 root      20   0 19356 1544 1224 S  0.0  0.0   0:04.47 init
     2 root      20   0     0    0    0 S  0.0  0.0   0:00.03 kthreadd
     3 root      RT   0     0    0    0 S  0.0  0.0   0:00.06 migration/0
     4 root      20   0     0    0    0 S  0.0  0.0   0:00.03 ksoftirqd/0
     5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
     6 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
     7 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/1
     8 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/1
     9 root      20   0     0    0    0 S  0.0  0.0   0:00.01 ksoftirqd/1
    10 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/1
    11 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/2
    12 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/2
    13 root      20   0     0    0    0 S  0.0  0.0   0:00.01 ksoftirqd/2
```

Figure 5.31: Serial Batch Gradient Descent, Poor Processor Utilization



CS:EIP	Symbol + Offset	Timer samples
0x151ae0	generatePhi	88
0x151ba0	sumSubset	10.24
0x153310	CIexp	1.13
0x151ce0	valueIterationAndHoldoutCheck	0.29
0x151250	pthread_create_wrapper	0.07
0x1526d0	main	0.06
0x151000	_pthread_once_raw	0.04
0x1511c0	pthread_exit	0.04
0x1512c0	modelStub	0.04
0x151060	_pthread_cleanup_dest	0.03
0x151110	pthread_self	0.03
0x153320	_ftol2_sse	0.01

Figure 5.32: Batch Gradient Descent Code Profiling



	258	`int i, j;`	
	259		
0x13283e	260	`phi[0] = 1.0; //Constant Bias`	0.16
0x132845	261	`for(j = ((singleIndex>0)?(singleIndex-1):0); j < ((singleIndex>0)...`	1.81
0x1328a1	262	`for(i = 0, L2Sum = 0; i < Rn; ++i)`	9.62
0x1328be	263	`L2Sum+=(state[i]-RBFCentroids[j*Rn+i])*(state[i]-RBFCentr...`	66.18
0x1328fa	264	`phi[1+j] = exp(-RBFVariances[j]*L2Sum);`	5.32
0x132920	265	`}`	0.21
0x132925	266	`}`	0.53
	267		
	268	`double RbfNetwork(double *state, double *weights, double *RBFVariance...`	
	269	`{`	
	270	`int i;`	
	271	`double phi[NRBF+1];`	
	272	`double networkValue;`	

Figure 5.33: The Performance Bottleneck is Identified

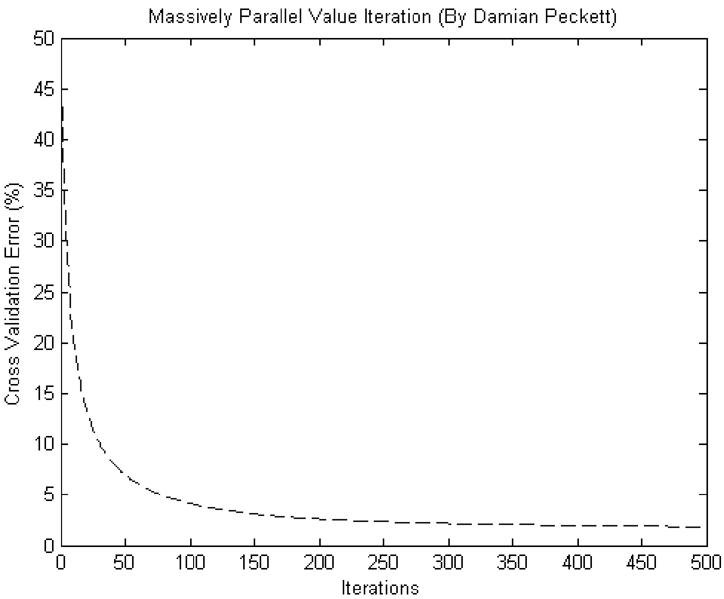


Figure 5.34: Performance Of Custom Parallel Implementation

5.2.3 Parallel Value Iteration Results

The following values were computated using the authors massively parallel MDP solver. The source code of which can be found in the appendix under the name <crazyfastMDPsolve.c>.

Table 5.4: Array of Identified RBF Centroids

TMP	RH	Flux	Water	Area	Wind	T24	RH24	W24	T48	RH48	W48	MJ	Lt
296	40	369	0.158	6265	1.8	296	41	1.8	296	42	1.6	27	0.621
298	38	374	0.168	21144	2.1	299	36	2.5	298	35	2.5	31	1.42
296	43	290	0.164	9870	1.8	296	44	2	295	45	2.1	26	0.772
293	51	340	0.175	577	1.8	291	55	1.7	291	55	1.7	21	0.0948
292	53	284	0.171	1826	1.7	292	54	1.7	292	54	1.6	21	0.225
297	41	299	0.151	15506	2.4	297	39	2.2	298	39	2.1	29	1.18
293	50	293	0.162	3713	1.8	292	50	1.8	293	50	1.8	23	0.406
295	44	291	0.159	10501	2	295	45	2	295	45	2.2	26	0.808

Maximum Yield Per Unit Water, Reward Function:

$$Water = \begin{cases} 0.01 & : Irrigation_L < 0.01 \\ \\ Irrigation_L & : Irrigation_L >= 0.01 \end{cases}$$

$R(x) = \frac{LeafArea_{mm^2}}{Water}$

Trained MDP Parsmeters: $\gamma = 1.1642e - 16$, Error $= 1.97\%$

$\theta = [315909, 315909, 315909, 315909, 315909, 315909, 315909, 315909, 315909]$

Constant Soil Water Volume Regulation, Reward Function: $\theta = 0.2$, cubic.

$R(x) = (1.176 \cdot (0.85 - abs(WaterAvailability - 0.15)))^3$

Trained MDP Parsmeters: $\gamma = 6.4e - 11$, Error $= 0.164\%$

$\theta = [4.79966, 4.79161, 4.6989, 4.78138, 4.79161, 4.7933, 4.7496, 4.79406, 4.77877]$

5.3 Model Validation (Published Observations)

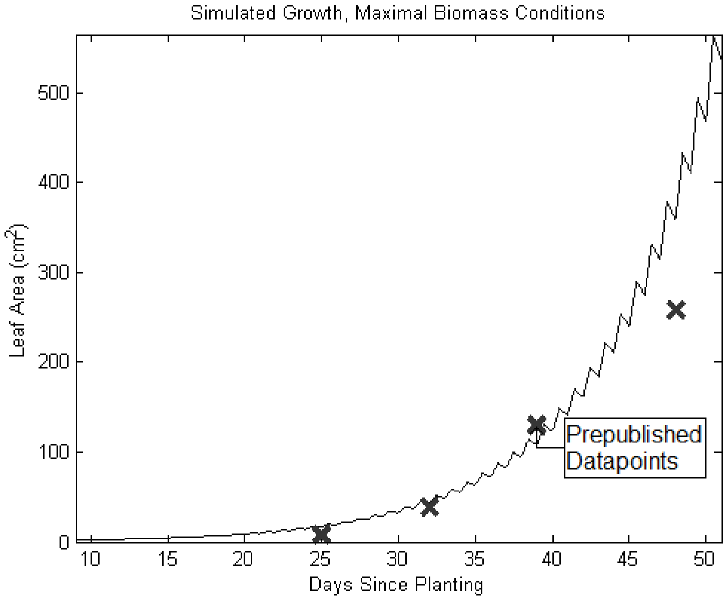


Figure 5.35: Maximal Biomass Accumulation Test

Table 5.5: Model Performance

Day	Modelled Leaf Area (cm^2)	Expected Area (cm^2)	Error %
25	16.8	8.31	+102%
32	42.8	38.9	+10%
39	108.6	130	-17%
48	359.9	258	+39%

5.4 Model Validation (Field Testing)

5.4.1 Soil Sensor Calibration

Figure 5.36: Undertaking Soil Moisture Probe Calibration

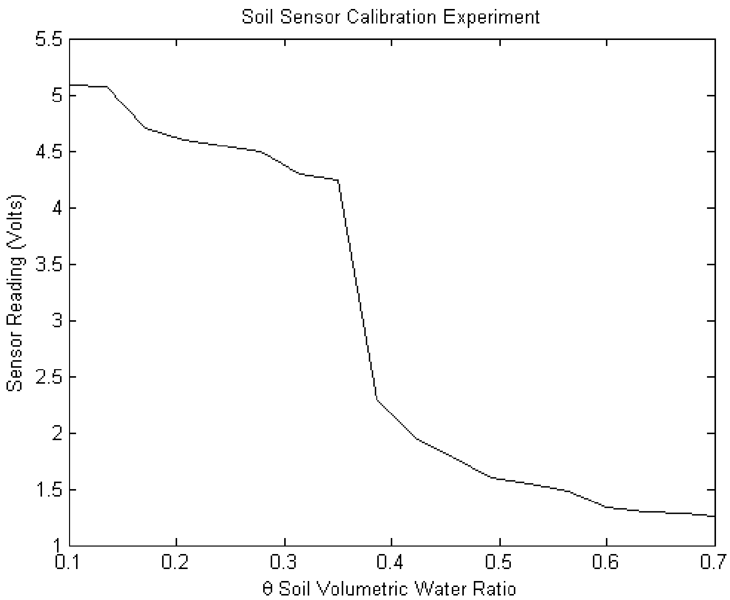


Figure 5.37: Voltage vs Soil Volumetric Water Ratio For Resistivity Sensing Device

5.4.2 Experimental Apparatus

Figure 5.38: Assembled Test Apparatus, Semi-Shaded Test Area

5.4.3 Biological Specimens

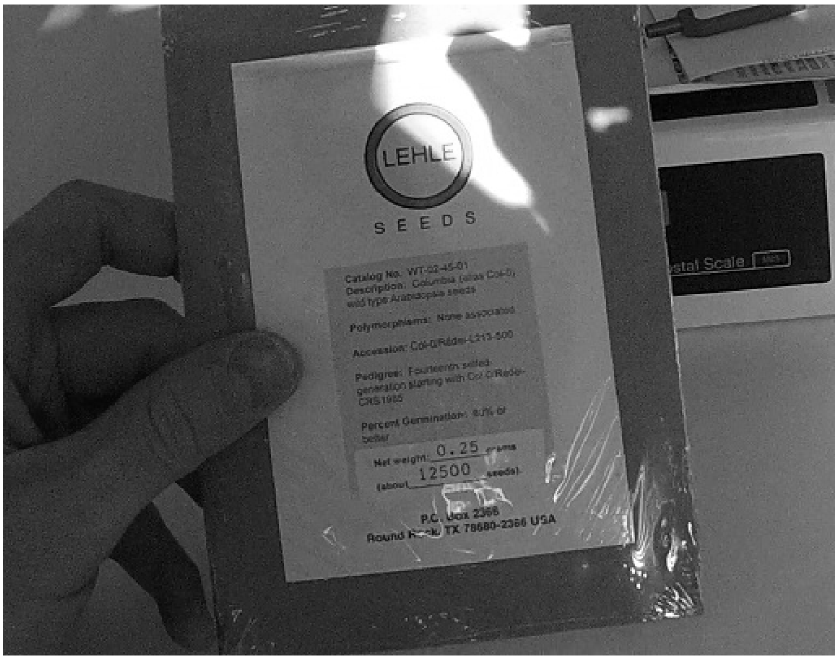


Figure 5.39: *A*rabidopsis Thaliana COL=0 Specimens From Lehle Seeds

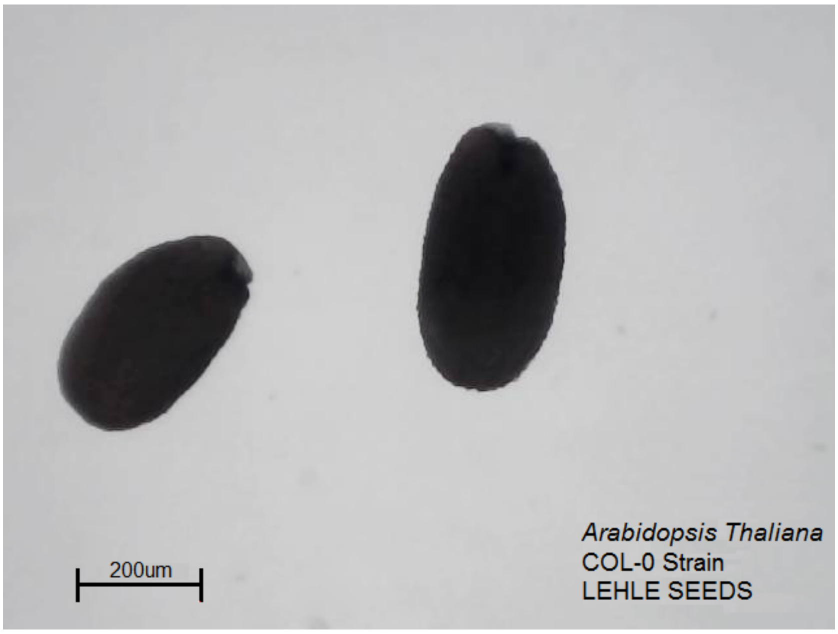


Figure 5.40: Verifying Seed Integrity Before Viability Testing

Figure 5.41: Preparing *Arabidopsis* Seeds For Planting With a Pipette

Figure 5.42: The Seeds Placed on The Surface of a Very Fine Propagation Mix

Figure 5.43: Freshly Germinated Seedlings Transfered To Larger Pots

Figure 5.44: Tray of Three Week Old Arabidopsis Seedlings Ready For Experimentation

5.4.4 Experiment Results

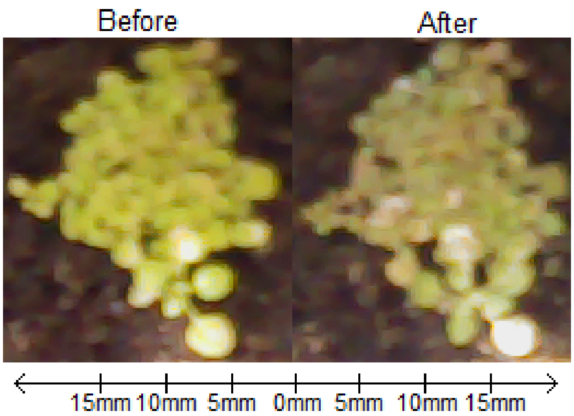


Figure 5.45: *Arabidopsis* Specimen Before and After 7-Day Experimental Test Period, If In Grayscale Refer Below

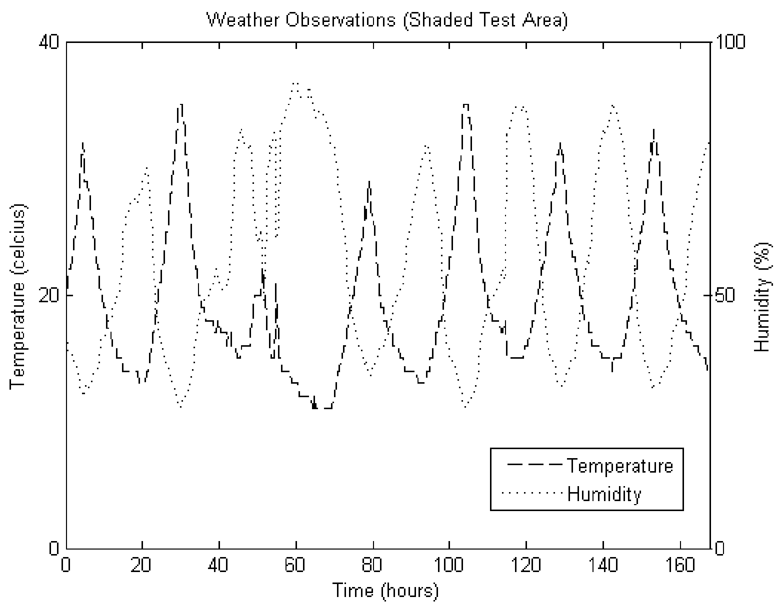


Figure 5.46: Humidity and Temperature Observations During The Experimental Test Period

The astute reader may recognize there was reference to a 10kg postal scale in the block diagram for the validation experiment. It turns out despite the high resolution of the scale it was unsuitable for the experiment as it had drift compensation. This lead to a stable null output which was entirely unusable.

For those readers limited to a black and white palette, Fig 5.45 shows little size change however in the after image, the leaves have lost their green lustre and are beginning to turn gray. The author also noted a large increase in the number leaf hairs.

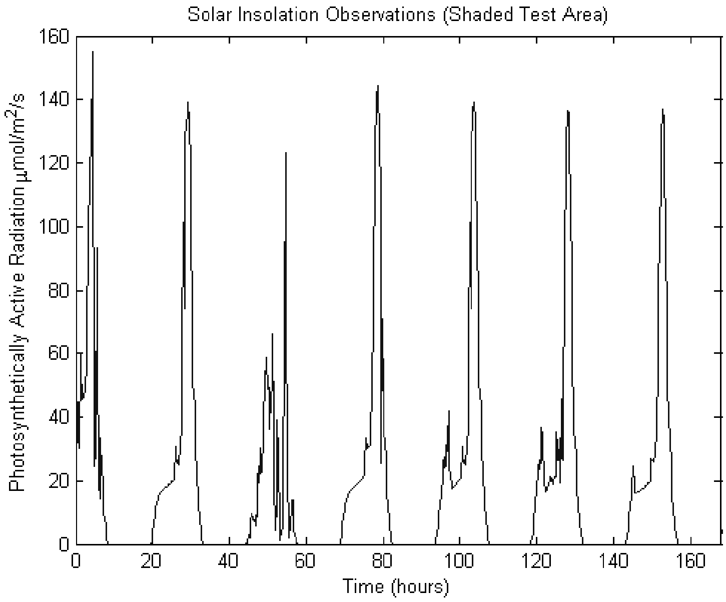


Figure 5.47: Solar Insolation Observations During The Experimental Test Period

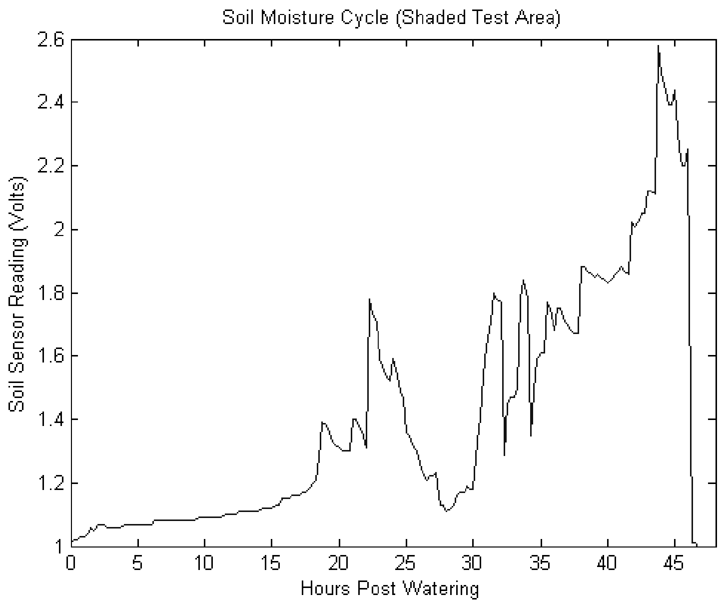


Figure 5.48: Soil Moisture Variance, 48 Hour, Normal Irrigation Schedule

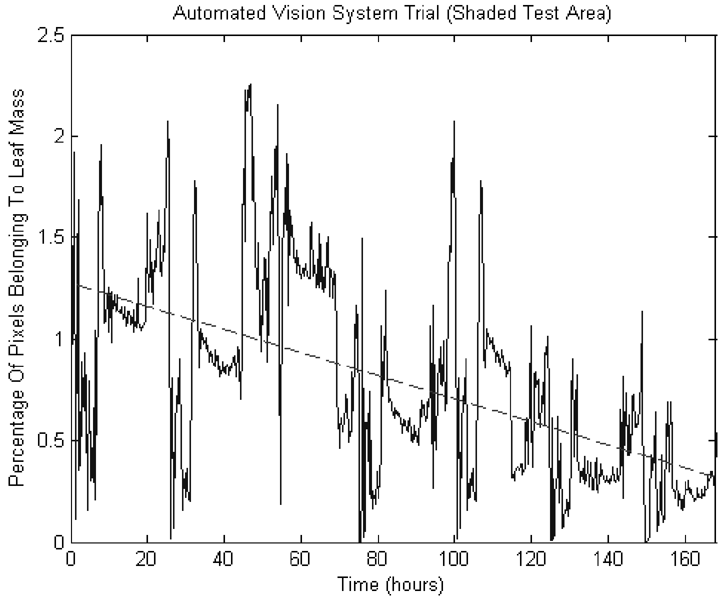


Figure 5.49: Leaf Area Observations (Healthy Leaf Mass), During The Experimental Test Period

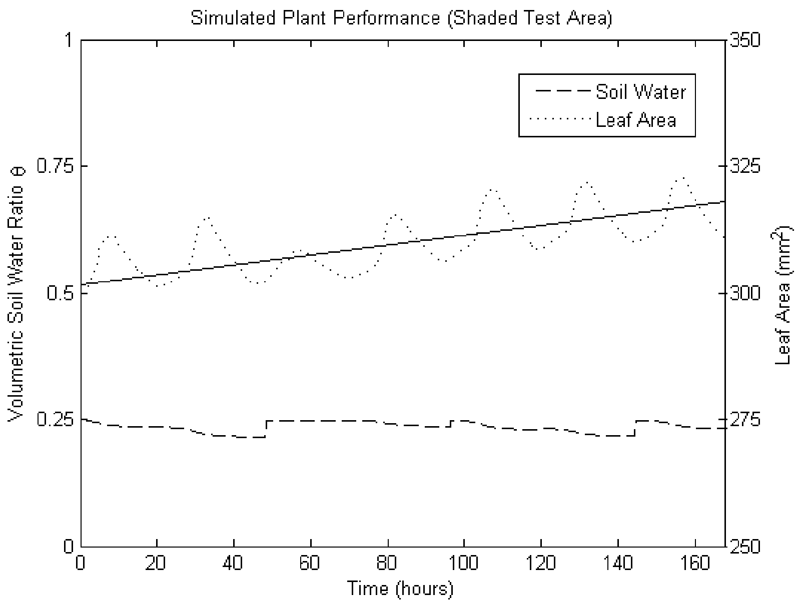


Figure 5.50: Simulation Run Using Observed Experimental Weather Conditions

5.5 Algorithm Performance (Crop Model)

Location: Fresno Air Terminal, Fresno, California, USA

Simulation Timespan: April 1st-29th (2010) (Season: Spring)

Average Monthly Temperature Range: 10->27 celcius (World Weather Online 2013)

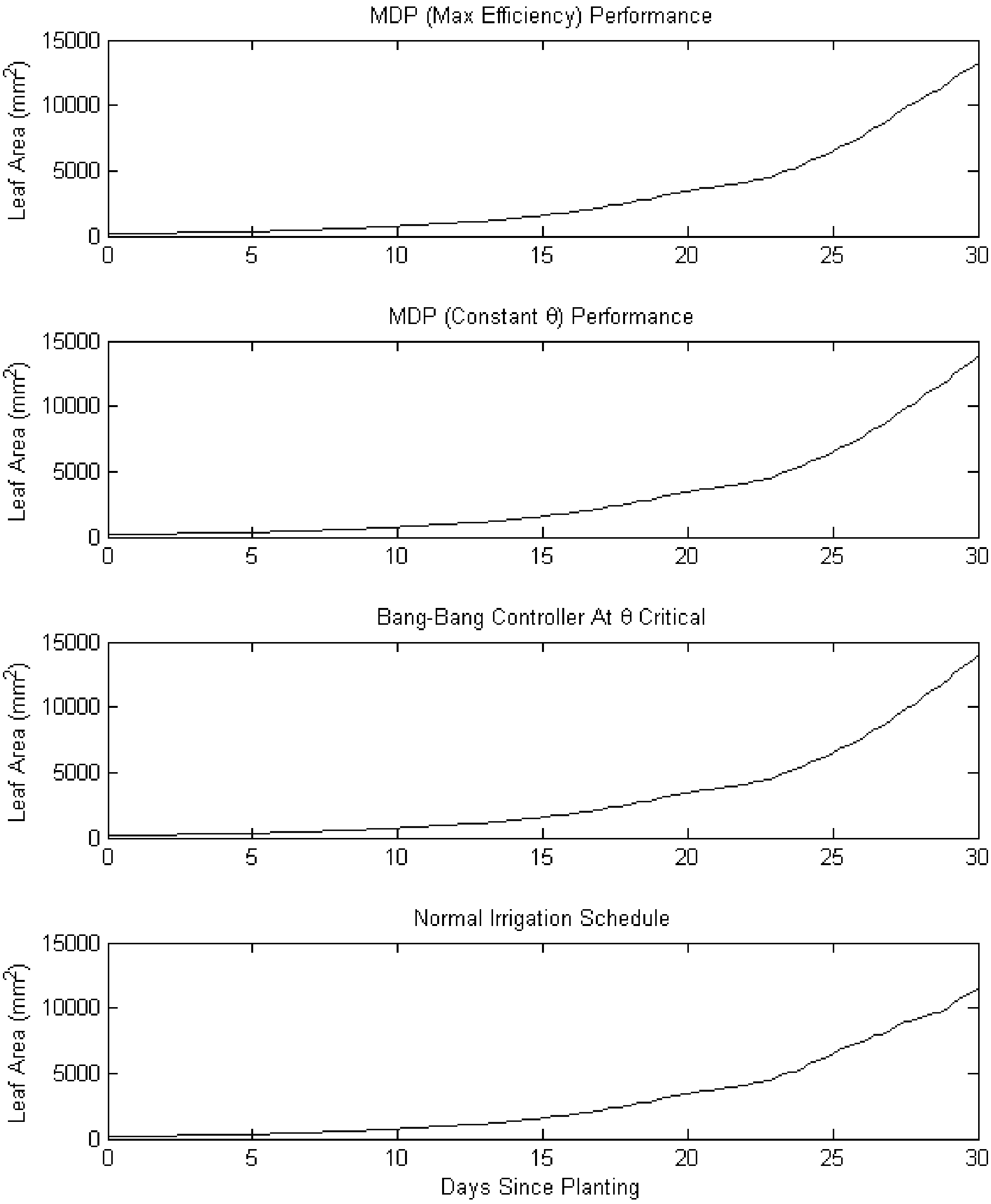


Figure 5.51: Total Accumulated Biomass, Comparison of Varying Control Strategies

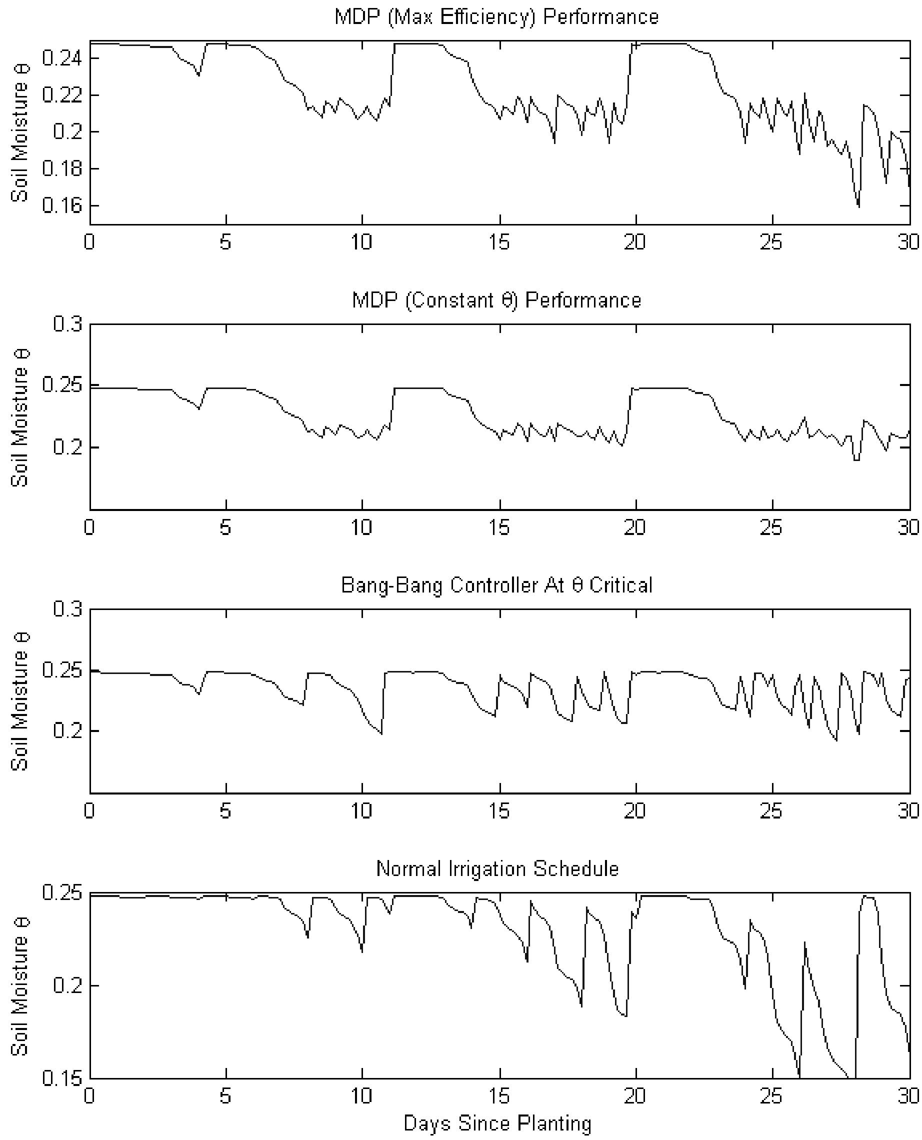


Figure 5.52: Soil Moisture Status, Comparison of Varying Control Strategies

5.5.1 Results

MDP, Maximum Water Efficiency Configuration:

Modelled Final Values:

Dry Shoot Biomass: 0.706 grams

Leaf Area: 13265 mm2

Total Applied Irrigation: 0.245 Litres

Irrigation Approach Achieved A Score Of: **54143.7**

MDP, Constant Theta Control:

Modelled Final Values:

Dry Shoot Biomass: 0.753 grams

Leaf Area: 13883 mm2

Total Applied Irrigation: 0.28 Litres

Irrigation Approach Achieved A Score Of: **49585**

Normal Irrigation Schedule:

Subirrigation to $\theta = \frac{\theta_{fc}+\theta_{sat}}{2}$ every 48 hours or when wilting is first noticed.

Modelled Final Values:

Dry Shoot Biomass: 0.607 grams

Leaf Area: 11502 mm2

Total Applied Irrigation: 1.38545 Litres

Irrigation Approach Achieved A Score Of: **8302.1**

Bang-Bang Controller on $\theta = \theta_{CRITICAL}$

Leading Manually Programmed Algorithm.

Modelled Final Values:

Dry Shoot Biomass: 0.763 grams

Leaf Area: 14060 mm2

Total Applied Irrigation: 0.289093 Litres

Irrigation Approach Achieved A Score Of: **48635.9**

5.6 Algorithm Performance (Field Testing)

5.6.1 Construction

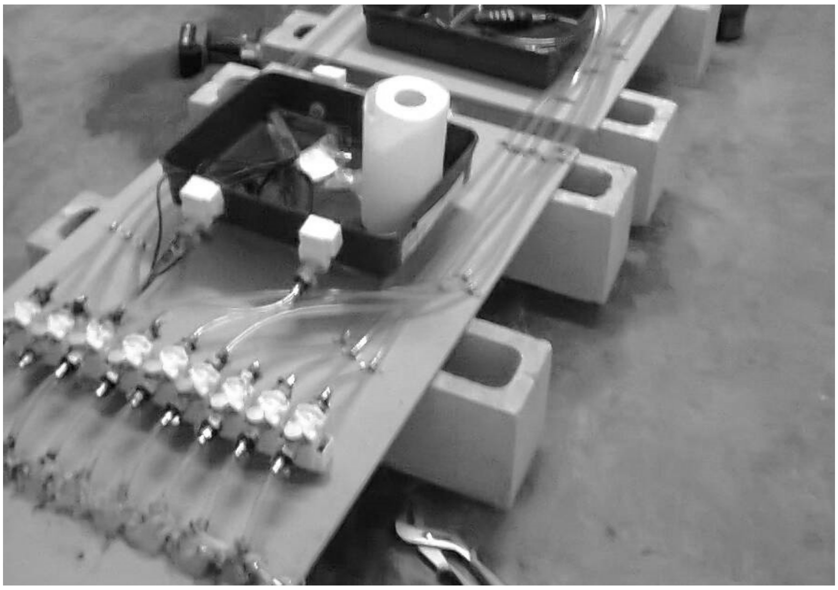


Figure 5.53: First Phase of The Field Experiment Contruction, Showing Valve Assembly

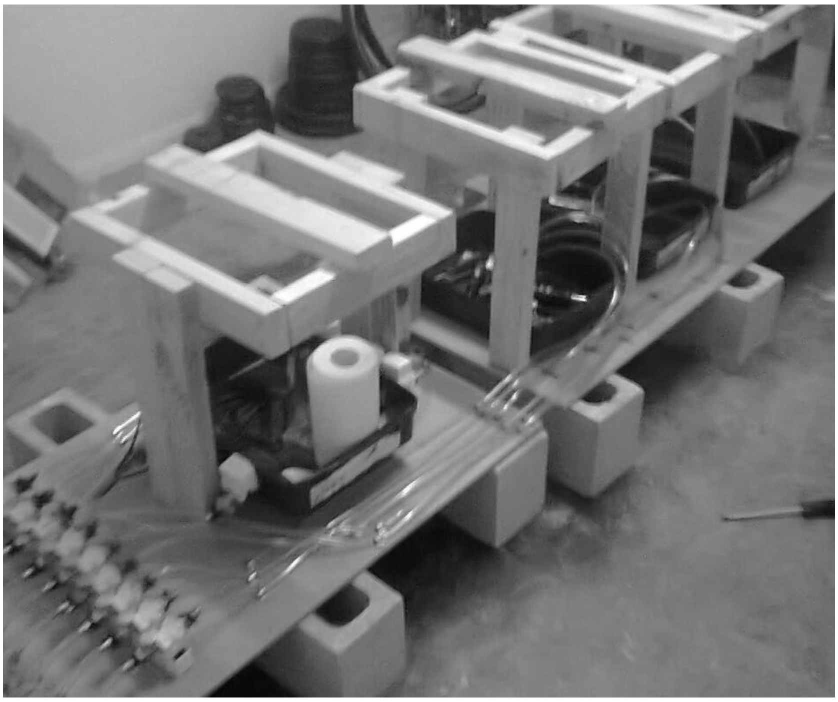


Figure 5.54: Field Experiment Contruction, Overhead Stands and Propagation Trays

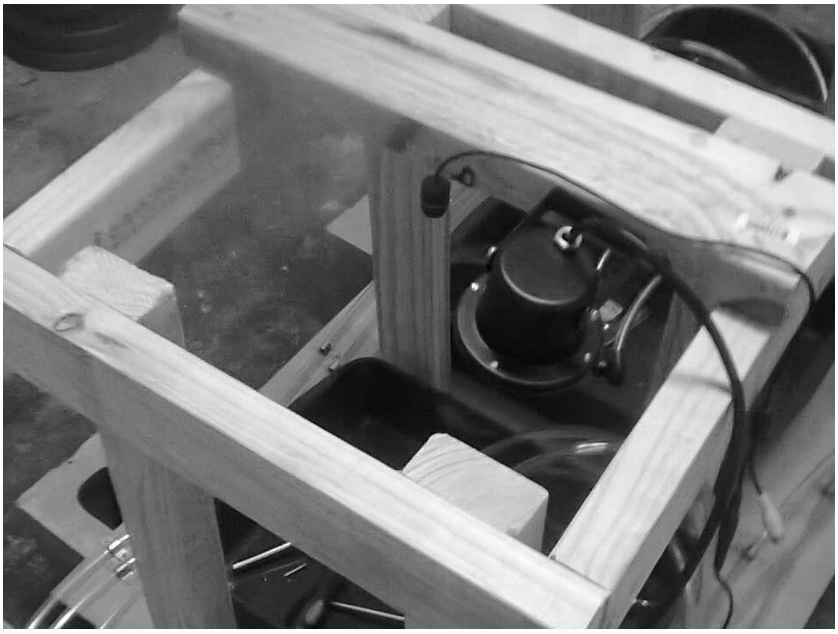


Figure 5.55: Field Experiment Contruction, Overhead Vision/Illumination System

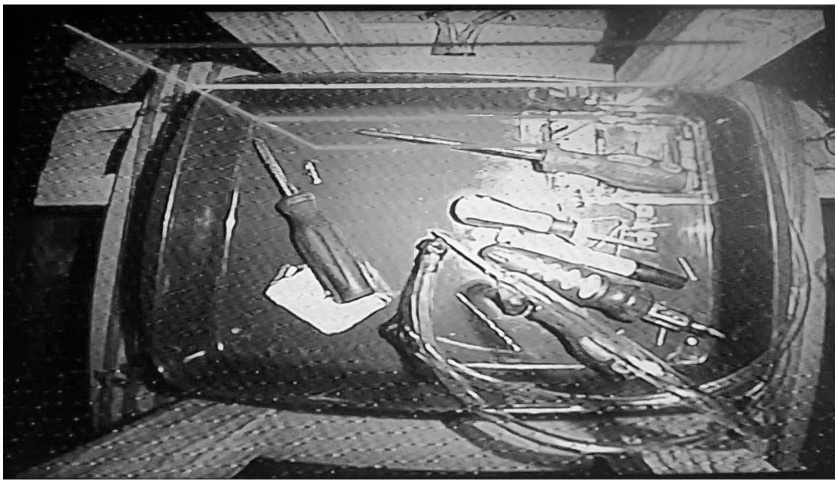


Figure 5.56: Field Experiment Contruction, Testing Camera Assembly

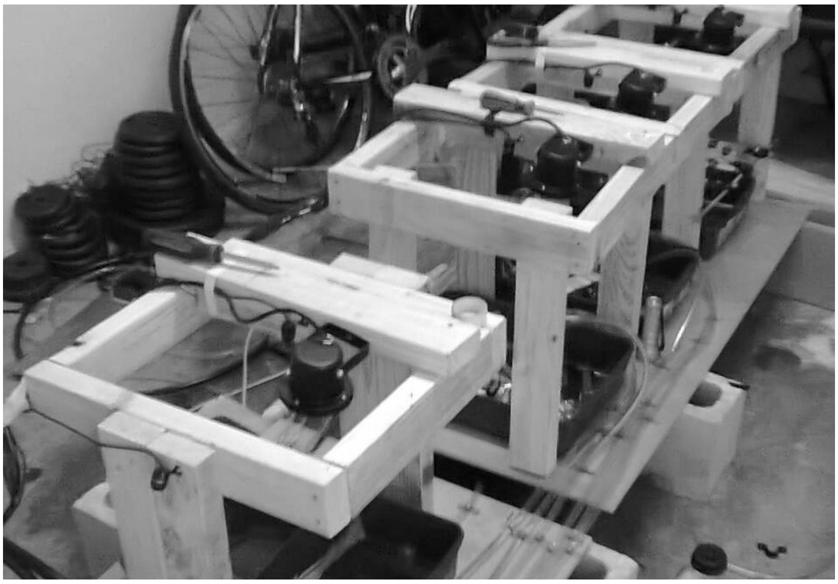


Figure 5.57: Field Experiment Contruction, Completed Vision System

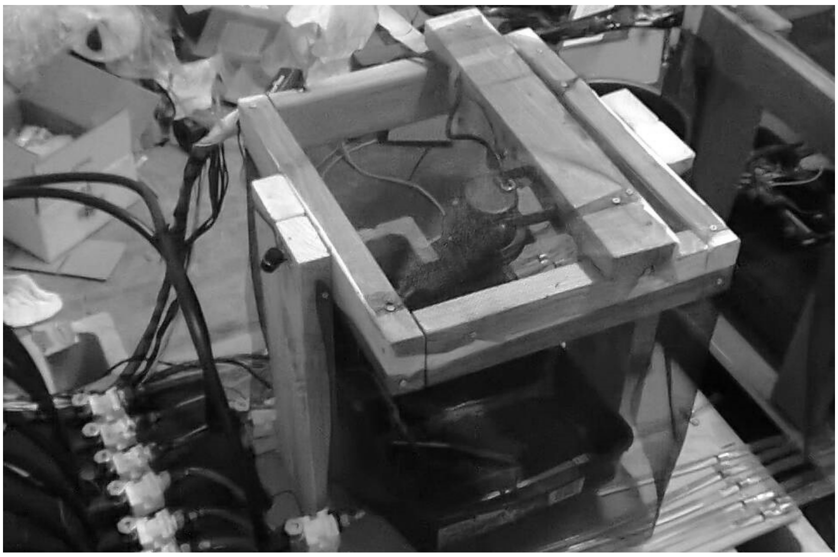


Figure 5.58: Field Experiment Contruction, Insect Proofing/Environment Protection

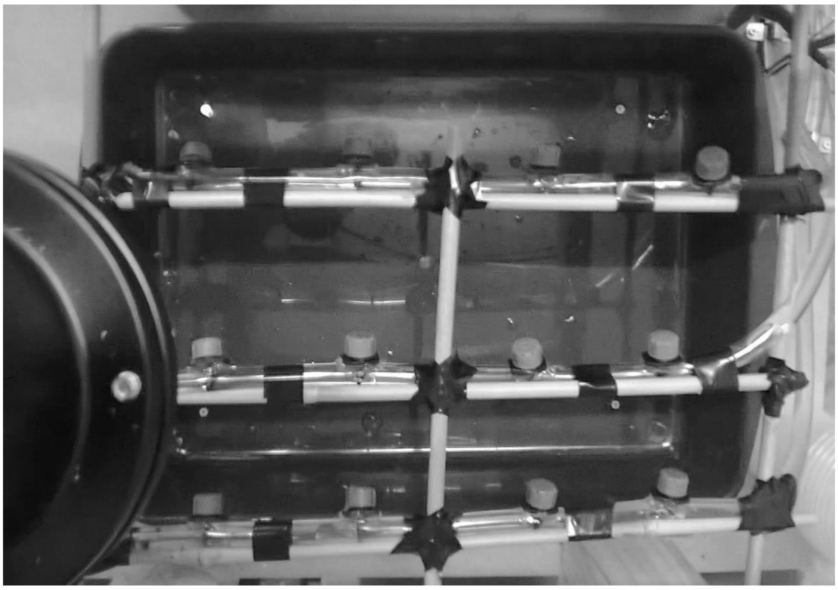


Figure 5.59: Field Experiment Contruction, Drip Irrigator Array

Figure 5.60: Completed Field Experiment Test Apparatus

5.6.2 Field Installation

Figure 5.61: Field Experiment Installed At Test Location

Figure 5.62: Field Experiment Processing Cabinet

5.6.3 Results

Due to time constraints and unrelenting technical issues the project was abandoned. The components used in its construction have been packed away and the apparatus assembly deconstructed.

The results of the model validation experiments should hopefully provide some reassurance in the utility of machine learning and its application to irrigation control.

Bit of a shame really but one important skill at managing something like this is recognizing when an investment has gone bad. Best regards, Damian.

Chapter 6

Discussion

6.0.1 A Brief Overview Of The Results

GFS And Fused Data Comparisons:

The GFS model performs far bettter during the summer months.
The fused data provides excellent summer performance R2 = 92.3%
The fused data provides acceptable winter season performance R2 = 68.2%

Leaf/Soil Models:

Leaf temperature is typically 4c lower than air, not listed in the results but I did check that temperature profile against observations and it matched.
5cm soil temperature can be nearly 7c cooler than air, profiles matched CSIRO paper.

Transpiration:

Transpiration varies exponentially with air temperature, low soil moisture blunts the effect. Exactly as expected.
Transpiration varies linearly with relative humidity, up until a saturation point.

The geometry of the Arabidopsis leaf makes it resistant to water loss through wind. Very Interesting.

Clustering:

Data generated by the bang bang controller generates a nice distribution of 8 centroid locations.
tSNE is great at compressing dimensions.

Value Iteration:

Stochastic gradient is many times faster than batch gradient descent.
The code profiler is a good warning to the phenomenon of premature optimization.
The authors algorithm converges very quickly, within roughly 100 iterations.

Value Iteration Results:

The fact every value is identical is NO BUG! It's math that even I can't wrap my head around. Somehow they've all "converged" to something I truly am bewildered by, the performance is also bewilderingly incredible ... what the heck just happened?!?
A validation error under 2

Model Validation Results:

The arabidopsis model is a surprisingly good match for the saturated biomass test.

Soil Sensor Calibration:

The graph of voltage to soil moisture is highly erratic and difficult to interpret.

Arabidopsis Samples:

They arrived in great condition from LEHLE and germanation was equally excellent.

Model Validation, Field Test:

The author didn't think the plants were dieing until the photos were put side by side.
Totally accidental mistake not giving them enough light.
The overhead vision system was able to detect the decline in plant health very reliably.
The crop model shows an extremely blunted growth curve, the plant is stagnating and dieing, it predicted it!
Nice work to the authors programming.

Algorithm performance:

Total biomass is very similar except for the normal irrigation control which is slightly blunted.
The Constant theta control performs surprisingly well on the scoring.
The developed algorithm is 6.5x as efficient as the normal schedule.
The developed algorithm is 10

Algorithm performance (field trial):

Very ambitious build which was completed.
Sad to see time run out.

Chapter 7

Conclusion

All in all the research was a very rewarding experience! The author's knowledge of machine learning has definitely gone up a few notches. Despite the fact that the time constraints on the completion of the project got in the way there was some intriguing and promising discoveries.

This project led to the development of a primitive Arabidopsis Crop model, at this point in time this may be the first crop model developed for the hugely popular species. While the model is required to undergo far more stringent testing before it could be used for more serious applications, initial trials have been great and definitely promising!

There are likely plenty of bugs and corner cases that need to be ironed out but it's good to see something working. The author is tempted to take the technologies he developed for the project and apply them to other species. Namely "Atlantic Giant Pumpkins", an old hobby some new technology might be able to put a spin on.

The project also led to the development of a general, highly multithreaded MDP solver. The performance was great and it introduced the author to many aspects of high performance computing. To be fair there is likely still a huge room for improvement in the algorithm but it was incredible to witness just how fast reinforcement learning can be.

The third interesting outcome came from the MDP solver itself, for some reason with the right Basis Function width it is possible to compress a vector of weights values into a single scalar. Witnessing that was a huge wow moment and the author now must figure out the secret to the math.

And one cannot forget the performance of the developed irrigation algorithm, beating human controls by nearly 600big data can come up with!

References

A. McCarthy et al. (2008), 'Towards evaluation of adaptive control systems for improved site-specific irrigation of cotton', *Proceedings of Irrigation Association of Australia National Conference and Exhibition* .

Alison C. McCarthy et al. (2010), 'Variwise: A general-purpose adaptive control simulation framework for spatially and temporally varied irrigation at sub-field scale', *Computers and Electronics in Agriculture* **70**(1), 117 – 128.

A.M. Wilczek et al. (2009), 'Effects of genetic perturbation on seasonal life history plasticity', *Science* **323**(5916), 930–934.

A.N. Dodd et al. (2006), 'Time of day modulates low-temperature ca2+ signals in arabidopsis', *The Plant Journal* **48**(6), 962–973.

B.A Keating et al. (1999), 'Modelling sugarcane production systems i. development and performance of the sugarcane module', *Field Crops Research* **61**(3), 253 – 271.

B.A Keating et al. (2003), 'An overview of apsim, a model designed for farming systems simulation', *European Journal of Agronomy* **18**(34), 267 – 288.

Bellman, R. (1957), 'A markovian decision process', *Indiana Univ. Math. J.* **6**.

CHARLES-EDWARDS, D. A. & FISHER, M. J. (1980), 'A physiological approach to the analysis of crop growth data. i. theoretical considerations', *Annals of Botany* **46**(4), 413–423.

Chew, Y. H. & Halliday, K. J. (2011), 'A stress-free walk from arabidopsis to crops', *Current Opinion in Biotechnology* **22**(2), 281 – 286.

C.H.M. van Bavel et al. (1996), 'Automatic irrigation based on monitoring plant transpiration', *Evapotranspiration and irrigation scheduling Proceedings of the International Conference* .

C.J. Phene et al. (1971), 'Measuring soil matric potential in situ by sensing heat dissipation within a porous body', *Soil Sci. Soc. Am. J.* .

C.J. Phene et al. (1972), 'Controlling automated irrigation with a soil matric potential sensor', *Transactions of the american society of agricultural engineers Vol. 16* .

D. Aimar et al. (2011), 'Drought tolerance and stress hormones: From model organisms to forage crops', *Plants and Environment Chap. 6* .

de Wit, C. (1965), 'Photosynthesis of leaf canopies', *Agricultural Research Report 663* .

de Wit, C. (1978), *Simulation of assimilation, respiration, and transpiration of crops*, Wiley.

Decker, J. P. & Skau, C. M. (1964), 'Simultaneous studies of transpiration rate and sap velocity in trees', *Plant Physiology* **39**(2), 213–215.

D.W. Meinke et al. (1998), 'Arabidopsis thaliana: A model plant for genome analysis', *Science* **282**(5389), 662–682.

EJ Hartman et al. (1990), 'Universal approximation using radial-basis-function networks', *Neural Computation* .

Evans, R. (2006), 'Irrigation technologies', `http://www.sidney.ars.usda.gov/Site_Publisher_Site/pdfs/personnel/Irrigation%20Technologies%20Comparisons.pdf`.

F. Capraro et al. (2008), Neural network-based irrigation control for precision agriculture, *in* 'Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on', pp. 357–362.

Fahrmeir, L. (1999), 'State space models a brief history and some recent developments', *Proceedings of the 52nd ISI Session* .

Fisher, M. (1969), 'The growth and development of townsville lucerne (stylosanthes humilis) in ungrazed swards at katherine, nt. aust', *Agric. Anim. Husb. 9:* .

Fuchs, M. (1990), 'Infrared measurement of canopy temperature and detection of plant water stress', *Theoretical and Applied Climatology* **42**(4), 253–261.

G. Jakab et al. (2005), 'Enhancing arabidopsis salt and drought stress tolerance by chemical priming for its abscisic acid responses', *Plant Physiology* **139**(1), 267–274.

G.D. Bilsborough et al. (2011), 'Model for the regulation of arabidopsis thaliana leaf margin development', *Proceedings of the National Academy of Sciences* .

Giussani, L. (2001), 'A molecular phylogeny of the grass subfamily panicoideae (poaceae) shows multiple origins of c4 photosynthesis', *American Journal of Botany* **88**(11), 1993–2012.

Hearn, A. & Roza, G. D. (1985), 'A simple model for crop management applications for cotton (gossypium hirsutum l.)', *Field Crops Research* **12**(0), 49 – 69.

HL Penman (1952), 'The physical bases of irrigation control.', *13th International hort. Congress* .

Hoffmann, M. H. (2002), 'Biogeography of arabidopsis thaliana (l.) heynh. (brassicaceae)', *Journal of Biogeography* **29**(1), 125–134.

Horton, B. (2012), 'Models for estimation of hourly soil temperature at 5 cm depth and for degree-day accumulation from minimum and maximum soil temperature', *Soil Research* .

Howard, R. (1960), *Dynamic Programming and Markov Processes.*, MIT Press.

Jackson, R. (1982), 'Canopy temperature and crop water stress', *Advances in Irrigation Research* .

Jackson, R. (1985), 'Evaluating evapotranspiration at local and regional scales', *Proceedings of the IEEE* **73**(6), 1086–1096.

J.E. Bergez et al. (2001), 'Moderato: an object-oriented decision tool for designing maize irrigation schedules', *Ecological Modelling* **137**(1), 43 – 60.

J.E. Bergez et al. (2002), 'Improving irrigation schedules by using a biophysical and a decisional model', *European Journal of Agronomy* **16**(2), 123 – 135.

Jensen, M. (1969), 'Scheduling irrigation with computers', *Journal of soil and water conservation Vol. 24* .

Jensen, M. (1972), 'Programming irrigation for greater efficiency.', *D. Hillel (ed.) Optimizing the soil physical environment toward greater crop yields.* .

J.K. Ward et al. (2000), 'Is atmospheric co2 a selective agent on model c3 annuals?', *Oecologia* **123**(3), 330–341.

Karimaldini, F. et al. (2012), 'Daily evapotranspiration modeling from limited weather data by using neuro-fuzzy computing technique', *Journal of Irrigation and Drainage Engineering* **138**(1), 21–34.

Keating, B. & Agriculture, C. T. (2001), *NWheat: Documentation and Performance of a Wheat Module for APSIM*, Tropical agriculture technical memorandum, CSIRO Tropical Agriculture.

K.J. Boote et al. (1996), 'Potential uses and limitations of crop models', *Agronomy Journal Vol. 88* .

K.S. Raju et al. (1983), 'Canopy temperature and crop water stress', *Advances in Irrigation Vol. 2* .

Mendoza, L. & Alvarez-Buylla, E. (1998), 'Dynamics of the genetic regulatory network forarabidopsis thalianaflower morphogenesis', *Journal of Theoretical Biology* **193**(2), 307 – 319.

Meyerowitz, E. M. (1987), 'Arabidopsis thaliana', *Annual Reviews of Genetics Vol. 21* .

Meyerowitz, E. M. (2001), 'Prehistory and history of arabidopsis research', *Plant Physiology* **125**(1), 15–19.

Meyerowitz, E. & Somerville, C. (1994), *Arabidopsis*, Cold Spring Harbor monograph series, Cold Spring Harbor Laboratory Press.

M.J. Robertson et al. (1999), Simulating growth and development of canola in australia, *in* 'Simulating growth and development of canola in Australia'.

M.J. Robertson et al. (2002), 'Simulation of growth and development of diverse legume species in apsim', *Australian Journal of Agricultural Research 53* .

Morison, J. & Gifford, R. (1983), 'Stomatal sensitivity to carbon dioxide and humidity', *Plant Physiology Vol. 71* .

Nakashima, K. (2009), 'Transcriptional regulatory networks in response to abiotic stresses in arabidopsis and grasses', *Plant Physiology* **149**(1), 88–95.

New York Times (2009), 'So much food. so much hunger', `http://www.nytimes.com/2009/09/20/weekinreview/20martin.html`.

Ng, A. (2012), 'Cs229 lecture notes', `http://cs229.stanford.edu/notes/cs229-notes1.pdf`.

O.A.M. Lewis et al. (1988), 'Effect of nitrogen source on growth response to salinity stress in maize and wheat', *New Phytologist Vol. 111* .

O'Keeffe, K. (2009), *Maize growth and development*, Department Of Primary Industries, NSW Australia.

O.W. Israelsen et al. (1944), 'Bulletin no. 311 - water-application efficiencies in irrigation', *UAES Bulletins. Paper 273* .

P. Mart et al. (2013), 'An artificial neural network approach to the estimation of stem water potential from frequency domain reflectometry soil moisture measurements and meteorological data', *Computers and Electronics in Agriculture* **91**(0), 75 – 86.

Park, J. & Sandberg, I. W. (1991), 'Universal approximation using radial-basis-function networks', *Neural Computation* .

P.B. de Reuille et al. (2006), 'Computer simulations reveal properties of the cell-cell signaling network at the shoot apex in arabidopsis', *Proceedings of the National Academy of Sciences of the United States of America* **103**(5), 1627–1632.

Peck, A. (1983), 'Field variability of soil physical properties', *Advances in Irrigation Vol. 2* .

Phene, C. & Howell, T. (1982), 'Soil sensor control of high-frequency irrigation systems', *Transactions of the american society of agricultural engineers Vol. 27* .

Q. Zhang et al. (1996), Application of fuzzy logic in an irrigation control system, *in* 'Industrial Technology, 1996. (ICIT '96), Proceedings of The IEEE International Conference on', pp. 593–597.

R. Jaarsma et al. (2013), 'Effect of salt stress on growth, na¡sup¿+¡/sup¿ accumulation and proline metabolism in potato (¡italic¿solanum tuberosum¡/italic¿) cultivars', *PLoS ONE* **8**(3), e60183.

R.D. Jackson et al. (1981), 'Canopy temperature as a crop water stress indicator', *Water Resources Research* **17**(4), 1133–1138.

Recht, B. (2005), 'Kernel methods', `http://fab.cba.mit.edu/classes/MIT/864.05/rkhs/Rec.RKHS.pdf`.

R.G. Allen et al. (1998), *Crop evapotranspiration: Guidelines for computing crop water requirements*, FAO.

Russell, S. & Norvig, P. (2010), *Artifcial Intelligence, A Modern Approach*, Prentice Hall.

S. Dabach et al. (2013), 'Numerical investigation of irrigation scheduling based on soil water status', *Irrigation Science* **31**(1), 27–36.

Schlenker, W. & Roberts, M. J. (2006), 'Nonlinear effects of weather on corn yields', *Applied Economic Perspectives and Policy* **28**(3), 391–398.

S.R. Evett et al. (1996), 'Canopy temperature based automatic irrigation control', *Evapotranspiration and irrigation scheduling Proceedings of the International Conference* .

S.R. Evett et al. (2000), 'Automatic drip irrigation of corn and soybean', *Proceedings of the 4th decennial national irrigation symposium* .

Sutton, R. & Barton, A. (1998), *Reinforcement Learning An Introduction*, MIT Press.

Tester, M. & Bacic, A. (2005), 'Abiotic stress tolerance in grasses. from model plants to crop plants', *Plant Physiology* **137**(3), 791–793.

United Nations (2007), 'Water scarcity', `http://www.un.org/waterforlifedecade/scarcity.shtml`.

van Dodeweerd, A. (1999), 'Identification and analysis of homoeologous segments of the genomes of rice and arabidopsis thaliana', *Genome / National Research Council Canada* .

Warrick, A. & Yates, S. (1987), 'Crop yield as influenced by irrigation uniformity', *Advances in Irrigation Vol. 4* .

W.J.S. Downton et al. (1988), 'Stomatal closure fully accounts for the inhibition of photosynthesis by abscisic acid', *New Phytologist* **108**(3), 263–266.

World Weather Online (2013), 'Fresno air terminal airport (fat) weather, united states weather averages', `http://www.worldweatheronline.com/v2/weather-averages.aspx?q=FAT`.

Y. Cohen et al. (1981), 'Improvement of the heat pulse method for determining sap flow in trees', *Plant, Cell and Environment* **4**(5), 391–397.

Yin, X. & Struik, P. C. (2009), 'Modelling the crop: from system dynamics to systems biology', *Journal of Experimental Botany* .

Z.M. Pei et al. (2011), 'Calcium channels activated by hydrogen peroxide mediate abscisic acid signalling in guard cells', *Nature* .

University of Southern Queensland

Faculty of Health, Engineering & Sciences

Adaptive Markov Decision Control Of High-Frequency Drip Irrigation Systems

Volume 2, Appendices

A dissertation submitted by

Damian Peckett

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Mechatronic Engineering

Submitted: October, 2013

Appendix A

Project Specification

ENG4111/4112 Research Project
Project Specification

FOR: **Damian PECKETT**

TOPIC: ARTIFICIALLY INTELLIGENT CROP IRRIGATION

SUPERVISOR: Dr. Alison McCarthy
Dr. Nigel Hancock

PROJECT AIM: To design and implement an "Artificially Intelligent" irrigation control algorithm that can potentially outperform an untrained human in the task of water efficient *Arabidopsis Thaliana* cultivation.

PROGRAMME: (Issue A, 13 March 2013)

1. Research environmental cues that will be used to train the algorithm, features important to determining if a plant requires irrigation.
2. Implement an initial rough computer model/simulation of the organism *Arabidopsis Thaliana.* [in the biological science community, *Arabidopsis T.* enjoys the status of being a model organism. As such it is widely studied and reported on within literature]
3. Design a reward function for the algorithm/s, which will balance the trade-off between growth and water consumption.
4. Research the available publications relating to *Arabidopsis Thaliana* irrigation and implement existing baseline comparison algorithm/s.
5. Investigate the performance of several machine learning algorithms on the simulator (e.g. Artificial Neural Networks, Support Vector Machines, etc.) for the task of unsupervised crop irrigation. [I have already undertaken significant background study important to this area, including existing fully implemented algorithm code].
6. Build a real world test apparatus comprising of an array of irrigation control solenoids and a vision system utilizing a commercial webcam, (possible sensors, including temperature humidity, etc., however first trials will utilize existing equipment) [Equipment support is available from NCEA and, if necessary, personal resources.]
7. Design an experimental program to evaluate the various algorithm/s.
8. Train leading simulation algorithm/s on the real world test apparatus.
9. Compare the performance of algorithm/s to a human control (normal irrigation schedule).

If time permits:
1. Prepare an academic paper on the research.
2. Train the algorithm on other plants, including potentially commercial crops.

AGREED ______________ (Student) ______________ (Supervisor)

Date: 13/3/2013 Date: 13 /3 / 2013

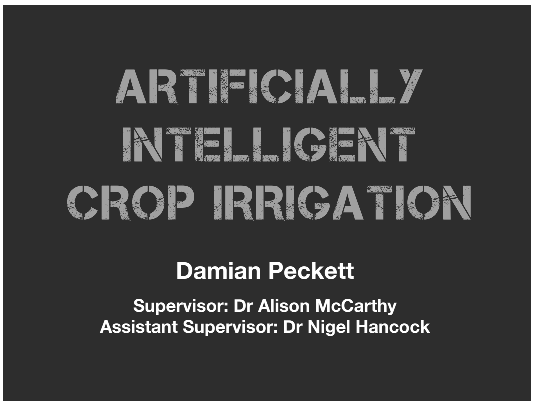
______________ (Associate Supervisor)

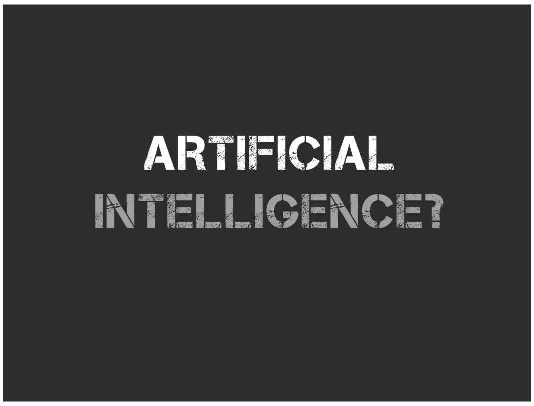
Date: 13/3/2013

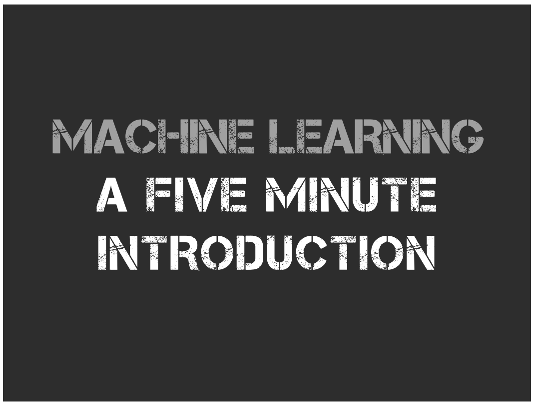
Examiner/Co-examiner: ______________ Date: / / 2013

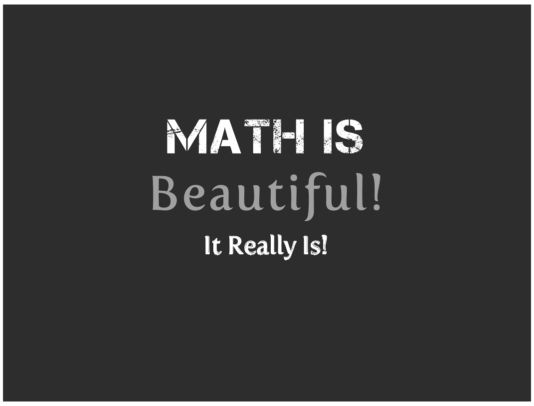
Appendix B

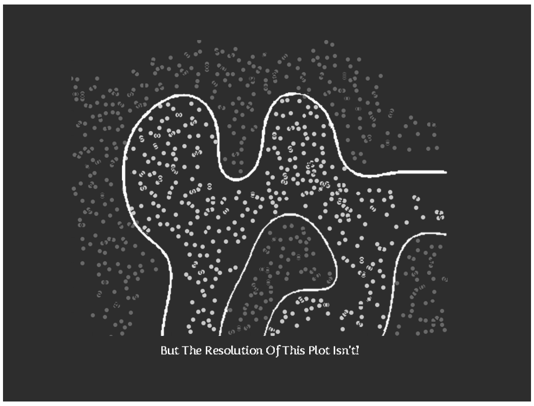
Seminar Slides






MACHINE LEARNING
A FIVE MINUTE
INTRODUCTION


MATH IS
Beautiful!
It Really Is!


But The Resolution Of This Plot Isn't!


REINFORCEMENT
OR CLASSIFICATION?


Sometimes There isn't
Just a "RIGHT" Answer


GROWING A PLANT?
REINFORCEMENT!

MARKOV DECISION PROCESS

- Construct a Model
- Develop a Value Estimator
- Solve Belmann's Equations
- ???
- PROFIT (Meme Reference)

A MODEL?

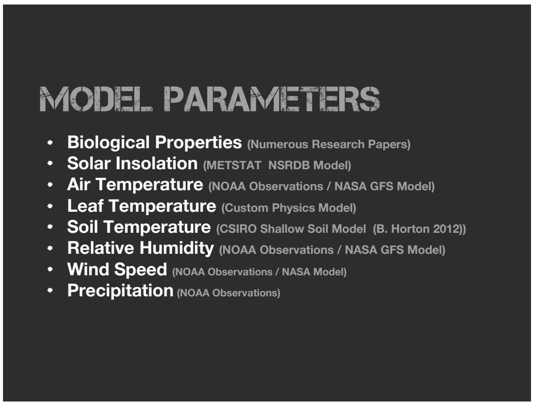
ARABIDOPSIS THALIANA

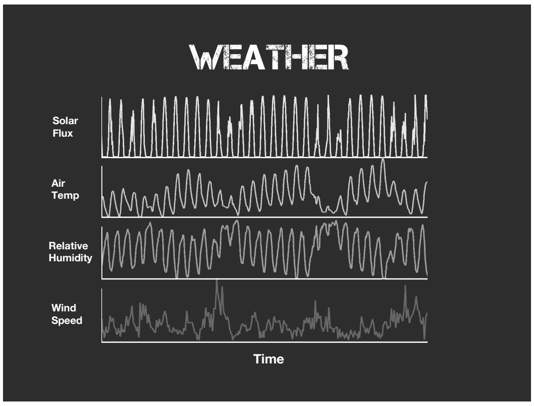
5 WEEKS
5 CHROMOSOMES
50 YEARS

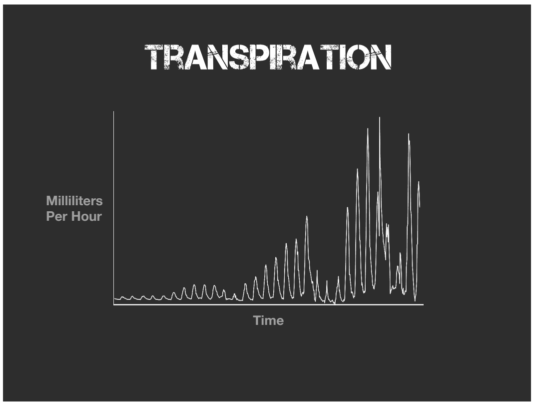
FRESNO CALIFORNIA

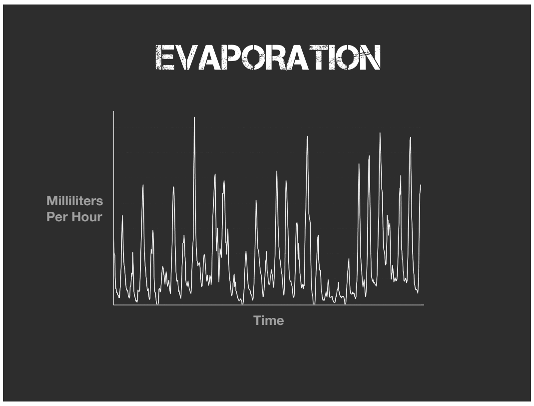
SIMILAR CONDITIONS
HUMAN IRRIGATION
GREAT DATA

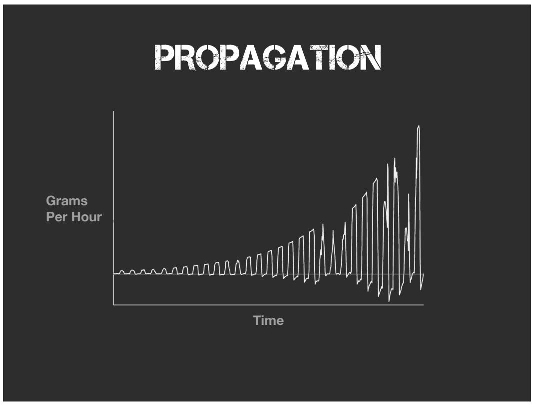

BIG DATA
and 2000 lines of C

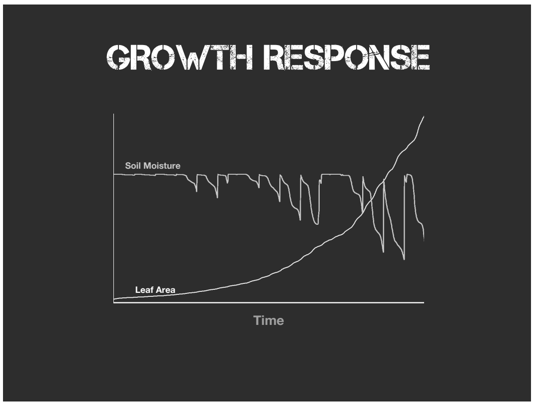

MODEL PARAMETERS
• Biological Properties (Numerous Research Papers)
• Solar Insolation (METSTAT NSRDB Model)
• Air Temperature (NOAA Observations / NASA GFS Model)
• Leaf Temperature (Custom Physics Model)
• Soil Temperature (CSIRO Shallow Soil Model (B. Horton 2012))
• Relative Humidity (NOAA Observations / NASA GFS Model)
• Wind Speed (NOAA Observations / NASA Model)
• Precipitation (NOAA Observations)


WEATHER
Solar Flux
Air Temp
Relative Humidity
Wind Speed
Time

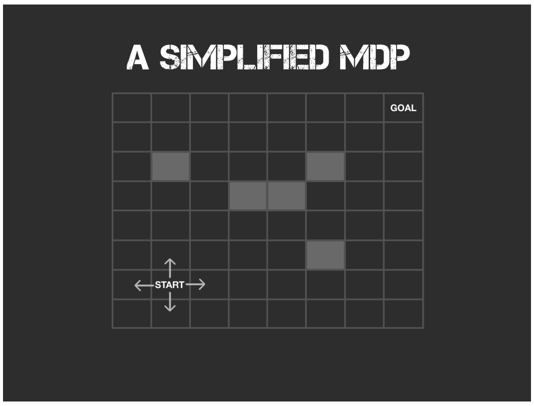

TRANSPIRATION
Milliliters Per Hour
Time

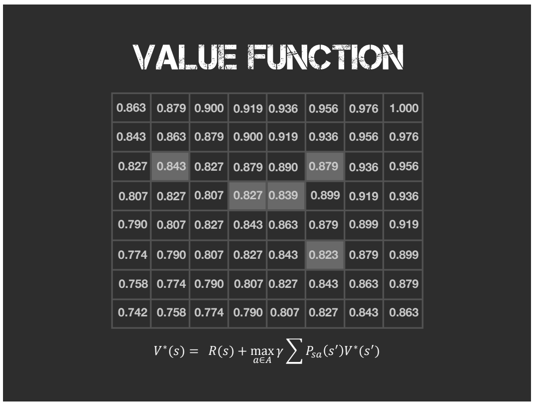

EVAPORATION
Milliliters Per Hour
Time

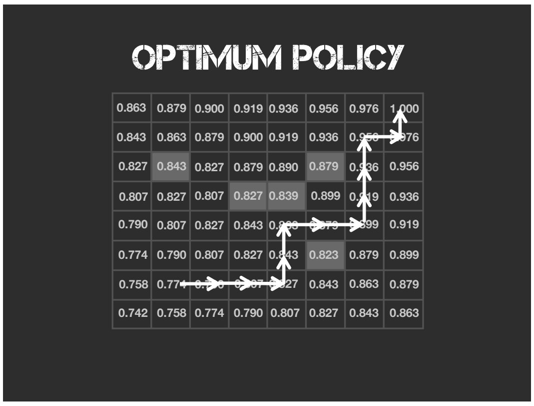

PROPAGATION
Grams Per Hour
Time

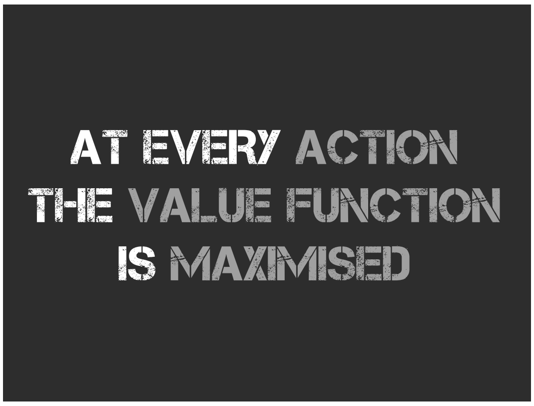

GROWTH RESPONSE
Soil Moisture
Leaf Area
Time


REINFORCEMENT
LEARNING


A SIMPLIFIED MDP
GOAL
START


VALUE FUNCTION
0.863 0.879 0.900 0.919 0.936 0.956 0.976 1.000
0.843 0.863 0.879 0.900 0.919 0.936 0.956 0.976
0.827 0.843 0.827 0.879 0.890 0.879 0.936 0.956
0.807 0.827 0.807 0.827 0.839 0.899 0.919 0.936
0.790 0.807 0.827 0.843 0.863 0.879 0.899 0.919
0.774 0.790 0.807 0.827 0.843 0.823 0.879 0.899
0.758 0.774 0.790 0.807 0.827 0.843 0.863 0.879
0.742 0.758 0.774 0.790 0.807 0.827 0.843 0.863
$V^*(s) = R(s) + \max_{a\in A}\gamma \sum P_{sa}(s')V^*(s')$

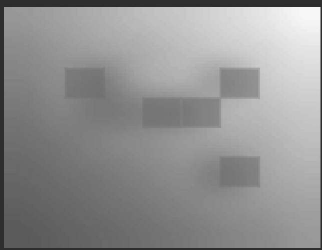

OPTIMUM POLICY
0.742 0.758 0.774 0.790 0.807 0.827 0.843 0.863


AT EVERY ACTION
THE VALUE FUNCTION
IS MAXIMISED

BUT SOMETIMES YOU CAN'T USE TILES!

VALUE ESTIMATION, USING REGRESSION

CONTINUOUS MDP



BUT THE CONTOUR MAP IS COMPLEX!

Try fitting a line of best fit to that!

INTRODUCING THE HERO!

GAUSSIAN KERNEL RBF

$$\emptyset_x = \exp(-\gamma\|x - x_k\|^2)$$

X Is The Current State,
X_k Is The RBF Centroid,
γ Is The Inverse Variance.

COMPLEX SHAPES REQUIRE COMPLEX FEATURE SPACES!

FEATURE DIMENSIONS

Taylor Series Expansion Of The Exponential Function:

$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

In the Gaussian RBF x is equal to the square of the L2 distance from the radial basis centroid. Therefore:

INFINITE! FEATURE DIMENSIONS!

WITH ENOUGH RBF'S YOU CAN ESTIMATE ANY FUNCTION!

MY ALGORITHM

- Value Estimation MDP
- Linear Combination Of Basis Functions
- Eight Gaussian Radial Basis Functions
- Fourteen Input Features

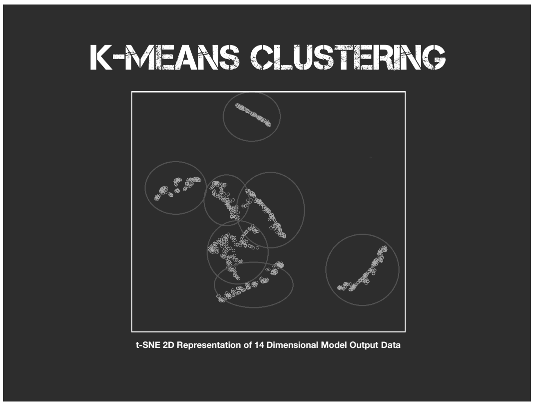
INPUT FEATURES

- Current Air Temperature
- Current Relative Humidity
- Current Solar Insolation
- Current Soil Water Availability Ratio
- Current Leaf Area
- Current Wind Speed
- 24 Hour Predicted Average Air Temperature
- 24 Hour Predicted Average Relative Humidity
- 24 Hour Predicted Average Wind Speed
- 24-48 Hour Predicted Average Air Temperature
- 24-48 Hour Predicted Average Relative Humidity
- 24-48 Hour Predicted Average Wind Speed
- Total Solar Energy Last 24 Hours
- Total Applied Irrigation

Phew that was a long list!

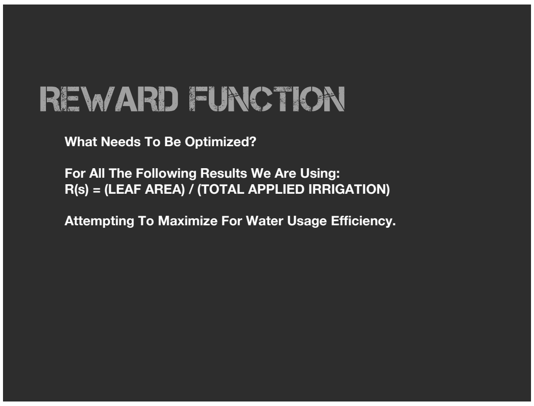
RBF CENTERS?

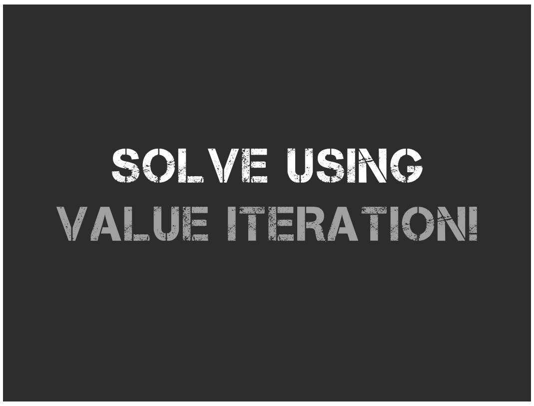

RUN THE MODEL
WITH A BANG-BANG
CONTROLLER


K-MEANS CLUSTERING
t-SNE 2D Representation of 14 Dimensional Model Output Data


CENTER THE RBF'S
ON THE CLUSTERS!

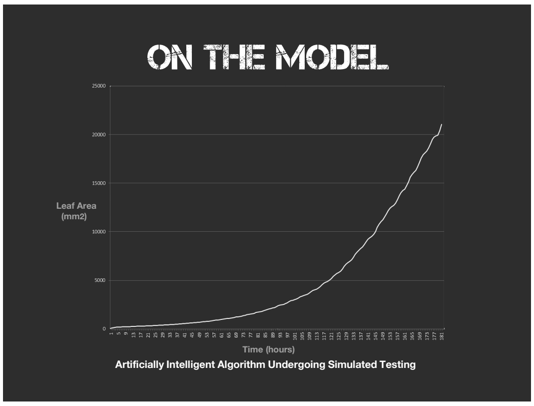

FIXED VARIANCE
FOUND USING
GRID SEARCH


REWARD FUNCTION
What Needs To Be Optimized?
For All The Following Results We Are Using:
R(s) = (LEAF AREA) / (TOTAL APPLIED IRRIGATION)
Attempting To Maximize For Water Usage Efficiency.

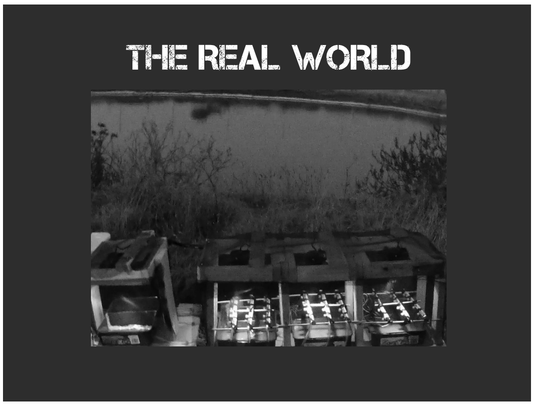

SOLVE USING
VALUE ITERATION!

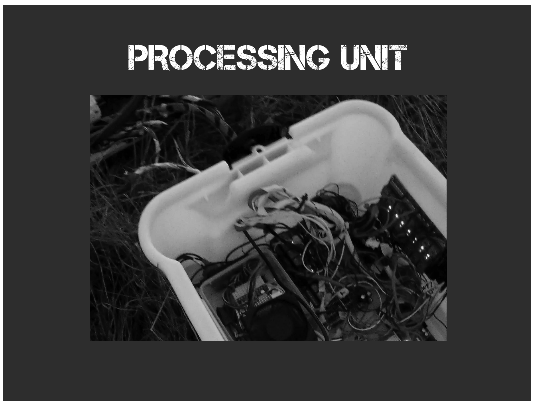

TRAINING RESULTS
Xk = [
297 41 356 0.162 16983 1.9 299 38 1.7 300 36 2 30 0.976
291 53 281 0.153 9173 2.2 291 53 2.3 291 52 2.3 25 0.533
293 48 301 0.168 2169 1.7 293 49 1.8 292 52 1.9 27 0.271
303 34 354 0.157 22727 1.9 303 32 2.3 300 28 2.4 31 1.5
292 49 359 0.152 12487 2.4 292 49 2.5 295 45 2.3 27 0.763
291 47 314 0.152 6300 2.4 291 47 2.2 291 49 2.1 29 0.45
291 49 392 0.174 609 2.5 290 52 2.5 291 52 2.5 26 0.107
290 55 301 0.183 3974 2.3 290 55 2.4 290 52 2.5 24 0.352]
Θ = [78519 63857 74194 76854
53085 70596 76115 76447 76815]
HOLDOUT CROSS VALIDATION ERROR: 3.6 PERCENT
The training worked!


LET'S TEST IT!


ON THE MODEL
25000
20000
15000
10000
5000
0
Leaf Area (mm2)
Time (hours)
Artificially Intelligent Algorithm Undergoing Simulated Testing


MODEL RESULTS
Normal Irrigation Schedule (Sub Irrigation):
Dry Shoot Biomass: 0.684 grams
Irrigation Approach Achieved A Score Of: 7011
Bang-Bang Control, Drip Irrigation:
Dry Shoot Biomass: 1.07 grams
Irrigation Approach Achieved A Score Of: 25673
Artificial Intelligence , Drip Irrigation:
Dry Shoot Biomass: 1.15 grams
Irrigation Approach Achieved A Score Of: 26335


THE REAL WORLD


PROCESSING UNIT

ARABIDOPSIS THALIANA

TEST APPARATUS

- 48 Test Specimens, Located in Four Propagation Trays
- 11 Soil Resistivity Sensors
- 2 Solar Insolation Meters
- 2 Humidity/Temperature Sensors
- 1 Weight Sensor
- 13 Solenoids
- 5 Overhead Cameras
- 4 Overhead Spot Illumination Lamps
- VIA Pico-ITX Board Running CentOS 6, OpenVPN , and an entirely custom software stack written in C.
- Arduino Mega 2560 as an embedded controller.
- Linode Virtual Server running AI Code.
- Weather Data From <http://meteojs.com>

MODEL VALIDATION

In the next two weeks experimental tests shall be carried out to:

- Confirm The Functionality Of The Crop Model
- Confirm The Functionality Of The Test Apparatus
- Eventually Confirm The Functionality Of The Intelligent Algorithm

If I have time of course ...

CONCLUSION

- It Works!
- Versatile Performance!
- Promising Avenue For The Future!

Congratulations on making it through my presentation!
Thanks to everyone for the attention! It was a very long night putting all this together.

Appendix C

Project Consequences

This particular project is brings about almost universal benefits to society, demand for food is only going to increase in the coming decades and it's the ethical responsibility of engineers to consider the consequences of such demands.

This project brings about the potential of increased crop yield, increased safety (through limiting runoff and excess pesticide/fertilizer usage), increased sustainability (through reduced water usage and reduced soil damage). This particular technology through indirect mechanisms may provide pathways to improving the safety of the agricultural industry.

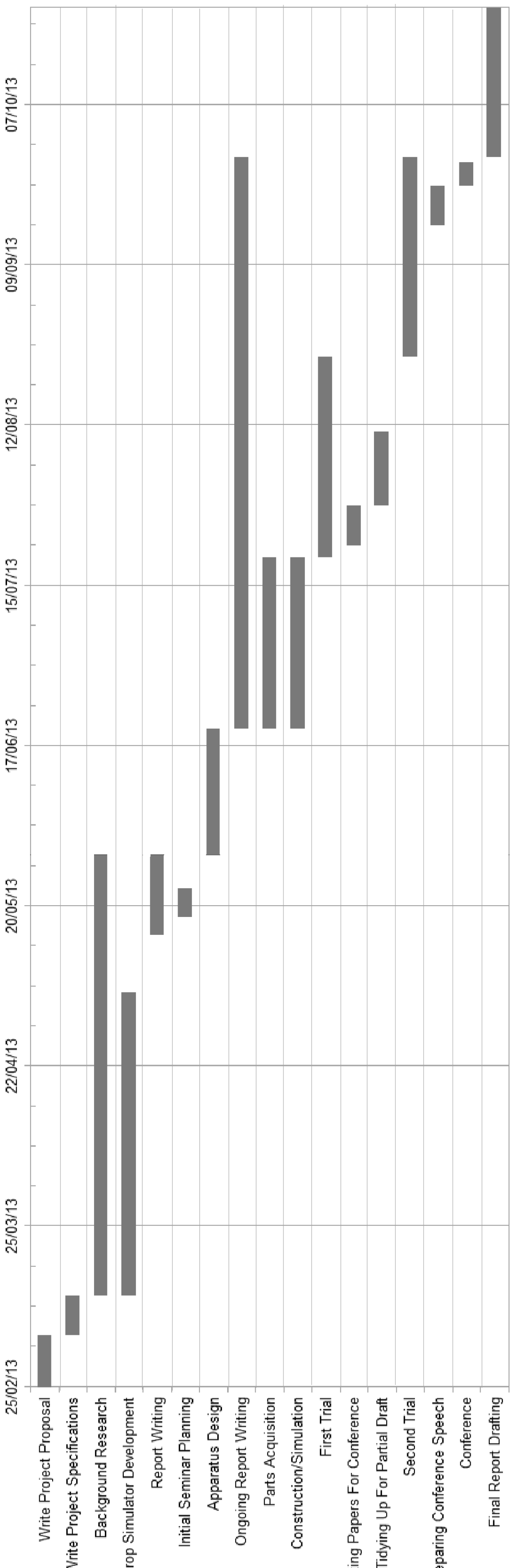
The societal impact is generally positive, while the technology may reduce labor and food costs it may lead to reduced jobs. However through efficient water use many small farms may become viable. This would result in a stimulus to the economy.

These performance increases are maintained with traditional crops, by increasing the efficiency of current strains it may prove possible to limit the introduction of transgenic species. While the author is fervent supporter of genetic engineering many individuals in our society have moral qualms.

Generally as long as the technology remains open and universally accessible. The general impact to society and the environment is overwhelming positive.

Appendix D

Initial Proposed Timeline



Appendix E

Preliminary Resource Analysis

E.1 Faculty Time

The project experiments have been designed in such a way they are almost entirely autonomous. The little required human intervention is more than capable of being fulfilled by the author, Damian Peckett,

As for help with the implementation of the experiments and debugging, faculty support will be somewhat limited due to the lack of expertise in machine learning. However the author is known for resilience and has successfully self taught up until the present time. Help will prove to be available in several machine learning and AI forums to which he is a member of.

As for report suggestions and help with the university relations, the supervisors, Dr. Alison McCarthy and Dr Nigel Hancock have been holding regular weekly meetings with the author. This arrangement has proven highly successful up until this point and the author believes it should remain so. This represents a two to three hour weekly commitment from each of the supervisors.

E.2 University Resources

Beyond access to several square meters of USQ Ag. Plot space, running water, electricity and internet connectivity the experiment is not particularly dependent on university resources. As for software requirements and equipment, the authors design is capable of operating using entirely open source and free software. As for construction of the experimental apparatus, the author has access to a fully equipped personal workshop that should prove ideal.

However this being said, university resources were invaluable during the research phase, with special mention to the USQ library with its well stocked book collection and journal subscriptions.

E.3 Project Budget

As identified previously a maximum worst case gross project cost of approximately 1400 dollars is predicted. This takes into account pricing variability amongst retailers and includes a twenty percent safety margin for unforeseen costs and equipment failures. This value in its gross form is presently higher than ideal, however with some inventive buying and recycling this should prove satisfactory and achievable.

This gross value excludes potential contributions from the university, however keeping this project as primarily self sponsored allows for a greater deal of flexibility and simplifies property ownership. The borrowing of several items from faculty such as computing devices etc, remains an avenue to be investigated.

The budget identifies critical components and has sourced potential suppliers for each, potential avenues for the unavailability of critical components has also been investigated, however at this point little trouble has been identified and action has therefore yet to be taken.

The primarily difficult to source component at the present time, is the biological samples for the emphArabidopsis Thaliana. No Australian retailers have been identified who carry specimens of the widely cultivated Columbia-0 phenotype. The import restrictions on *Arabidopsis* have been researched and it has been identified that international ordering is a viable avenue. The author has asked for a quotation on the importation of two hundred seeds from the American retailer *"Carolina Biological Supply Company."*

In the event that such importation proves unsuccessful, several nearby universities have been shown to work with *Arabidopsis* Contact may be made with them to identify their suppliers or potentially obtain a biological sample.

Appendix F

Preliminary Budget

Project Budget

Version 1.0 Preliminary, Damian Peckett May 2013

System	Item	Suppliers	Price	Uncertainty
Biological Samples	**Arabidopsis Thaliana 200 Seeds**	Carolina.com	$30.00	$20.00
The Pots	48 x 425ml Polypropylene Cups	coles	$6.30	$0.00
	2 x 1kg Aquarium Pebbles	Kmart	$8.00	$2.00
	25L of Potting Mix	coles	$3.00	$0.00
Construction Supplies	Wood Fasteners (Screws etc)	Bunnings	$0.00	$15.00
	Epoxy Glue	Bunnings	$6.95	$0.00
	Silicone Sealant	Bunnings	$5.00	$0.00
	Teflon Plumbing Tape	Ebay	$3.00	$2.00
	Electrical Tape	Bunnings	$6.00	$0.00
	Hookup Wire	Jaycar	$30.00	$0.00
	Heatshrink Tubing	Ebay	$6.00	$4.00
	Permanent Pen	Bunnings	$3.00	$0.00
	Cable Ties (zipties)	Bunnings	$5.00	$0.00
Physical Structure	4 x Kitty Litter Trays	Coles	$27.40	$5.00
	2 x Wardrobe Shelves	Bunnings	$23.00	$0.00
	Divider For Control Pots	None	$0.00	$5.00
	2400 x 1200mm 12mm MDF	Bunnings	$28.00	$5.00
	16M of 90x45mm pine	Bunnings	$45.00	$10.00
	10 Besser Blocks	Bunnings	$32.00	$5.00
Hydraulic System	30L Water Drum	Bunnings	$20.00	$5.00
	Screw in 20mm Tap	Bunnings	$2.00	$1.00
	12V Pump	Ebay	$17.00	$10.00
	30M of 10mm Clear Tubing	Ebay	$28.00	$5.00
	13 x 12v Solenoids, 1/2” Thread	Ebay	$106.60	$10.00
	22 x 1/2” to 10mm Barb Reducers	Ebay	$45.76	$5.00
	11 x 10mm T piece splitters	Ebay	$12.42	$3.00
	36 x Adjustable Drippers	Ebay	$15.00	$4.00
Computing System	**My laptop / Second Hand Donated PC**	None	$0.00	$50.00
	50L Waterproof Storage Container	Bunnings	$20.00	$5.00
	Extension Lead	Bunnings	$0.00	$15.00
	Double Adapter	Bunnings	$0.00	$2.50
Hardware Interface	12V Power Supply ~8A	Ebay	$25.00	$5.00
	Arduino Mega 2560	Ebay	$22.00	$10.00
	2 x 16-Channel 12V Relay Module	Ebay	$42.00	$15.00
	Easycap composite to USB converter	Ebay	$10.00	$15.00
Sensors	**4 x Car Reverse Cameras (waterproof)**	Ebay	$44.00	$5.00
	4 x Clear Boxes For Sensors (waterproof)	Ebay	$25.00	$10.00
	4 x Temp/Humid Sensor Modules	Ebay	$14.00	$3.00
	4 x Light Intensity Modules	Ebay	$12.00	$5.00
	9 x Soil Resistivity Probes For Arduino	Ebay	$40.50	$10.00
Automated Subirrigation	**USB interface postal scales 10kg**	Ebay	$90.00	$20.00
	USB Extension Cord	Ebay	$5.00	$2.00
		TOTAL:	**$863.93**	
		Including Uncertainty:		**$1,152.43**
		20% Safety Margin:		**$1,415.54**

Critical Components Identified in Red.

Appendix G

Risk Analysis, Algorithm Performance Experiment

Risk Management Chart **For Location: USQ Ag Plot** **Hazard Category: Electrical**

Description of Hazards	People at Risk	Number at Risk	Parts of Body	Risk Level
High Voltage Electric Shock Due to 240V Mains Leakage	Persons within Several Metres of Current Leak	2	All	High Consequences, Low Probability

Categories	Short Term Controls	Long Term Controls	Completion Details
Design Substitution Redesign Separation Administration P.P.E.	Low Voltage DC Used in Moisture Prone Areas. All High Voltage Connections To Be Securely Insulated and Kept Away From Human Contact,	* Low Voltage Mains Isolated Supplies * Earth Current Leakage Safety Switches. * Doubly Insulated 240V Connections. * Properly Earthed Power Supply.	

Risk Management Chart **For Location: USQ Ag Plot** **Hazard Category: Electrical**

Description of Hazards	People at Risk	Number at Risk	Parts of Body	Risk Level
Fire Hazard Due To Electrical Malfunction	All in Area	More Than 5.	All	High Consequences, Very Low Probability

Categories	Short Term Controls	Long Term Controls	Completion Details
Design Substitution Redesign Separation Administration P.P.E.	Employ The Use of Fuse's and Current Breakers. Test Apparatus is to be Isolated From Flammable Materials.	* Proper Fusing * Locating Away From Flammable Materials. * Ensure CO2 Fire Extinguishers Available	

Risk Management Chart **For Location: USQ Ag Plot** **Hazard Category: Obstacle**

Description of Hazards	People at Risk	Number at Risk	Parts of Body	Risk Level
Tripping on Exposed Leads	All in Area	1	All	Moderate

Categories	Short Term Controls	Long Term Controls	Completion Details
Design Substitution Redesign Separation Administration P.P.E.	Ensure All Cables Are Well Secured and Out of The Way of Peoples Feet Ensure Cables are Located In Rarely Traversed Areas.	* Warning Sign-age * Buried Cables	

Risk Management Chart For Location: USQ Ag Plot   Hazard Category: Sharp Objects

Description of Hazards	People at Risk	Number at Risk	Parts of Body	Risk Level
Cutting Oneself on Exposed Sharp Surfaces (Corner, Screws, etc.)	Operator	1	All	Low

Categories	Short Term Controls	Long Term Controls	Completion Details
Design Substitution Redesign Separation Administration P.P.E.	Ensure All Sharp Surfaces are Sanded Smooth And No Nails/Screws Are Exposed.	* All Sharp Surfaces are Sanded Smooth * Necessary Sharp Surfaces Shielded With Safety Guarding	

Risk Management Chart For Location: USQ Ag Plot   Hazard Category: Gravity

Description of Hazards	People at Risk	Number at Risk	Parts of Body	Risk Level
Falling Water Reserviour, Seedling Trays, Etc	All in Area	1	All	Low

Categories	Short Term Controls	Long Term Controls	Completion Details
Design Substitution Redesign Separation Administration P.P.E.	Ensure All Trays/Tanks Securely Affixed.	* Build a Fence around the test apparatus	

Risk Management Chart For Location: USQ Ag Plot   Hazard Category: Obstacle

Description of Hazards	People at Risk	Number at Risk	Parts of Body	Risk Level
Tripping On Slippery Surfaces Due to Run Off	All in Area	1	All	Moderate

Categories	Short Term Controls	Long Term Controls	Completion Details
Design Substitution Redesign Separation Administration P.P.E.	Ensure All Run Off Is Drained Into Porous Soil Make Sure Drainage Area Is Inaccessible To Foot traffic.	*Attach To Proper Drainage System/Bed * Fence Around Drainage Area	

Appendix H

Arabidopsis Crop Model

```
/* Arabidopsis Thaliana Crop Model
*  WARNING: MAY OR MAY NOT GENERATE CORRECT RESULTS
*  kind of like schrodingers cat :P
*
* You are welcome to use this however you wish, however I'd love an email
* if you do something cool with it! Word Out! Damo
*
*  Copyright (c) 2013, Damian Peckett <damian.peckett@gmail.com>
*  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* 1. Redistributions of source code must retain the above copyright notice, this
*    list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
*    this list of conditions and the following disclaimer in the documentation
*    and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
* ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define VERBOSELOG

//simulation parameters
#define SIMULATE_DAYS 30
#define SIMULATE_TIMESTEP 5 //seconds
#define WATER_RATE 2 //water application rate in ml/s

#define SATURATETEST

//datasets
int n;
double *precip, *wind, *soiltmp, *airtmp,
```

```
    *humidity, *potentialleaftmp, *flux;

//load matlab formated data files
int load(char fname[], double **values)
{
  int ALLOCBLKSIZ=512, FILEBLOCKSZ=4096;
  int blocks=0, n, posread=0, j, cnt, oneempty=0;
  char *fileblock, *nextblock, combinestr[32];
  float val;
  FILE *data;

  //open file for reading
  data = fopen(fname, "rb");
  if(!data)return -1;

  //initial memory allocation
  *values = NULL;
  fileblock = (char*)malloc(FILEBLOCKSZ*sizeof(char));
  nextblock = (char*)malloc(FILEBLOCKSZ*sizeof(char));

  //initial chunk reads
  fread(fileblock, 1, FILEBLOCKSZ, data);
  fread(nextblock, 1, FILEBLOCKSZ, data);

  //read in loop
  for(n = 0;;++n){
    //allocate memory for the double array
    if(n >= blocks*ALLOCBLKSIZ)
      *values = (double*)realloc(*values,
        (++blocks)*ALLOCBLKSIZ*sizeof(double));

    //skip leading whitespace characters
    for(;;){
      if(posread >= FILEBLOCKSZ){
        memcpy(fileblock, nextblock, FILEBLOCKSZ*sizeof(char));
        cnt=fread(nextblock, 1, FILEBLOCKSZ*sizeof(char), data);
        if(cnt != FILEBLOCKSZ*sizeof(char)){
          if(oneempty == 2){fclose(data); return n;}
          if(cnt < 0)memset(nextblock, ' ', FILEBLOCKSZ*sizeof(char));
          else memset(nextblock+cnt, ' ', (FILEBLOCKSZ*sizeof(char)
            -cnt*sizeof(char)));
          oneempty++;
        }
        posread-=FILEBLOCKSZ;
      }
      if(fileblock[posread] != ' ')break;
      posread++;
    }

    //combine elements from two file chunks
    for(j = 0; combinestr[j-1] != ' '; ++j){
      if((posread+j) >= FILEBLOCKSZ)combinestr[j] = nextblock[(posread+j)
      -FILEBLOCKSZ];
      else combinestr[j] = fileblock[posread+j];
    }
    j--;
    combinestr[j] = 0; posread+=j;

    //read in value
    sscanf(combinestr, "%f", &val);
    (*values)[n] = (double)val;
  }

  return n;
}

int loaddatasets()
{
  int individn;

  n = 10e6;
  if((individn=load("dataset/precip.txt", &precip)) < 0){
```

```
    printf("Error_Reading_precip_Dataset_File!\n");
    return -1;
  }if(individn < n)n = individn;

  if((individn=load("dataset/wind.txt", &wind)) < 0){
    printf("Error_Reading_wind_Dataset_File!\n");
    return -1;
  }if(individn < n)n = individn;

  if((individn=load("dataset/soiltmp.txt", &soiltmp)) < 0){
    printf("Error_Reading_soiltmp_Dataset_File!\n");
    return -1;
  }if(individn < n)n = individn;

  if((individn=load("dataset/airtmp.txt", &airtmp)) < 0){
    printf("Error_Reading_airtmp_Dataset_File!\n");
    return -1;
  }if(individn < n)n = individn;

  if((individn=load("dataset/rh.txt", &humidity)) < 0){
    printf("Error_Reading_humidity_Dataset_File!\n");
    return -1;
  }if(individn < n)n = individn;

  if((individn=load("dataset/potentialleaftmp.txt", &potentialleaftmp)) < 0){
    printf("Error_Reading_potentialleaftmp_Dataset_File!\n");
    return -1;
  }if(individn < n)n = individn;

  if((individn=load("dataset/flux.txt", &flux)) < 0){
    printf("Error_Reading_flux_Dataset_File!\n");
    return -1;
  }if(individn < n)n = individn;

  return 0;
}

//The Actual Simulator
void simulationloop(int STARTINGDAYINYEAR)
{
  //constants
  /* soil resistance, from theta = 0.12 to theta = 0.25, for peat/perlite 50/50 mix
    (Haofang Yan, Chuan Zhang, Hiroki Oue, Hideki Sugimoto 2012)
   */
  double retable[] = {
    71.4000,60.1800,53.0400,46.9200,41.8200,37.7400,34.1700,31.1100,27.5400,25.5000,
    23.4600,21.9300,20.9100,19.8900,17.8500,16.3200,14.7900,13.7700,12.7500,12.2400,
    11.2200,10.7100,9.6900,8.6700,8.6700,8.6700,8.1600,6.6300,5.6100,5.6100};

  //critical volumetric water ratio
  //http://www.treemail.nl/download/treebook7/soil/chapt6.htm#eq6_4
  //moderately drought sensitive crop
  double ET0COEFF[] = {
    1.0, 0.75,0.65,0.55,0.45,0.40,0.38,0.33,0.30,0.25
  }; //ET0 = 0.0-1.0cm.d^-1


  //Rate of photosynthesis vs Photosynthetic Flux Density
  //Growth and photosynthesis under high and low irradiance of Arabidopsis
  //thaliana  Eckardt et al., 1997 Derived from figure 4
  double photovsPPFD[] = {
    0.00, 0.16, 0.33, 0.46, 0.56, 0.66, 0.72, 0.78, 0.83, 0.86, 0.88, 0.92,
    0.95, 0.97, 1.00}; //0-700 umol*m^-2*s^-1
  //The efficiency of the photosynthetic pathway at converting exchanged
  //carbon into biomass Tuned empirically
  double photosynthetic_efficiency = 0.83;

  //(The ELF4ELF3LUX complex links the circadian clock to diurnal control of
  //hypocotyl growth)
  double stochasticmagic = 3.5; //scaling factor, chosen empirically so that
                                //the stochastic application converges
  double growthvshoursincesunrise[] = {
```

```
    0.0807,0.0719,0.0590,0.0462,0.0386,0.0339,0.0316,0.0298,0.0292,0.0292,
    0.0292,0.0292,0.0292,0.0292,0.0292,0.0292,0.0292,0.0304,0.0327,0.0357,
    0.0432,0.0549,0.0701,0.0783};

//Growing Pot Constraints
double pot_volume = 425; //in ml
double pot_surface_area = 44; //in cm^2

/* van genutchen modelled values, for 50/50 peat perlite mixture
     Simultaneous Determination of Water Retention Curve and Unsaturated
   Hydraulic Conductivity of Substrates Using a Steady-state Laboratory Method
   ???(Haofang Yan, Chuan Zhang, Hiroki Oue, Hideki Sugimoto 2012)
 */
double theta_fc = 0.248;
double theta_wilt = 0.120;
double theta_critical = 0.216;
double theta_sat = 0.650;

//Variables

//internal math
double gva, gv;
double esatleaf, esatsoil, eair, molswaterpersecperm2;
double potential_ml_lost_transpiration, wateravailability, transpirationloss;
double Pn, PAR, PPFD, photosynthetic_rate, photopercent, photodiff, photointerp;
int PPFDidx, PPFDidxb;
double dmarea, carbon_exchanged;
double drymassgain, wet_biomass_gain_in_g, ml_lost;
double lastsunrise=-29600; //dodgy first pass parameter to get things going
double sincesunrise, leafareadeficit;
int rateidx, reidx;
double soilresistance, boundaryresistance, latentflux, latentjoules;
double mlrainfall;

double ET0,megajoulesflux,esatair; //cm.d^-1
double Pdiff, Pinterp, P, beta;
int Rnindex, Pidx, Pidxb;

//starting condition, pot at field capacity
double current_volumetric_ratio = theta_fc;

/* dry biomass in grams, 10mg at the end of the germination phase (rough 7
 * days), sure that might sound small but (GLUTATHIONE DYNAMICS IN ARABIDOPSIS
 * SEED DEVELOPMENT AND GERMINATION) shows the arabidopsis seed has a dry
 * mass of roughly 20ug and that includes the seed coat, 10mg is probably
 * on the order of 500x the stored biomass!!
 */
double current_shoot_biomass = 0.01;//dry biomass in grams
//100mm^2, or the usual area after 8 days,
//great initial condition and approximation for the germination phase
double current_leaf_area = 100e-6;

double simutime;
int hour_idx;

int lastidx = 0, last_record = 0;
FILE *logf;
double lastrecord = -1e6;


#ifdef VERBOSELOG
    logf = fopen("log.csv", "wb");
    if(!logf){puts("Error Opening Logging File!");return;}
#endif

//Main loop
for(simutime = 0; simutime <= (SIMULATE_DAYS*24*3600); simutime+=SIMULATE_TIMESTEP)
{
        //which index in the dataset
    hour_idx = (int)(((STARTING_DAY_IN_YEAR-1)*24) + floor(simutime/3600));
    if(hour_idx >= n)hour_idx-=n;

#ifdef SATURATETEST
```

```
    //saturation, conditions found in study:
    //Maximal Biomass of Arabidopsis fha/iana Using a Simple, Low-Maintenance
    //Hydroponic Method and Favorable Environmental Conditions
    if(hour_idx%24 <= 10){ //10 hour photoperiod
      flux[hour_idx] = 202; //about 400 umol*m^-2*s^-1
    }else flux[hour_idx] = 0;
    precip[hour_idx] = 0;
    humidity[hour_idx] = 75;
    wind[hour_idx] = 1;
    soiltmp[hour_idx] = 293; //20c
    airtmp[hour_idx] = 293; //20c
    potentialleaftmp[hour_idx] = 290; //21c ... bit of estimate
    current_volumetric_ratio = theta_fc; //continuous field capacity
#endif

    //Transpiration
    //all from environmental biophysics chap 4.1
    //boundary layer resistances
    //convection formulas break down at low wind speed, (1962 paper)
    if(wind[hour_idx] < 2.0)gva = 1.4*0.147*sqrt(2/0.0086);
    else gva = 1.4*0.147*sqrt(wind[hour_idx]/0.0086);
    //stomata conductivities for arabidopsis
    gv = (0.5*0.24*gva)/(0.24+gva) + (0.5*0.17*gva)/(0.17+gva); //mol*m^-2*s^-1

    //Saturated vapor pressure at leaf temperature (Murray, 1967),
    //pressure in kpa, T in kelvin
    esatleaf = 0.61078*exp(17.269388*((potentialleaftmp[hour_idx]-273.16)
      /(potentialleaftmp[hour_idx]-35.86)));
    eair = (humidity[hour_idx]/100)*0.61078*exp(17.269388*((airtmp[hour_idx]-273.16)
      /(airtmp[hour_idx]-35.86)));
    molswaterpersecperm2 = gv*((esatleaf-eair)/100);

    //How many mls of water does this equate to
    potential_ml_lost_transpiration = molswaterpersecperm2*18.0152
      *SIMULATE_TIMESTEP*current_leaf_area;
    //now this is assuming infinite water supply, of course this is wrong,

    //Difficult to source something direct but
    //Drought stress inhibits photosynthesis by decreasing stomatal a
    //perturenot by affecting ATP synthesis Interspecies differences in
    //photosynthetic gas exchange characteristics and acclimation to soil
    //moisture stress... So stomata conductivity is proportional to
    //photosynthetic rate, so we need stomatol conductivity vs water
    //Effect of soil moisture on canopy conductance of Amazonian rainforest
    //A canopy conductance and photosynthesis model for use in a GCM land
    //surface scheme http://www.fao.org/docrep/x0490e/x0490e08.htm
    //very rough penmann approximation
    esatair = 0.61078*exp(17.269388*((airtmp[hour_idx]-273.16)/(airtmp[hour_idx]-35.86)));
    for(Rnindex = hour_idx, megajoulesflux = 0; Rnindex < hour_idx+24; ++Rnindex)
      megajoulesflux+= 0.0036*flux[(Rnindex >= n)?Rnindex-n:Rnindex];
    if(wind[hour_idx] < 2)ET0 = (0.006*megajoulesflux+(4.5/airtmp[hour_idx])
      *(esatair-eair))/0.217; //cm*d^-1 low windspeed approximation
    else ET0 = (0.006*megajoulesflux+(4.5/airtmp[hour_idx])*wind[hour_idx]
      *(esatair-eair))/(0.20 + 0.017*wind[hour_idx]); //cm*d^-1
    if(ET0 > 0.9)P = 0.25;
    else{
      Pidx = (int)(floor(ET0*(sizeof(ET0COEFF)/sizeof(double))));
      Pidxb = (int)(ceil(ET0*(sizeof(ET0COEFF)/sizeof(double))));
      Pdiff = ET0COEFF[Pidxb]-ET0COEFF[Pidx];
      Pinterp = ET0*(sizeof(ET0COEFF)/sizeof(double)) - (double)Pidx;
      P = ET0COEFF[Pidx] + Pinterp*Pdiff;
    }
    //our actual value for theta critical
    //http://www.treemail.nl/download/treebook7/soil/chapt6.htm
    theta_critical = (1-P)*(theta_fc-theta_wilt)+theta_wilt;
    //Now we model the water availability using ... A canopy conductance
    //and photosynthesis model for use in a GCM land surface scheme
    //'Jarvis' Model, Just used some fairly generic tuning parameters
    if(current_volumetric_ratio >= theta_critical)beta = 1;
```

```
else if(current_volumetric_ratio > theta_wilt && current_volumetric_ratio
  < theta_critical)beta = (current_volumetric_ratio-theta_wilt)
  /(theta_critical-theta_wilt);
else beta = 0;
wateravailability = (1-exp(0.3*beta))/(1-exp(0.3));
//Thornthwaite (1955), claims transpiration varies linearly with soil
//water availability, 'Jarvis' model is surprisingly consistent!
//How much water is predicted that we did lose?
transpirationloss = potential_ml_lost_transpiration*wateravailability;
current_volumetric_ratio = current_volumetric_ratio
  - (transpirationloss/pot_volume); //update our counter


//Propagation
//Impact of Elevated CO2 on Growth and Development of Arabidopsis
//thaliana L. http://www.landesmuseum.at/pdf_frei_remote/PHY_36_2_0173-0184.pdf
//(SASAKI, H., M. FUKUYAMA AND T. ONOUE 2001)
Pn = -0.031*((potentialleaftmp[hour_idx]-273.15)*
(potentialleaftmp[hour_idx]-273.15))
+ 1.897*(potentialleaftmp[hour_idx]-273.15) - 5.229;
Pn *=0.84; //tuned to match Eckardt et al., 1997
//(M. Tsubo and S. Walker 2005), fresno california
PAR = 0.45*flux[hour_idx]; //photosynthetically active radiation
//(Forest ecosystems and environments p 336, Takashi. Kohyama, Josep.
//Canadell, Dennis S. Ojima, Louis F. Pitelka)
PPFD = PAR*4.4; //bit of an approximation but close!
if(PPFD >= 650){
  photopercent = 1.0;
}else{ //linear interpolation
  PPFDidx = (int)(floor((PPFD/700)*(sizeof(photovsPPFD)/sizeof(double))));
  PPFDidxb = (int)(ceil((PPFD/700)*(sizeof(photovsPPFD)/sizeof(double))));
  photodiff = photovsPPFD[PPFDidxb]-photovsPPFD[PPFDidx];
  photointerp = ((PPFD/700)*(sizeof(photovsPPFD)/sizeof(double))) - (double)PPFDidx;
  photopercent = photovsPPFD[PPFDidx] + photointerp*photodiff;
}

//earlier we established the stomatol response to water limitation is somewhat linear
//as stomata conductivity limits carbon dioxide exchange, we will assume the effect
//on pohtosynthetic rate is therefore also linear
//Thornthwaite (1955), claims transpiration varies linearly with soil water availability
photosynthetic_rate = photopercent*Pn*wateravailability; //in umol(CO2)*m^-2*s^-1
//completely saturated photosynthetic pathway
if(photosynthetic_rate > 15*photosynthetic_efficiency)photosynthetic_rate
  = 15*photosynthetic_efficiency;

//Net Carbon Exchange Rates of Field-grown Crops in Relation to
//Irradiance and Dry Weight Accumulation, based on caulifower/cabbage
//values
if(PAR == 0){//nighttime, carbon efflux, 44000mg a mole of co2
  dmarea = 100*current_leaf_area;
  if(potentialleaftmp[hour_idx] < 5)carbon_exchanged = 0; //dormant
  //in moles, at higher temperatures, the plant is more metabolically active
  else carbon_exchanged = ((-0.4*(potentialleaftmp[hour_idx]-278.15))
    *dmarea)/(44000*(3600/SIMULATE_TIMESTEP));
}else carbon_exchanged = 0.000001*photosynthetic_rate*SIMULATE_TIMESTEP
  *current_leaf_area; //in moles

//now how much biomass did we accumulate, Growth and carbon economy of
//a fastgrowing and a slow-growing grass species as dependent on ontogeny 2001
//very rough to apply cross species
//from graphs, carbon content is a pretty consistent 35 mmol*g^-1
//(dry weight), just for giggles that is 42% of the total dry mass
drymassgain = (carbon_exchanged*1000)/35; //Growth and carbon economy of
//a fastgrowing and a slow-growing grass species as dependent on ontogeny 2001
wet_biomass_gain_in_g = drymassgain/0.08; //arabidopsis dry matter ~8% of
//total, Browse and Somerville (1994) Chap 32 of Arabidopsis, Cold Spring
//Harbor Laboratory Press
//WATER USED BY PHOTOSYNTHESIS, OR RELEASED AS PART OF METABOLISM
ml_lost = wet_biomass_gain_in_g-drymassgain;
current_volumetric_ratio = current_volumetric_ratio - (ml_lost/pot_volume);
```

```
    //update our counter
    //Roughly 66% of the biomass exists as leaf tissue, (An aeroponic culture
    //system for the study of root herbivory on Arabidopsis thaliana 2011)
    //Growth of Arabidopsis thaliana seedlings under water deficit studied
    //by control of water potential in nutrient-agar media
    current_shoot_biomass = current_shoot_biomass + 0.66*drymassgain;

#ifdef SATURATETEST
    //See Below For Justification
    current_leaf_area=20*(current_shoot_biomass/1000);
#else
    //now lets convert our biomass gain to a corresponding leaf area gain,
    //very rough and approximate step!!! Response of mannitol-producing
    //Arabidopsis thaliana to abiotic stress 2007 from the control graphs
    //we see ~10% dry weight ratio and we see a leaf area to mass ratio of
    //approximately 15m^2kg^-1 (dry weight), this figure is identical to
    //(Growth and carbon economy of a fastgrowing and a slow-growing grass
    //species as dependent on ontogeny 2001)
    //very interesting observation.
    //Further observations of the study Maximal Biomass of Arabidopsis
    //fha/iana Using a Simple, Low-Maintenance Hydroponic Method and
    //Favorable Environmental Conditions
    //Indicates approximately 25m^2kg^-1 (dry weight)
    //I think a compromise can be made at roughly ~20m^2kg^-1
    //growth rate varies with the time in the dirunal cycle, we will
    //model for this (The ELF4ELF3LUX complex links the circadian clock
    //to diurnal control of hypocotyl growth) day length calculations
    if(hour_idx > 0 && flux[hour_idx] > 0 && flux[hour_idx-1] == 0)
      lastsunrise = simutime; //sunrise
    sincesunrise = simutime-lastsunrise;
    if(sincesunrise < 0)sincesunrise+=(24*3600);
    rateidx = (int)floor(sincesunrise/3600 + 0.5);
    if(rateidx == 24)rateidx = 0;

    leafareadeficit = 20*(current_shoot_biomass/1000) - current_leaf_area;
    current_leaf_area = current_leaf_area + leafareadeficit*((stochasticmagic
      *growthvshoursincesunrise[rateidx])/(3600/SIMULATE_TIMESTEP));
#endif

    //Evaporation
    reidx = floor(((current_volumetric_ratio-theta_wilt)/(theta_fc-theta_wilt))
      *(sizeof(retable)/sizeof(double)) + 0.5);
    if(reidx < 0)reidx = 0;//simple bounds checking
    else if(reidx >= (sizeof(retable)/sizeof(double)))reidx =
      (sizeof(retable)/sizeof(double));
    soilresistance = retable[reidx];
    //(Haofang Yan, Chuan Zhang, Hiroki Oue, Hideki Sugimoto 2012) eq 6,
    //wind at 2m
    if(wind[hour_idx] < 1){ //for low wind speeds, natural convection becomes
                //dominant, very rough approximation
                //boundary layer resistance
      boundaryresistance = (1/(0.1681))*log(2/0.001)*log(2/0.001);
    }else boundaryresistance = //boundary layer resistance
      (1/(0.1681*wind[hour_idx]))*log(2/0.001)*log(2/0.001);

    //Saturated vapor pressure (Murray, 1967), pressure in kpa, T in kelvin
    esatsoil = 0.61078*exp(17.269388*((soiltmp[hour_idx]-273.16)
      /(soiltmp[hour_idx]-35.86)));
    //(Haofang Yan, Chuan Zhang, Hiroki Oue, Hideki Sugimoto 2012) eq 5,
    //approximated for sea level ~20c
    latentflux = (1005*1.2*(esatsoil-eair))/(0.0665*(boundaryresistance
      +soilresistance)); //in watts per m^2 of surface
    latentjoules = latentflux*(pot_surface_area*0.0001)*SIMULATE_TIMESTEP;
    //YES SOMETIMES THIS WILL BE NEGATIVE, MORNING DEW :D
    ml_lost = (latentjoules/2.45e3); //using latent heat of vaporisation
    //approximated to 2.45MJkg^-1
    current_volumetric_ratio = current_volumetric_ratio - (ml_lost/pot_volume);

    //Precipitation
    mlrainfall = (precip[hour_idx]*(pot_surface_area/10))/(3600/SIMULATE_TIMESTEP);
```

```
    current_volumetric_ratio = current_volumetric_ratio + (mlrainfall/pot_volume);


    //Irrigation
    //Basic Bang-Bang irrigation control strategy
    if(current_volumetric_ratio < (theta_critical+theta_wilt)/2){
      printf("Applying Water\n");
      lastidx = hour_idx;
      current_volumetric_ratio = theta_fc;
    }

    //Normalize/Sanitize Values
    if(current_volumetric_ratio < theta_wilt){
      puts("Water loss will not continue at the current rate!");
      puts("soil resistance will become very dominant!");
      puts("Poorly Defined Region Of Model!");
    }
    if(current_volumetric_ratio < 0.75*theta_wilt)current_volumetric_ratio =
      0.75*theta_wilt; //sanitize value
    else if(current_volumetric_ratio > theta_fc)current_volumetric_ratio =
      theta_fc; //water lost to drainage

    //Record Interim Values If We Want...
    #ifdef VERBOSELOG
      if((lastrecord + 15*60) <= simutime){ //Every 15 mins in sim time
        lastrecord = simutime;
        if(simutime == 0)fprintf(logf, "\"Volumetric Water Ratio\",\"Leaf Area\","
          "\"Photosynthetic Rate (umol*m^-2*s^-1)\",\"Transpiration Rate\"\n");
        fprintf(logf, "%g,%g,%g,%g\n", current_volumetric_ratio,
          current_leaf_area*1000000, ((carbon_exchanged*1000000)/
          (current_leaf_area*SIMULATE_TIMESTEP)), transpirationloss*
          (3600/SIMULATE_TIMESTEP));
      }
    #endif
  }

  fclose(logf);

  printf("Modelled Final Values:\nDry Shoot Biomass: %.3g grams\nLeaf Area: %d mm^2\n",
    current_shoot_biomass, (int)(current_leaf_area*1000000));
}

int main()
{
  //Load Datasets
  if(loaddatasets() < 0)
    return -1;

  //Run A Simulation Starting On Day 90 Of The Year
  simulationloop(1);

  system("pause");

  return 0;
}
```

Appendix I

Leaf Temperature Model

```
% You are welcome to use this however you wish, however I'd love an email
% if you do something cool with it! Word Out! Damo
%
%  Copyright (c) 2013, Damian Peckett <damian.peckett@gmail.com>
%  All rights reserved.
%
% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are met:
%
% 1. Redistributions of source code must retain the above copyright notice, this
%    list of conditions and the following disclaimer.
% 2. Redistributions in binary form must reproduce the above copyright notice,
%    this list of conditions and the following disclaimer in the documentation
%    and/or other materials provided with the distribution.
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
% ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
% WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
% DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
% ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
% (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
% LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
% ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
% (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
% SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

%written from chap 4.1 in "Environmental Biophysics"

airtmp = load('processed/airtmp.txt');
soiltmp = load('processed/soiltmp.txt');
humid = load('processed/rh.txt');
flux = load('processed/flux.txt');
wind = load('processed/wind.txt');

tablea3 = [ ...
  268.2,-5,0.422,32,293,0.149,0.66; ...
  269.2,-4,0.455,34,298,0.151,0.67; ...
  270.2,-3,0.490,36,302,0.153,0.67; ...
  271.2,-7,0.528,39,307,0.154,0.68; ...
  272.2,-1,0.568,42,311,0.156,0.68; ...
  273.2,0,0.611,44,316,0.158,0.69; ...
  274.2,1,0.657,47,320,0.160,0.69; ...
  275.2,2,0.706,50,325,0.161,0.70; ...
  276.2,3,0.758,54,330,0.163,0.70; ...
  277.2,4,0.813,57,335,0.165,0.71; ...
  278.2,5,0.872,61,339,0.167,0.71; ...
  279.2,6,0.935,65,344,0.168,0.72; ...
```

```
  280.2,7,1.001,69,349,0.170,0.72; ...
  281.2,8,1.072,73,354,0.172,0.73; ...
  282.2,9,1.147,77,359,0.174,0.73; ...
  283.2,10,1.227,82,365,0.176,0.74; ...
  284.2,11,1.312,87,370,0.178,0.74; ...
  285.2,12,1.402,92,375,0.179,0.75; ...
  286.2,13,1.497,98,380,0.181,0.75; ...
  287.2,14,1.597,104,386,0.183,0.76; ...
  288.2,15,1.704,110,391,0.185,0.76; ...
  289.2,16,1.817,116,396,0.187,0.77; ...
  290.2,17,1.936,123,402,0.189,0.77; ...
  291.2,18,2.062,130,407,0.191,0.78; ...
  292.2,19,2.196,137,413,0.193,0.79; ...
  293.2,20,2.336,145,419,0.195,0.79; ...
  294.2,21,2.485,153,425,0.197,0.80; ...
  295.2,22,2.642,161,430,0.199,0.80; ...
  296.2,23,2.808,170,436,0.201,0.81; ...
  297.2,24,2.982,179,442,0.203,0.81; ...
  298.2,25,3.166,189,448,0.205,0.82; ...
  299.2,26,3.360,199,454,0.207,0.82; ...
  300.2,27,3.564,209,460,0.209,0.83; ...
  301.2,28,3.778,220,466,0.211,0.83; ...
  302.2,29,4.004,232,473,0.214,0.84; ...
  303.2,30,4.242,244,479,0.216,0.85; ...
  304.2,31,4.492,256,485,0.218,0.85; ...
  305.2,32,4.754,269,492,0.220,0.86; ...
  306.2,33,5.030,283,498,0.222,0.86; ...
  307.2,34,5.320,297,505,0.224,0.87; ...
  308.2,35,5.624,311,511,0.227,0.87; ...
  309.2,36,5.943,327,518,0.229,0.88; ...
  310.2,37,6.278,343,525,0.231,0.89; ...
  311.2,38,6.629,359,532,0.233,0.89; ...
  312.2,39,6.996,376,538,0.235,0.90; ...
  313.2,40,7.382,394,545,0.238,0.90; ...
  314.2,41,7.785,413,552,0.240,0.91; ...
  315.2,42,8.208,432,559,0.242,0.91; ...
  316.2,43,8.650,452,567,0.245,0.92; ...
  317.2,44,9.113,473,574,0.247,0.93; ...
  318.2,45,9.597,495,581,0.249,0.93; ...
];

%implementing (Gaylon S. Campbell & John M. Norman)

leaftmp = [0];

for i=1:8736
  %convection formulas break down at low wind speed, (1962 paper)
  if wind(i) < 2
    gHa = 1.4*0.135*sqrt(2/0.0086);
    gva = 1.4*0.147*sqrt(2/0.0086);
  else
    gHa = 1.4*0.135*sqrt(wind(i)/0.0086);
    gva = 1.4*0.147*sqrt(wind(i)/0.0086);
  end

  idx = round((airtmp(i)-273)+6);
  if idx < 1
    idx = 1;
  elseif idx > length(tablea3)
    idx=length(tablea3);
  end

  gr = tablea3(idx,6);
  gHr = gHa + gr;

  %stomata conductivities for arabidopsis, proud of that!!!
  gv = (0.5*0.24*gva)/(0.24+gva) + (0.5*0.17*gva)/(0.17+gva);

  ystar = ((6.66e-4)*gHr)/gv;

  s = tablea3(idx,4)/100000;
  D = tablea3(idx,3) - (humid(i)/100)*tablea3(idx,3);

  Rabs = 0.55*flux(i);
```

```
  Remitt = 0.97*tablea3(idx,5);
  Rni = Rabs - Remitt;

  delta = (ystar/(ystar+s))*(Rni/(gHr*29.3) - D/(100*ystar));
  if flux(i) < 100 %nightitme sanitization, empiraclly calculated by damian peckett
    if flux(i) > 0
      %MOVING AVERAGE SMOOTHING
      leaftmp(i) = (leaftmp(i-1) + airtmp(i) + 0.4*delta)/2;
    else
      leaftmp(i) = airtmp(i) + 0.4*delta;
    end


  else
    leaftmp(i) = airtmp(i) + delta;
  end
end

save('leaftmp.txt', 'leaftmp','-ascii');
```

Appendix J

Soil Temperature Model

```
% You are welcome to use this however you wish, however I'd love an email
% if you do something cool with it! Word Out! Damo
%
%  Copyright (c) 2013, Damian Peckett <damian.peckett@gmail.com>
%  All rights reserved.
%
% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are met:
%
% 1. Redistributions of source code must retain the above copyright notice, this
%    list of conditions and the following disclaimer.
% 2. Redistributions in binary form must reproduce the above copyright notice,
%    this list of conditions and the following disclaimer in the documentation
%    and/or other materials provided with the distribution.
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
% ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
% WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
% DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
% ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
% (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
% LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
% ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
% (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
% SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

% CSIRO SHALLOW SOIL MODEL

%Fresno California
latitude = 36.75;
longitude = 240.25;

minmaxtmp = load('soiltemp/minmax.txt');

plottmp = [0];
elements = 0;

for dayoftheyear=1:364
  %convert the longitude to hour value and calculate an approximate time for
  %sunrise/sunset
  lngHour = longitude / 15;
  trise = dayoftheyear + ((6 - lngHour) / 24);
  tset = dayoftheyear + ((18 - lngHour) / 24);

  %calculate the Sun's true longitude
  Lrise = (0.9856*trise) - 3.289 + (1.916 * sin(degtorad(((0.9856*trise) -
  3.289)))) + (0.020 * sin(2*degtorad(((0.9856*trise) - 3.289)))) + 282.634;
```

```
if Lrise > 360
  Lrise=Lrise-360;
elseif Lrise < 0
  Lrise=Lrise+360;
end

Lset = (0.9856*tset) - 3.289  + (1.916 * sin(degtorad(((0.9856*tset) -
3.289)))) + (0.020 * sin(2*degtorad(((0.9856*tset) - 3.289)))) + 282.634;
if Lset > 360
  Lset=Lset-360;
elseif Lset < 0
  Lset=Lset+360;
end

%calculate the Sun's right ascension
RArise = (180/3.14159)*atan(0.91764 * tan(degtorad(Lrise)));
RAset = (180/3.14159)*atan(0.91764 * tan(degtorad(Lset)));

%sanitize quadrant
Lquadrant  = (floor(Lrise/90)) * 90;
RAquadrant = (floor(RArise/90)) * 90;
RArise = (RArise + (Lquadrant - RAquadrant))/15;
Lquadrant  = (floor(Lset/90)) * 90;
RAquadrant = (floor(RAset/90)) * 90;
RAset = (RAset + (Lquadrant - RAquadrant))/15;

%lots of maths, but turn the angle of sunset/rise into minutes
Hrise = ((360 - (180/3.14159)*acos((-((0.39782*sin(degtorad(Lrise)))
*sin(degtorad(latitude))))/(cos(asin(0.39782*sin(degtorad(Lrise))))
*cos(degtorad(latitude)))))/15);
Hset = ((180/3.14159)*acos((-((0.39782*sin(degtorad(Lset)))
*sin(degtorad(latitude))))/(cos(asin(0.39782*sin(degtorad(Lset))))
*cos(degtorad(latitude))))/15);

%convert to local time at our longitude
Hrise = (Hrise+RArise-(0.06571*trise)-6.622)*60;
Hset = (Hset+RAset-(0.06571*tset)-6.622)*60;

if Hrise < 0
  Hrise=Hrise+1440;
elseif Hrise > 1440
  Hrise=Hrise-1440;
end
if Hset < 0
  Hset=Hset+1440;
elseif Hset > 1440
  Hset=Hset-1440;
end

%variables
airtempmin = minmaxtmp(dayoftheyear*2-1);
airtempmax = minmaxtmp(dayoftheyear*2);
if dayoftheyear == 1
  airtempminlastday = airtempmin;
  airtempmaxlastday = airtempmax;
else
  airtempminlastday = minmaxtmp(dayoftheyear*2-3);
  airtempmaxlastday = minmaxtmp(dayoftheyear*2-2);
end
if dayoftheyear == 364
  airtempminnextday = airtempmin;
else
  airtempminnextday = minmaxtmp(dayoftheyear*2+1);
end

for minutes_since_midnight=0:60:1380
  % Algorithm implemented from:
  % Models for estimation of hourly soil temperature at 5 cm depth and for
  % degree-day accumulation from minimum and maximum soil temperature.
  % (B. Horton 2012, CSIRO Publishing)
  % Damian Peckett, April 2013
```

```
    Hmid = (Hset+Hrise)/2;
    Hmin = Hrise + 0.19*(Hmid-Hrise) + 28.5; %Time of minimum soil temperature
    Hmax = Hmid + 0.097*(Hset-Hmid) + 118.5; %Time of maximum soil temperature

    %transistion point Hx is roughly 1 hour after sunset
    Hx = Hset+60;
    %Calculated using data from the BOM and G. W. Leeper
    Dmax = airtempmax-0.3*(airtempmax-airtempmin);
    Dmin = airtempmin+0.1*(airtempmax-airtempmin);
    Dx = 0.5*(Dmax+Dmin); %~1 hour after sunset (B. Horton)
    Dmaxyester = airtempmaxlastday-0.3*(airtempmaxlastday-airtempminlastday);
    Dminyester = airtempminlastday+0.1*(airtempmaxlastday-airtempminlastday);
    Dx_yesterday = 0.5*(Dmaxyester+Dminyester); %~1 hour after sunset (B. Horton)
    Dmintom = airtempminnextday+0.1*(airtempmax-airtempminnextday);

    timeplot = minutes_since_midnight;
    if timeplot >= 1440
      timeplot = timeplot - floor(timeplot/1440)*1440; %sanitization within day
    end

    %timeplot < Hmin (decay function eq 5a)
    %timeplot > Hmin and timeplot < Hmax (triple sin function eq 6)
    %timeplot > Hmax and timeplot < Hx (sin decay eq 4)
    %timeplot > Hx (decay function eq 5a)
    if timeplot < Hmin %eq 5a
      D = Dx_yesterday + (Dx_yesterday-Dmin)*sin(3.14159+0.5*3.14159
      *((timeplot+(1440-Hx))/(Hmin+(1440-Hx))));
    elseif timeplot > Hmin && timeplot < Hmax %eq 6
      D = (Dmin+Dmax)/2 + 0.5*(Dmax-Dmin)*sin(3.14159*((timeplot-0.5
      *(Hmin+Hmax))/(Hmax-Hmin)));
    elseif timeplot > Hmax && timeplot < Hx %eq 4
      D = Dx + (Dmax-Dx)*sin(0.5*3.14159 + 0.5*3.14159*((timeplot-Hmax)
      /(Hx-Hmax)));
    elseif timeplot > Hx %eq 5a
      D = Dx + (Dx-Dmintom)*sin(3.14159+0.5*3.14159*((timeplot-Hx)
      /((Hmin+1440)-Hx)));
    end

    %our results
    %D

    elements = elements + 1;
    plottmp(elements)=D;
  end

  if dayoftheyear == 365 %complete the dataset
    timeplot = 1439;
    if timeplot >= 1440
      %sanitization within day
      timeplot = timeplot - floor(timeplot/1440)*1440;
    end
    if timeplot < Hmin %eq 5a
      D = Dx_yesterday + (Dx_yesterday-Dmin)*sin(3.14159+0.5*3.14159
      *((timeplot+(1440-Hx))/(Hmin+(1440-Hx))));
    elseif timeplot > Hmin && timeplot < Hmax %eq 6
      D = (Dmin+Dmax)/2 + 0.5*(Dmax-Dmin)*sin(3.14159*((timeplot
      -0.5*(Hmin+Hmax))/(Hmax-Hmin)));
    elseif timeplot > Hmax && timeplot < Hx %eq 4
      D = Dx + (Dmax-Dx)*sin(0.5*3.14159 + 0.5*3.14159*((timeplot-Hmax)
      /(Hx-Hmax)));
    elseif timeplot > Hx %eq 5a
      D = Dx + (Dx-Dmintom)*sin(3.14159+0.5*3.14159*((timeplot-Hx)
      /((Hmin+1440)-Hx)));
    end
    elements = elements + 1;
    plottmp(elements)=D;
  end
end

meas = plottmp(1:(length(plottmp)-8));
```

```
meas = [meas(17:24),meas];
meas = meas+273.15;

save('soiltmp.txt', 'meas','-ascii');
```

Appendix K

Massively Parallel MDP Solver

```
/* Pretty darn fast, portable, MDP solver, Value iteration  with RBF Network
*   Value Estimation, WARNING: MAY OR MAY NOT GENERATE CORRECT RESULTS
*   kind of like schrodingers cat :P
*
* You are welcome to use this however you wish, however I'd love an email
* if you do something cool with it! Word Out! Damo
*
*   Copyright (c) 2013, Damian Peckett <damian.peckett@gmail.com>
*   All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* 1. Redistributions of source code must retain the above copyright notice, this
*    list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
*    this list of conditions and the following disclaimer in the documentation
*    and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
* ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include <math.h>
#include <time.h>

//#define WINDOWS

#ifdef WINDOWS
// Dodgy implementation of pthreads for Windows
//http://locklessinc.com/
//makes linux porting way easier
#include <windows.h>
#include "pthreads.h"
#endif

#define TRAINSAMPLES 2000
```

```
#define HOLDOUTSAMPLES 50
#define DISCOUNT 0.98 //common empirical value

#define Rn 14 //dimensions

//from decompose centroids matlab script
#define NRBF 8
double RBFCentroids[NRBF*Rn] = {
296,40,369,0.158,6265,1.8,296,41,1.8,296,42,1.6,27,0.621,
298,38,374,0.168,21144,2.1,299,36,2.5,298,35,2.5,31,1.42,
296,43,290,0.164,9870,1.8,296,44,2,295,45,2.1,26,0.772,
293,51,340,0.175,577,1.8,291,55,1.7,291,55,1.7,21,0.0948,
292,53,284,0.171,1826,1.7,292,54,1.7,292,54,1.6,21,0.225,
297,41,299,0.151,15506,2.4,297,39,2.2,298,39,2.1,29,1.18,
293,50,293,0.162,3713,1.8,292,50,1.8,293,50,1.8,23,0.406,
295,44,291,0.159,10501,2,295,45,2,295,45,2.2,26,0.808
};

#define ALLOCBLOCKSZ 1024
double **dataset = NULL;
int nsamples = 0, allocsamples = 0;


// Low accuracy stub for modelling purposes
// Linear approximation for a fixed 120s timestep
// comments stripped for brevity, to understand whats going on, read
// the crop model source code
void modelStub(double *state, int addwater, double *retstate)
{
  double retable[] = {
    71.4000,60.1800,53.0400,46.9200,41.8200,37.7400,34.1700,31.1100,27.5400,
    25.5000,23.4600,21.9300,20.9100,19.8900,17.8500,16.3200,14.7900,13.7700,
    12.7500,12.2400, 11.2200,10.7100,9.6900,8.6700,8.6700,8.6700,8.1600,
    6.6300,5.6100,5.6100};
  double ET0COEFF[] = {
    1.0, 0.75,0.65,0.55,0.45,0.40,0.38,0.33,0.30,0.25
  };
  double photovsPPFD[] = {
    0.00, 0.16, 0.33, 0.46, 0.56, 0.66, 0.72, 0.78, 0.83, 0.86, 0.88, 0.92,
    0.95, 0.97, 1.00};
  double photosynthetic_efficiency = 0.83;
  double stochasticmagic = 3.5;
  double growthvshoursincesunrise[] = {
    0.0807,0.0719,0.0590,0.0462,0.0386,0.0339,0.0316,0.0298,0.0292,0.0292,
    0.0292,0.0292,0.0292,0.0292,0.0292,0.0292,0.0292,0.0304,0.0327,0.0357,
    0.0432,0.0549,0.0701,0.0783};
  double pot_volume = 425;
  double pot_surface_area = 44;
  double theta_fc = 0.248;
  double theta_wilt = 0.120;
  double theta_critical = 0.216;
  double theta_sat = 0.650;
  double gva, gv;
  double esatleaf, esatsoil, eair, molswaterpersecperm2;
  double potential_ml_lost_transpiration, wateravailability, transpirationloss;
  double Pn, PAR, PPFD, photosynthetic_rate, photopercent, photodiff, photointerp;
  int PPFDidx, PPFDidxb;
  double dmarea, carbon_exchanged;
  double drymassgain, wet_biomass_gain_in_g, ml_lost;
  int reidx;
  double soilresistance, boundaryresistance, latentflux, latentjoules;
  double ET0,megajoulesflux,esatair;
  double Pdiff, Pinterp, P, beta;
  int Pidx, Pidxb, i;
  double avail;
  double current_volumetric_ratio = state[3]*(theta_sat-theta_wilt)+theta_wilt;
  double current_leaf_area = state[4]/1e6;
  double current_shoot_biomass = 0.005 + current_leaf_area*50;

  if(addwater){
    //add 5ml as minimum increment size
    current_volumetric_ratio += 5/pot_volume;
```

```
}

for(i=0;i<15;++i){
  if(state[5] < 2.0)gva = 1.4*0.147*sqrt(2/0.0086);
  else gva = 1.4*0.147*sqrt(state[5]/0.0086);
  gv = (0.5*0.24*gva)/(0.24+gva) + (0.5*0.17*gva)/(0.17+gva);
  esatleaf = 0.61078*exp(17.269388*((state[0]-2-273.16)/
    (state[0]-2-35.86)));
  eair = (state[1]/100)*0.61078*exp(17.269388*((state[0]-273.16)/
    (state[0]-35.86)));
  molswaterpersecperm2 = gv*((esatleaf-eair)/100);

  esatair = 0.61078*exp(17.269388*((state[0]-273.16)/(state[0]-35.86)));
  megajoulesflux = state[12];
  if(state[5] < 2)ET0 = (0.006*megajoulesflux+(4.5/state[0])*(esatair-eair))
    /0.217;
  else ET0 = (0.006*megajoulesflux+(4.5/state[0])*state[5]*(esatair-eair))
    /(0.20 + 0.017*state[5]); //cm*d^-1
  if(ET0 > 0.9)P = 0.25;
  else{
    Pidx = (int)(floor(ET0*(sizeof(ET0COEFF)/sizeof(double))));
    Pidxb = (int)(ceil(ET0*(sizeof(ET0COEFF)/sizeof(double))));
    Pdiff = ET0COEFF[Pidxb]-ET0COEFF[Pidx];
    Pinterp = ET0*(sizeof(ET0COEFF)/sizeof(double)) - (double)Pidx;
    P = ET0COEFF[Pidx] + Pinterp*Pdiff;
  }

  theta_critical = (1-P)*(theta_fc-theta_wilt)+theta_wilt;
  if(current_volumetric_ratio >= theta_critical)beta = 1;
  else if(current_volumetric_ratio > theta_wilt && current_volumetric_ratio
    < theta_critical)beta = (current_volumetric_ratio-theta_wilt)/
                      (theta_critical-theta_wilt);
  else beta = 0;
  wateravailability = (1-exp(0.3*beta))/(1-exp(0.3));
  transpirationloss = potential_ml_lost_transpiration*wateravailability;
  current_volumetric_ratio = current_volumetric_ratio - (transpirationloss
    /pot_volume);

  Pn = -0.031*((state[0]-2-273.15)*(state[0]-2-273.15)) +
    1.897*(state[0]-2-273.15) - 5.229;
  Pn *=0.84;
  PAR = 0.45*state[2];
  PPFD = PAR*4.4;
  if(PPFD >= 700){
    photopercent = 1.0;
  }else{
    PPFDidx = (int)(floor((PPFD/700)*(sizeof(photovsPPFD)/sizeof(double))));
    PPFDidxb = (int)(ceil((PPFD/700)*(sizeof(photovsPPFD)/sizeof(double))));
    photodiff = photovsPPFD[PPFDidxb]-photovsPPFD[PPFDidx];
    photointerp = ((PPFD/700)*(sizeof(photovsPPFD)/sizeof(double)))
      - (double)PPFDidx;
    photopercent = photovsPPFD[PPFDidx] + photointerp*photodiff;
  }
  photosynthetic_rate = photopercent*Pn*wateravailability;
  if(photosynthetic_rate > 15*photosynthetic_efficiency)photosynthetic_rate
    = 15*photosynthetic_efficiency;

  if(PAR == 0){
    dmarea = 100*current_leaf_area;
    if(state[0]-2 < 5)carbon_exchanged = 0;
    else carbon_exchanged = ((-0.4*(state[0]-2-278.15))*dmarea)/(44000
      *(3600/120));
  }else carbon_exchanged = 0.000001*photosynthetic_rate*120*current_leaf_area;

  drymassgain = (carbon_exchanged*1000)/35;
  wet_biomass_gain_in_g = drymassgain/0.08;
  ml_lost = wet_biomass_gain_in_g-drymassgain;
  current_volumetric_ratio = current_volumetric_ratio - (ml_lost/pot_volume);
  current_shoot_biomass = current_shoot_biomass + 0.66*drymassgain;
  current_leaf_area=20*(current_shoot_biomass/1000);
```

```
    reidx = (int)floor(((current_volumetric_ratio-theta_wilt)/(theta_fc-
      theta_wilt))*(sizeof(retable)/sizeof(double)) + 0.5);
    if(reidx < 0)reidx = 0;
    else if(reidx >= (sizeof(retable)/sizeof(double)))reidx = (sizeof(retable)
      /sizeof(double));
    soilresistance = retable[reidx];
    if(state[5] < 1){
      boundaryresistance = (1/(0.1681))*log(2/0.001)*log(2/0.001);
    }else boundaryresistance = (1/(0.1681*state[5]))*log(2/0.001)*log(2/0.001);
    esatsoil = 0.61078*exp(17.269388*((state[0]-2-273.16)/(state[0]-2-35.86)));
    latentflux = (1005*1.2*(esatsoil-eair))/(0.0665*(boundaryresistance
      +soilresistance));
    latentjoules = latentflux*(pot_surface_area*0.0001)*120;
    ml_lost = (latentjoules/2.45e3);
    current_volumetric_ratio = current_volumetric_ratio - (ml_lost/pot_volume);

    if(current_volumetric_ratio < 0.75*theta_wilt
      current_volumetric_ratio = 0.75*theta_wilt;
    else if(current_volumetric_ratio > theta_fc)
      current_volumetric_ratio = theta_fc;
  }

  avail = (current_volumetric_ratio-theta_wilt)/(theta_sat-theta_wilt);
  memcpy(retstate, state, sizeof(double)*Rn);
  retstate[3] = avail;
  retstate[4] = current_leaf_area*1e6;
  retstate[13] = state[13]+0.005; //5ml increment size
}

//Marsaglia's xorshf generator, very fast PRNG
unsigned int xorshf96()
{
  static unsigned int x=123456789, y=362436069, z=521288629, t;
    x ^= x << 16; x ^= x >> 5; x ^= x << 1;
  t = x; x = y; y = z; z = t ^ x ^ y;
  return z>>17; //value in range 0-32767
}

//Cross platform fork functions by damian peckett
//dumb and ugly but portable!
#ifdef WINDOWS
void crossfork(char *argv[], char *argString[], HANDLE *child)
{
  char argStringOne[512];
  STARTUPINFO StartupInfo;
  PROCESS_INFORMATION ProcessInfo;

  sprintf(argStringOne, "\"%s\" \"%s\" \"%s\"", argv[0],
    argString[0], argString[1]);

  memset(&StartupInfo, 0, sizeof(StartupInfo));
  StartupInfo.cb = sizeof(STARTUPINFO);
  if(!CreateProcessA(argv[0], argStringOne, NULL, NULL, FALSE,
    NORMAL_PRIORITY_CLASS, NULL, NULL, &StartupInfo, &ProcessInfo))
    fprintf(stderr, "Error Executing Program (%d)\n", GetLastError());
  *child = ProcessInfo.hProcess;
  CloseHandle(ProcessInfo.hThread);
}
double getreturn(HANDLE child)
{
  char path[16];
  FILE *fp;
  double val;
  sprintf(path, "%d.ret", GetProcessId(child));
  WaitForSingleObject(child, INFINITE);
  do{
    fp = fopen(path, "rb");
    if(!fp)Sleep(50);
  }while(!fp);
```

```
  fscanf(fp, "%lf", &val);
  fclose(fp);
  sprintf(path, "del /Q %d.ret", GetProcessId(child));
  system(path);
  return val;
}
int returnvalue(double val)
{
  int pid;
  char path[16];
  FILE *fp;
  pid = GetCurrentProcessId();
  sprintf(path, "%d.ret", pid);
  fp = fopen(path, "w");
  fprintf(fp, "%g\n", val);
  fclose(fp);
  exit(0);
}
#else //likely linux
#include <unistd.h>
typedef unsigned int HANDLE;
void crossfork(char *argv[], char *argString[], HANDLE *child)
{
  *child = fork();
  if(!*child){
    execlp(argv[0], argv[0], argString[0], argString[1], NULL);
    exit(-1);
  }
}
double getreturn(HANDLE child)
{
  char path[16];
  FILE *fp;
  double val;
  sprintf(path, "/tmp/%d.ret", child);
  do{
    fp = fopen(path, "rb");
    if(!fp)usleep(50000);
  }while(!fp);
  fscanf(fp, "%lf", &val);
  fclose(fp);
  sprintf(path, "rm -f /tmp/%d.ret", child);
  system(path);
  return val;
}
int returnvalue(double val)
{
  int pid;
  char path[16];
  FILE *fp;
  pid = getpid();
  sprintf(path, "/tmp/%d.ret", pid);
  fp = fopen(path, "w");
  fprintf(fp, "%g\n", val);
  fclose(fp);
  exit(0);
}
#endif

void generatePhi(double *phi, double *state, double *RBFVariances, int singleIndex)
{
  double L2Sum;
  int i, j;

  phi[0] = 1.0; //Constant Bias
  for(j = ((singleIndex>0)?(singleIndex-1):0); j < ((singleIndex>0)?singleIndex:NRBF); ++j){
    for(i = 0, L2Sum = 0; i < Rn; ++i)
      L2Sum+=(state[i]-RBFCentroids[j*Rn+i])*(state[i]-RBFCentroids[j*Rn+i]);
    phi[1+j] = exp(-RBFVariances[j]*L2Sum);
```

```
  }
}

double RbfNetwork(double *knownphi, double *state, double *weights, double *RBFVariances)
{
    int i;
    double phi[NRBF+1];
    double networkValue;

    //generate features
    if(knownphi)memcpy(phi, knownphi, sizeof(phi));
    else generatePhi(phi, state, RBFVariances, -1);

    //Sum With Our Weightings
    for(i = 0, networkValue = 0; i < NRBF+1; ++i)
      networkValue+=weights[i]*phi[i];

    //return value
    return networkValue;
}

double reward(double *state)
{
  double rew;

  //prevent divide by zero
  if(state[13] <= 1e-5)
    state[13] = 0.005;

  //ratio of biomass to applied irrigation
  rew = state[4] / state[13];

  return rew;
}

//http://www.research.rutgers.edu/~lihong/pub/Zinkevich11Parallelized.pdf
#define SGDTHREADS 6
#define LEARNRATE 1e-4 //tuned experimentally
struct stochParam{
  unsigned int *trainIndex;
  int minIndex;
  int maxIndex;
  double wi[NRBF+1];
  double weights[NRBF+1];
  double variances[NRBF+1];
  double *y;
};
void *SimuParallelSGD(void *vparam)
{
  struct stochParam *param =
    (struct stochParam *)vparam;
  double hypothesis, phi[NRBF+1];
  double **chunk, *y, *tmp, tmpd;
  int i, j, from, to, nsamples;
  double wi[NRBF+1];

  //Allocate local memory, don't have to deal with
  //memory locks etc then
  nsamples = param->maxIndex-param->minIndex+1;
  chunk = (double**)malloc(nsamples*sizeof(double**));
  y = (double*)malloc(nsamples*sizeof(double));

  for(i = param->minIndex; i <= param->maxIndex; ++i){
    chunk[i-param->minIndex] = dataset[param->trainIndex[i]];
    y[i-param->minIndex] = param->y[i];
  }

  //shuffle our chunk
  for(i = 0; i < nsamples/4; ++i){
    from=(xorshf96()*nsamples)/32768;
    to=(xorshf96()*nsamples)/32768;
```

```
    tmp = chunk[from];
    chunk[from] = chunk[to];
    chunk[to] = tmp;

    tmpd=y[from];
    y[from] = y[to];
    y[to] = tmpd;
  }

  //zero wi in the original algorithm
  //set to iterative approach in mine
  memcpy(wi, param->weights, sizeof(wi));

  //parallel stochastic gradient descent
  for(i = 0; i < nsamples; ++i){
    generatePhi(phi, chunk[i], param->variances, -1);
    hypothesis = RbfNetwork(phi, chunk[i], param->weights, param->variances);
    for(j = 0; j < (NRBF+1); ++j)
      wi[j]+=LEARNRATE*(y[i]-hypothesis)*phi[j];
  }

  //copy our results
  memcpy(param->wi, wi, sizeof(param->wi));

  //free local memory
  free(y);
  free(chunk);
  pthread_exit(NULL);
}

double valueIterationAndHoldoutCheck(double *variances, double *weights, int maxIter)
{
  //indexs
  int i, j;
  unsigned int trainIndex[TRAINSAMPLES];
  unsigned int validIndex[HOLDOUTSAMPLES];

  //mdp stuff
  int iter, gradIter;
  double y[TRAINSAMPLES];
  double outstate[14], noapply, apply;
  double diff;
  double yi;

  //Parallel stuff
  pthread_t threadHandle[SGDTHREADS];
  struct stochParam threadParam[SGDTHREADS];
  int samplesPerWorker, ret;

  //select representative samples to train with
  for(i = 0; i < TRAINSAMPLES; ++i)
    trainIndex[i] = xorshf96();
  //select holdout samples
  for(i = 0; i < HOLDOUTSAMPLES; ++i)
    validIndex[i] = xorshf96();

  //clear memory
  memset(y, 0, sizeof(y));

  for(iter = 0; iter < maxIter; ++iter){
    for(i = 0; i < TRAINSAMPLES; ++i){
      // k = 1 because our modelstub is deterministic
      // Action Don't Apply Water
      modelStub(dataset[trainIndex[i]], 0, outstate);
      noapply = reward(dataset[trainIndex[i]])
        + DISCOUNT*RbfNetwork(NULL, outstate, weights, variances);

      // Action Apply Water
      modelStub(dataset[trainIndex[i]], 1, outstate);
      apply = reward(dataset[trainIndex[i]])
        + DISCOUNT*RbfNetwork(NULL, outstate, weights, variances);
```

```
      if(apply > noapply)y[i] = apply;
      else y[i] = noapply;
    }

    //ratio of optimizaztion to bellman loops tuned experimentally
    for(gradIter=0; gradIter < (int)ceil((double)7500/TRAINSAMPLES); ++gradIter){
      samplesPerWorker=TRAINSAMPLES/SGDTHREADS;
      for(i = 0; i < SGDTHREADS; ++i){
        threadParam[i].trainIndex = trainIndex;
        threadParam[i].minIndex = i*samplesPerWorker;
        threadParam[i].maxIndex = i*samplesPerWorker + samplesPerWorker - 1;
        memcpy(threadParam[i].weights, weights, sizeof(threadParam[i].weights));
        memcpy(threadParam[i].variances, variances, sizeof(threadParam[i].variances));
        threadParam[i].y = y;
      }

      //spawn our threads
      for(i = 0; i < SGDTHREADS; ++i){
      if(pthread_create(&threadHandle[i], NULL, SimuParallelSGD, (void *)&threadParam[i]))
        fprintf(stderr,"Ignoring Broken Thread\n");
      }

      //Wait For All The Threads To Return
      for(i = 0; i < SGDTHREADS; ++i)
        pthread_join(threadHandle[i], (void**)&ret);

      //update parameters
      for(j = 0; j < (NRBF+1); ++j){
        //calculate a mean of the parallel runs
        weights[j] = 0;
        for(i = 0; i < SGDTHREADS; ++i){
          weights[j]+=threadParam[i].wi[j];
        }
        weights[j]/=(double)SGDTHREADS;
      }
    }

    // Holdout Cross Validation
    diff = 0;
    for(i = 0; i < HOLDOUTSAMPLES; ++i){
        // Action Don't Apply Water
        modelStub(dataset[trainIndex[i]], 0, outstate);
        noapply = reward(dataset[validIndex[i]]) + DISCOUNT
          *RbfNetwork(NULL, outstate, weights, variances);
        // Action Apply Water
        modelStub(dataset[trainIndex[i]], 1, outstate);
        apply = reward(dataset[validIndex[i]]) + DISCOUNT
          *RbfNetwork(NULL, outstate, weights, variances);
        if(apply > noapply)yi = apply;
        else yi = noapply;
        diff += fabs((RbfNetwork(NULL, dataset[validIndex[i]],
          weights, variances) - yi)/yi);
    }
    diff = diff / HOLDOUTSAMPLES;

    //Have we converged yet?
    //general cutoff of error within 10%
    //minimum of 25 iterations for consistency
    //if(iter >= 25 && diff < 0.10){
    //  fprintf(stderr,"Value Iteration Converged In %d Iterations\n", iter);
    //  fprintf(stderr,"Final Cross Validation Error: %.3g Percent\n", diff*100);
    //  break;
    //}
  }

  return diff;
}

//Tuned experimentally, and from experience
```

```
#define STEPTHREADS 8 //must be a multiple of 2
#define INITIALGAMMA 10e-9

struct workerArg{
  double variances[NRBF];
  double weights[NRBF+1];
  double score;
};

int main(int argc, char *argv[])
{
  int i, j;
  char line[256], *ptr;
  double *leakchunk;
  FILE *fp;

  int step, maxsteps = 50;

  //network parameters
  double *variances;
  double *weights;
  double val;
  double stepsz;
  int unchanged = 0;

  //thread stuff
  struct workerArg arg[STEPTHREADS+1];
  char **argPass;
  HANDLE handles[STEPTHREADS+1];

  argPass = (char**)malloc(sizeof(char*)*2);
  argPass[0] = (char*)malloc(512);
  argPass[1] = (char*)malloc(512);

  fp = fopen("data.csv", "rb");
  if(!fp)return -1;

  //fast file load
  do{
    //Need More Memory ?
    if(nsamples >= allocsamples){
      //purposely leak this chunk of heap memory, loading speed up
      leakchunk = (double*)malloc(sizeof(double)*ALLOCBLOCKSZ*Rn);
      dataset = (double**)realloc(dataset, (allocsamples+ALLOCBLOCKSZ)
        *sizeof(double*));
      for(i = 0; i < ALLOCBLOCKSZ; ++i)
        dataset[i+allocsamples] = leakchunk+i*Rn;
      allocsamples+=ALLOCBLOCKSZ;
    }

    if(!fgets(line, sizeof(line), fp))
      break; //read error

    //replace comma with space
    for(ptr = line; *ptr; ++ptr)
      if(*ptr == ',')*ptr = ' ';

    sscanf(line, "%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf",
      dataset[nsamples], dataset[nsamples]+1, dataset[nsamples]+2,
      dataset[nsamples]+3,dataset[nsamples]+4, dataset[nsamples]+5,
      dataset[nsamples]+6, dataset[nsamples]+7,dataset[nsamples]+8,
      dataset[nsamples]+9, dataset[nsamples]+10, dataset[nsamples]+11,
      dataset[nsamples]+12, dataset[nsamples]+13);
    nsamples++;
  }while(!feof(fp));
  fclose(fp);

  weights = (double*)malloc(sizeof(double)*(NRBF+1));
  variances = (double*)malloc(sizeof(double)*NRBF);

  if(argc == 3){ //child process
    //load weights from command line
```

```
    for(i = 0, ptr = argv[1]; i < (NRBF+1); ++i){
      if(i)for(;*ptr; ptr++)
        if(*ptr == ' '){ptr++;break;}
      sscanf(ptr, "%lf", &weights[i]);
    }

    //load variances from command line
    for(i = 0, ptr = argv[2]; i < NRBF; ++i){
      if(i)for(;*ptr; ptr++)
        if(*ptr == ' '){ptr++;break;}
      sscanf(ptr, "%lf", &variances[i]);
    }

    //calculate for 100 iterations
    val = valueIterationAndHoldoutCheck(variances, weights, 100);
    fprintf(stderr, "Child Calculated: %g Percent\n", val*100);
    returnvalue(val*100);
  }else{
    stepsz = 10; //logsearch

    //initial conditions
    memset(weights, 0, sizeof(double)*(NRBF+1));
    for(j = 0; j < NRBF; ++j)variances[j] = INITIALGAMMA;

    if(STEPTHREADS%2){
      fprintf(stderr, "Number of threads must be a multiple of 2!\n");
      return -1;
    }

    //random coordinate descent
    //with some twists by Damian Peckett
    fprintf(stderr,"Performing Gradient Descent On RBF Variances\n");
    for(step = 0; step < maxsteps; ++step){
      //tuned experimentally
      if(step){
        stepsz/=2; //gradually resduce our grid size
      }

      //set memory
      memset(arg, 0, sizeof(arg));
      for(i = 0; i < (STEPTHREADS+1); ++i){
        memcpy(arg[i].variances, variances, sizeof(arg[i].variances));
        memcpy(arg[i].weights, weights, sizeof(arg[i].weights));
      }

      //step forwards and back, log search
      //arg[0] is current position
      for(i = 0; i < STEPTHREADS/2; ++i){
        for(j = 0; j < NRBF; ++j){
          arg[1+i*2].variances[j]*=pow(stepsz,(double)i); //step forward
          arg[2+i*2].variances[j]/=pow(stepsz,(double)i); //step back
        }
      }

      //spawn our worker processes
      for(i = 0; i < (STEPTHREADS+1); ++i){
        //construct our arguments
        //long because its generalized
        argPass[0][0] = 0;
        for(j = 0; j < (NRBF+1); ++j){
          sprintf(line, "%g ", arg[i].weights[j]);
          strcat(argPass[0], line);
        }
        argPass[1][0] = 0;
        for(j = 0; j < NRBF; ++j){
          sprintf(line, "%g ", arg[i].variances[j]);
          strcat(argPass[1], line);
        }

        //spawn the process
```

```
      crossfork(argv, argPass, &handles[i]);
    }

    //wait for the worker processes to finish
    //and record their results
    for(i = 0; i < (STEPTHREADS+1); ++i){
      arg[i].score = getreturn(handles[i]);
      if(arg[i].score <= 0)arg[i].score = 100;
    }

    //make changes
    for(i = 0, j = 0; i < STEPTHREADS/2; ++i){
      if(arg[1+i*2].score < arg[0].score){
        for(j = 0; j < NRBF; ++j)
        variances[j]*=pow(stepsz,(double)i); //step forward
        break;
      }else if(arg[2+i*2].score < arg[0].score){
        for(j = 0; j < NRBF; ++j)
        variances[j]/=pow(stepsz,(double)i); //step back
        break;
      }else j++;
    }

    if(unchanged == 3)break;
    if(j == STEPTHREADS/2)unchanged++; //pretty much converge
    else unchanged=0;

    fprintf(stderr,"Updated Variances:\n");
    for(j = 0; j < NRBF; ++j)
      fprintf(stderr,"%.5g ", variances[j]);
    fprintf(stderr,"\n");
  }

  //Calculate The Corresponding weights
  fprintf(stderr, "Winning Variance, gamma = %g\n", variances[0]);
  val = valueIterationAndHoldoutCheck(variances, weights, 500);
  fprintf(stderr, "Winning Weights, Accurate To (%g Percent)\n", val*100);
  for(i = 0; i < (NRBF+1); ++i)
    fprintf(stderr, "%g,", weights[i]);
  fprintf(stderr, "\n");

  return 0;
 }

 return -1;
}
```

Appendix L

Generate Seed Clusters

```
% You are welcome to use this however you wish, however I'd love an email
% if you do something cool with it! Word Out! Damo
%
%  Copyright (c) 2013, Damian Peckett <damian.peckett@gmail.com>
%  All rights reserved.
%
% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are met:
%
% 1. Redistributions of source code must retain the above copyright notice, this
%    list of conditions and the following disclaimer.
% 2. Redistributions in binary form must reproduce the above copyright notice,
%    this list of conditions and the following disclaimer in the documentation
%    and/or other materials provided with the distribution.
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
% ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
% WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
% DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
% ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
% (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
% LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
% ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
% (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
% SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

% GENERATE A POOL OF POSSIBLE CLUSTER CENTROIDS
% BASED ON KMEANS

dataset = load('datapoints.csv');

NCLUSTERS = 8;

fprintf('Generating Seed Cluster Pool\n');

% Take a random selection of 1000 points from the dataset
points = round(1 + (length(dataset)-1).*rand(1000));
subset = zeros(length(points), 14);
for i=1:length(points)
  subset(i,:) = dataset(points(i),:);
end

for z=1:10

  opts = statset('MaxIter',10000);
  [~,centroid] = kmeans(subset, NCLUSTERS, 'options', opts);
```

```
  fname = sprintf('centers.txt', z);
  fileID = fopen(fname, 'a+');

  for i=1:NCLUSTERS
    fprintf(fileID, '%g_%g_%g_%g_%g_%g_%g_%g_%g_%g_%g_%g_%g\n', ...
      centroid(i,1),centroid(i,2),centroid(i,3),centroid(i,4),centroid(i,5),centroid(i,6), ...
      centroid(i,7),centroid(i,8),centroid(i,9),centroid(i,10),centroid(i,11),centroid(i,12),centroid(i,13),
    );
  end

  fclose(fileID);
end
```

Appendix M

Average Out Clusters

```
% You are welcome to use this however you wish, however I'd love an email
% if you do something cool with it! Word Out! Damo
%
%  Copyright (c) 2013, Damian Peckett <damian.peckett@gmail.com>
%  All rights reserved.
%
% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are met:
%
% 1. Redistributions of source code must retain the above copyright notice, this
%    list of conditions and the following disclaimer.
% 2. Redistributions in binary form must reproduce the above copyright notice,
%    this list of conditions and the following disclaimer in the documentation
%    and/or other materials provided with the distribution.
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
% ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
% WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
% DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
% ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
% (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
% LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
% ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
% (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
% SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

% Find Some Averages From Our Pool Of Clusters

X = load('centers.txt');

NUMBERCLUSTERS = 8;

% Group Close Together Points
% http://homepage.tudelft.nl/19j49/t-SNE.html
X2D = tsne(X);
distances = squareform(pdist(X2D));
closethresh = 0.5; % Tuned Experimentally

while 1
  groups = zeros(length(X2D),1);
  curgroupid = 1;

  for i=1:length(distances)
    Z = distances(i,i:length(distances)) < closethresh;

    assign_group = curgroupid;
    for j=1:length(Z)
```

```
          if Z(j) > 0
            if groups(j+i-1) ~= 0
              assign_group = groups(j+i-1);
              break;
            end
          end
        end

        for j=1:length(Z)
          if Z(j) > 0
            groups(j+i-1) = assign_group;
          end
        end

        if assign_group == curgroupid
          curgroupid = curgroupid + 1;
        end

      end

      fprintf('%d Clusters This Run\n', curgroupid-1);

      %got the desired number of clusters
      if curgroupid-1 <= NUMBERCLUSTERS
        break;
      else
        closethresh = closethresh + 0.1; %expand out
      end
    end

    % Calculate The Averages Of Each Group
    avgcluster = [];
    outfp = fopen('averagecenters.txt', 'w');
    for i=1:(curgroupid-1)
      sum = zeros(1,14);
      npoint = 0;

      for j=1:length(groups)
        if groups(j,1) == i
          sum = sum + X(j,:);
          npoint = npoint + 1;
        end
      end

      sum = sum / npoint;
      avgcluster = [avgcluster; sum];
      fprintf(outfp, '%d %d %d %0.3g %d %2.2g %d %d %2.2g %d %d %2.2g %2.2g %0.3g\n', ...
      round(sum(1)), round(sum(2)), round(sum(3)), sum(4), round(sum(5)), sum(6), ...
      round(sum(7)), round(sum(8)), sum(9), round(sum(10)), ...
      round(sum(11)), sum(12), sum(13), sum(14));
    end
    fclose(outfp);


    %Display data
    labels = num2str(groups,'%d');
    gscatter(X2D(:,1),X2D(:,2),labels);
```